Knowledge Discovery and Data Mining 1
(Data Mining Algorithms 1)

Winter Semester 2019/20
Agenda

1. Introduction
   1.1 Organisation
   1.2 Motivation
   1.3 Knowledge Discovery Process

2. Basics

3. Supervised Methods

4. Unsupervised Methods

5. Process Mining
People

Lecturer

- Prof. Dr. Thomas Seidl

Assistants

- Maximilian Huenemoerder
- Janina Sontheim

Student Assistants

- Marcel Baur
- Florian von Keller
## Schedule

### Lecture (begins: 15.10.2018)

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue</td>
<td>09:15-11:45</td>
<td>E004 (HGB)</td>
</tr>
</tbody>
</table>

### Tutorials (begin: 24.10.2018)

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thu</td>
<td>12:15-13:45</td>
<td>Lehrturm-VU107 (Prof.-Huber-Pl. 2)</td>
</tr>
<tr>
<td>Thu</td>
<td>14:15-15:45</td>
<td>Lehrturm-VU107 (Prof.-Huber-Pl. 2)</td>
</tr>
<tr>
<td>Fri</td>
<td>12:15-13:45</td>
<td>Lehrturm-V005 (Prof.-Huber-Pl. 2)</td>
</tr>
<tr>
<td>Fri</td>
<td>14:15-15:45</td>
<td>Lehrturm-V005 (Prof.-Huber-Pl. 2)</td>
</tr>
</tbody>
</table>

### Exam

1. Hauptklausur: tba
2. Nachholklausur: tba
Material, Tutorials & Exam

Material (Slides, Exercises, etc.)

Available on course webpage:


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

1. Introduction
   1.1 Organisation
   1.2 Motivation
   1.3 Knowledge Discovery Process

2. Basics

3. Supervised Methods

4. Unsupervised Methods

5. Process Mining
Textbook / Acknowledgements

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


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 \textit{(non-trivial, implicit, previously unknown and potentially useful)} information or patterns from data in \textit{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
Stairs of Knowledge (K. North) \(^1\)

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

1. Introduction
   1.1 Organisation
   1.2 Motivation
   1.3 Knowledge Discovery Process

2. Basics

3. Supervised Methods

4. Unsupervised Methods

5. Process Mining
The Knowledge Discovery Process

- The KDD-Process (Knowledge Discovery in Databases)

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

1. Introduction
1.3 Knowledge Discovery Process
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]
    \[\text{JoinedTable} = [O Id, Info1, Info2]\]
- 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., \((\text{id}, \text{name}, \text{date}, \text{location}, \text{amount}) \mapsto (\text{id}, \text{date}, \text{amount})\)

### Transformations, e.g.:

- Discretization of numerical attributes, e.g.,
  \(\text{amount}: [0, 100] \mapsto \text{d}_\text{amount}: \{\text{low}, \text{medium}, \text{high}\}\)
- Computation of derived tuples/rows and derived attributes:
  - aggregation of sets of tuples, e.g., total amount per months
  - new attributes, e.g., \(\text{diff} = \text{sales current month} - \text{sales previous month}\)
KDD Process: Basic Data Mining Tasks

Goal
Find patterns of interest

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

**Setting**

Given a database of transactions, e.g.

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Items Bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>A,B,C</td>
</tr>
<tr>
<td>1000</td>
<td>A,C</td>
</tr>
<tr>
<td>4000</td>
<td>A,D</td>
</tr>
<tr>
<td>5000</td>
<td>B,E,F</td>
</tr>
</tbody>
</table>

**Motivation**

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

**Examples**

- buys(x, "diapers") ⇒ buys(x, "beers")  [supp: 0.5%, conf: 60%]
- major(x, "CS") ∧ takes(x, "DB") ⇒ grade(x,"A")  [supp: 1.0%, conf: 75%]
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 $sim : O \times O \rightarrow \mathbb{R}$

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

<table>
<thead>
<tr>
<th>Outlier Detection</th>
<th>Find objects that do not comply with the general behaviour of the data (fraud detection, rare events analysis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trends and Evolution Analysis</td>
<td>Sequential patterns (find re-occurring sequences of events)</td>
</tr>
<tr>
<td>Methods for special data types, and applications</td>
<td></td>
</tr>
</tbody>
</table>
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

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

1. Introduction

2. Basics
   2.1 Data Representation
   2.2 Data Reduction
   2.3 Visualization
   2.4 Privacy

3. Supervised Methods

4. Unsupervised Methods

5. Process Mining
Objects and Attributes (Conceptual Modeling)

Entity-Relationship Diagram (ER)

UML Class Diagram

Data Tables (Relational Model)
<table>
<thead>
<tr>
<th>Overview of (Attribute) Data Types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simple Data Types</strong></td>
</tr>
<tr>
<td>Numeric/metric, Categorical/nominal, ordinal</td>
</tr>
<tr>
<td><strong>Composed Data Types</strong></td>
</tr>
<tr>
<td>Sets, sequences, vectors</td>
</tr>
<tr>
<td><strong>Complex Data Types</strong></td>
</tr>
<tr>
<td>▶ Multimedia: images, videos, audio, text, documents, web pages, etc.</td>
</tr>
<tr>
<td>▶ Spatial, geometric: shapes, molecules, geography, etc.</td>
</tr>
<tr>
<td>▶ Structures: graphs, networks, trees, etc.</td>
</tr>
</tbody>
</table>
## 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, \ldots \}
  - subjects = \{ physics, biology, math, music, literature, \ldots \}
- 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

\[ \text{Dedicated similarity matrix, } O \times O \rightarrow \mathbb{R}_0^+ \]
Metric Space

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

- **Symmetry:** \(\forall p, q \in O : d(p, q) = d(q, p)\)
- **Identity of Indiscernibles:** \(\forall p, q \in O : d(p, q) = 0 \iff p = q\)
- **Triangle Inequality:** \(\forall p, q, o \in O : d(p, q) \leq d(p, o) + d(o, q)\)

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 $O$:

- **Transitivity:** $\forall p, q, o \in O : p \leq q \land q \leq o \implies p \leq o$
- **Antisymmetry:** $\forall p, q \in O : p \leq q \land q \leq p \implies p = q$
- **Totality:** $\forall p, q \in O : p \leq q \lor q \leq p$

### Examples

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

**Characteristic**

Unordered collection of individual values

**Example**

- skills = \{Java, C, Python\}

**Comparison**

- Symmetric Set Difference:
  \[ R \Delta S = (R - S) \cup (S - R) \]
  \[ = (R \cup S) - (R \cap S) \]

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

<table>
<thead>
<tr>
<th></th>
<th>Direct</th>
<th>Feature engineering</th>
<th>Feature learning</th>
<th>Kernel-based</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graphs</strong></td>
<td>Structural Alignment</td>
<td>Degree Histograms</td>
<td>Node embeddings</td>
<td>Label Sequence Kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Geometry</strong></td>
<td>Hausdorff Distance</td>
<td>Shape Histograms</td>
<td>Spectral Neural Network</td>
<td>Spatial Pyramid Kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sequences</strong></td>
<td>Edit Distance</td>
<td>Symbol Histograms</td>
<td>Recurrent neural network (RNN)</td>
<td>Cosine Distance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 $\varepsilon \in \mathbb{R}_0^+$:

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

**Nearest neighbor query**

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

**k-nearest neighbor query**

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

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

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

**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 \(\text{range}(DB, q, d, \varepsilon) = \{o \in DB \mid d(o, q) \leq \varepsilon\}\)
- 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 \(O(n), O(\log n), \text{or even } O(1)\)
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

- ICES criteria for filter quality
  - I ndexable – Index enabled
  - C omplete – No false dismissals
  - E fficient – Fast individual calculation
  - S elective – Small candidate set

\[^a\text{GEMINI: Faloutsos 1996; KNOP: Seidl & Kriegel 1998}\
\[^b\text{Assent, Wenning, Seidl: ICDE 2006}\

2. Basics 2.1 Data Representation
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

- ICES criteria for filter quality
  - **Indexable** – Index enabled
  - **Complete** – No false dismissals
  - **Efficient** – Fast individual calculation
  - **Selective** – 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
Agenda

1. Introduction

2. Basics
   2.1 Data Representation
   2.2 Data Reduction
   2.3 Visualization
   2.4 Privacy

3. Supervised Methods

4. Unsupervised Methods

5. Process Mining
## 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

Numerosity Reduction
Reduce number of objects

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

Dimensionality Reduction
Reduce number of attributes

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

Quantization, Discretization
Reduce number of values per domain

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 → add counting attribute and merge duplicate tuples

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Major</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann</td>
<td>27</td>
<td>CS</td>
</tr>
<tr>
<td>Bob</td>
<td>26</td>
<td>CS</td>
</tr>
<tr>
<td>Eve</td>
<td>19</td>
<td>CS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Major</th>
</tr>
</thead>
<tbody>
<tr>
<td>(any)</td>
<td>Twen</td>
<td>CS</td>
</tr>
<tr>
<td>(any)</td>
<td>Twen</td>
<td>CS</td>
</tr>
<tr>
<td>(any)</td>
<td>Teen</td>
<td>CS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Age</th>
<th>Major</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twen</td>
<td>CS</td>
<td>2</td>
</tr>
<tr>
<td>Teen</td>
<td>CS</td>
<td>1</td>
</tr>
</tbody>
</table>

2. Basics 2.2 Data Reduction
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**

- $\text{avg}(D_1 \cup D_2) = \frac{\text{sum}(D_1 \cup D_2)}{\text{count}(D_1 \cup D_2)} = \frac{\text{sum}(D_1) + \text{sum}(D_2)}{\text{count}(D_1) + \text{count}(D_2)}$
- Note: $\text{avg}$ is not distributive, $\text{avg}(D_1 \cup D_2) \neq \text{avg}(\text{avg}(D_1), \text{avg}(D_2))$
- $\text{standard deviation}(D_1 \cup D_2)$
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)

  $\text{median}(D_1 \cup D_2) \neq \text{simple-function}(\text{median}(D_1), \text{median}(D_2))$

- **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").

\[
\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \quad \text{and} \quad \bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}
\]

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, rarely, often, usually, usually, always
- tiny, small, big, big, big, big, big, big, huge, huge
- tiny, tiny, small, medium, big, big, large, huge

*What if there is no ordering?*
Measuring the Central Tendency

Unimodal

Bimodal

Mode

- Value that occurs most frequently in the data
- Example: blue, red, blue, yellow, green, blue, red
- Unimodal, bimodal, trimodal, \ldots: There are 1, 2, 3, \ldots 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

Iris Dataset

<table>
<thead>
<tr>
<th>setosa</th>
<th>versicolor</th>
<th>virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>5.0</td>
<td>5.5</td>
</tr>
<tr>
<td>6.0</td>
<td>6.5</td>
<td>7.0</td>
</tr>
<tr>
<td>7.0</td>
<td>7.5</td>
<td>8.0</td>
</tr>
</tbody>
</table>

sepal length (cm)

2. Basics 2.2 Data Reduction
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

2. Basics
2.2 Data Reduction
Concept Hierarchies: Examples

Hierarchies from schema information

Place: all
- North America
  - Canada
  - Vancouver
  - Toronto
- Asia
- Europe
  - Germany
  - Aachen
  - Munich
  - France

Major: all
- Science
  - CS
  - Math
  - Physics
- Business
- Engineering
  - EE
  - ...
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.

Conceptual levels:

1. all
2. federal states
3. states
4. countries
5. cities

Example:

- 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

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

Roll-Up

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

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

Drill-Down

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

```sql
SELECT country, quarter, product, some_agg_fnc(...) 
FROM business 
GROUP BY country, quarter, product
```
Example: Roll up and Drill-Down in a Data Cube
Example: Slice Operation

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

VCR dimension is chosen

![Diagram showing the slice operation with dimensions Product (TV, PC, VCR), Quarter (1Qtr, 2Qtr, 3Qtr, 4Qtr), and Country (U.S.A, Canada, Mexico). The sum, avg, count metrics are indicated.]
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
Example: Pivot (rotate)

<table>
<thead>
<tr>
<th>year</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TV</td>
<td>PC</td>
<td>VCR</td>
</tr>
</tbody>
</table>

\[\Downarrow\] Pivot (rotate) \[\Downarrow\]

<table>
<thead>
<tr>
<th>product</th>
<th>TV</th>
<th>PC</th>
<th>VCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

2. Basics

2.2 Data Reduction
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
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

<table>
<thead>
<tr>
<th>Attribute Removal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove attribute $A$ in the following cases:</td>
</tr>
<tr>
<td>▶ There is a large set of distinct values for $A$ but there is no generalization operator (concept hierarchy) on $A$, or</td>
</tr>
<tr>
<td>▶ $A$’s higher level concepts are expressed in terms of other attributes (e.g. street is covered by city, state, country).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute Generalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 $A$.</td>
</tr>
</tbody>
</table>
### Attribute Oriented Induction: Example

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

- **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:

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

▶ Prime Generalized Relation:

<table>
<thead>
<tr>
<th>Gender</th>
<th>Major</th>
<th>Birth region</th>
<th>Age Range</th>
<th>Residence</th>
<th>GPA</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Science</td>
<td>Canada</td>
<td>20-25</td>
<td>Richmond</td>
<td>Very good</td>
<td>16</td>
</tr>
<tr>
<td>F</td>
<td>Science</td>
<td>Foreign</td>
<td>25-30</td>
<td>Burnaby</td>
<td>Excellent</td>
<td>22</td>
</tr>
</tbody>
</table>

▶ Crosstab for generalized relation:

<table>
<thead>
<tr>
<th></th>
<th>Canada</th>
<th>Foreign</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>16</td>
<td>14</td>
<td>30</td>
</tr>
<tr>
<td>F</td>
<td>10</td>
<td>22</td>
<td>32</td>
</tr>
<tr>
<td>Total</td>
<td>26</td>
<td>36</td>
<td>62</td>
</tr>
</tbody>
</table>
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
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

1. Introduction

2. Basics
   2.1 Data Representation
   2.2 Data Reduction
   2.3 Visualization
   2.4 Privacy

3. Supervised Methods

4. Unsupervised Methods

5. Process Mining
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]

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

2. Basics

2.3 Visualization

88
# Data Visualization Techniques

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

Characteristics

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

Benefits

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

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.
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.)

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
Again, the ordering and orientation of the dimensions is important to reveal correlations. The farther the attributes, the harder their correlations are perceived; coloring helps visualize clusters and 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

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?

Chernoff Faces

Example: Weather Data

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

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

## Chernoff Faces

### Example: Finance Data

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

<table>
<thead>
<tr>
<th>Date Dimensions</th>
<th>5</th>
<th>4</th>
<th>Year to Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Return on Assets</td>
<td>0.10</td>
<td>0.11</td>
<td>0.06</td>
</tr>
<tr>
<td>2. Debt Service</td>
<td>3.66</td>
<td>3.79</td>
<td>1.55</td>
</tr>
<tr>
<td>3. Cash Flows</td>
<td>1.53</td>
<td>1.48</td>
<td>1.39</td>
</tr>
<tr>
<td>4. Capitalization</td>
<td>0.22</td>
<td>0.20</td>
<td>0.18</td>
</tr>
<tr>
<td>5. Current Ratio</td>
<td>71.40</td>
<td>89.10</td>
<td>97.85</td>
</tr>
<tr>
<td>6. Cash Turnover</td>
<td>24.03</td>
<td>25.92</td>
<td>25.62</td>
</tr>
<tr>
<td>7. Receivables Turnover</td>
<td>5.25</td>
<td>4.46</td>
<td>4.26</td>
</tr>
<tr>
<td>8. Inventory Turnover</td>
<td>5.38</td>
<td>4.77</td>
<td>4.57</td>
</tr>
<tr>
<td>9. Sales per Dollar</td>
<td>Working Capital</td>
<td>6.74</td>
<td>6.33</td>
</tr>
<tr>
<td>10. Retained Earning/Total Assets</td>
<td>0.32</td>
<td>0.30</td>
<td>0.01</td>
</tr>
<tr>
<td>11. Total Assets</td>
<td>0.94</td>
<td>0.76</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Agenda

1. Introduction

2. Basics
   2.1 Data Representation
   2.2 Data Reduction
   2.3 Visualization
   2.4 Privacy

3. Supervised Methods

4. Unsupervised Methods

5. Process Mining
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,

  Strong Privacy Protection $\implies$ Poor Data Utility
  Good Data Utility $\implies$ Weak Privacy Protection

**Challenge**

Find a good trade-off between Data Utility and 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

<table>
<thead>
<tr>
<th>Private</th>
<th>Public</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Sex</td>
<td>Age</td>
</tr>
<tr>
<td>Alice</td>
<td>F</td>
<td>29</td>
</tr>
<tr>
<td>Janes</td>
<td>F</td>
<td>27</td>
</tr>
<tr>
<td>Jones</td>
<td>M</td>
<td>21</td>
</tr>
<tr>
<td>Frank</td>
<td>M</td>
<td>35</td>
</tr>
<tr>
<td>Ben</td>
<td>M</td>
<td>33</td>
</tr>
<tr>
<td>Betty</td>
<td>F</td>
<td>37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Public Records from Sport Club</th>
<th>Public</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Sex</td>
</tr>
<tr>
<td>Alice</td>
<td>F</td>
</tr>
<tr>
<td>Theo</td>
<td>M</td>
</tr>
<tr>
<td>John</td>
<td>M</td>
</tr>
<tr>
<td>Betty</td>
<td>F</td>
</tr>
<tr>
<td>James</td>
<td>M</td>
</tr>
</tbody>
</table>
**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.)
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.

⇒ Alice has BC!

<table>
<thead>
<tr>
<th>Release</th>
<th>Quasi Identifier</th>
<th>Sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex  Age Zip</td>
<td>Disease</td>
<td></td>
</tr>
<tr>
<td>F  2? 520??</td>
<td>Breast Cancer</td>
<td></td>
</tr>
<tr>
<td>F  2? 520??</td>
<td>Breast Cancer</td>
<td></td>
</tr>
<tr>
<td>M  2? 520??</td>
<td>Lung Cancer</td>
<td></td>
</tr>
<tr>
<td>M  3? 520??</td>
<td>Heart Disease</td>
<td></td>
</tr>
<tr>
<td>M  3? 520??</td>
<td>Fever</td>
<td></td>
</tr>
<tr>
<td>F  3? 520??</td>
<td>Nose Pains</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Quasi Identifier</th>
<th>Sensitive</th>
<th>2-diverse</th>
<th>2-diverse</th>
</tr>
</thead>
<tbody>
<tr>
<td>QI 1</td>
<td>Headache</td>
<td>QI 1</td>
<td>Headache</td>
</tr>
<tr>
<td>QI 1</td>
<td>Headache</td>
<td>QI 1</td>
<td>Cancer</td>
</tr>
<tr>
<td>QI 1</td>
<td>Headache</td>
<td>QI 1</td>
<td>Headache</td>
</tr>
<tr>
<td>QI 2</td>
<td>Cancer</td>
<td>QI 2</td>
<td>Headache</td>
</tr>
<tr>
<td>QI 2</td>
<td>Cancer</td>
<td>QI 2</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
/-Diversity

Other Variants

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

Limitations

- Not necessary at times
- Difficult to achieve: For large record size, many equivalent classes will be needed to satisfy /-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 \( \epsilon \)-differential privacy if for any two databases \( D_1 \) and \( D_2 \) that differ on at most one element, and all outputs \( S \subseteq \text{Range}(R) \)

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

\( \epsilon \) is a parameter called \textit{privacy budget/level}. 

2. Basics 2.4 Privacy
Data Perturbation

Data perturbation is achieved by noise addition.

Some Kinds of Noise

- Laplace noise
- Gaussian noise
- Exponential Mechanism