Data Mining and Knowledge Discovery

Part of Jožef Stefan IPS Programme - ICT3 and UL Programme - Statistics

2015 / 2016

Nada Lavrač

Jožef Stefan Institute
Ljubljana, Slovenia
Jožef Stefan Institute and IPS

• Jožef Stefan Institute (JSI, founded in 1949)
  – named after a distinguished physicist Jožef Stefan (1835-1893)
  – leading national research organization in natural sciences and technology (~700 researchers and students)

• JSI research areas
  – information and communication technologies
  – chemistry, biochemistry & nanotechnology
  – physics, nuclear technology and safety

• Jožef Stefan International Postgraduate School (IPS, founded in 2004)
  – offers MSc and PhD programs (ICT, nanotechnology, ecotechnology)
  – research oriented, basic + management courses
  – in English
Jožef Stefan Institute
Department of Knowledge Technologies

• **Head:** Nada Lavrač, **Staff:** 30 researchers, 10 students

• **Machine learning & Data mining**
  – ML (decision tree and rule learning, subgroup discovery, …)
  – Text and Web mining
  – Relational data mining - inductive logic programming
  – Equation discovery

• **Other research areas:**
  – Knowledge management
  – Decision support
  – Human language technologies

• **Applications:**
  – Medicine, Bioinformatics, Public Health
  – Ecology, Finance, …
Course Outline

I. Introduction
   – Data Mining in a Nutshell
   – Predictive and descriptive DM techniques
   – Data Mining and KDD process
   – DM standards, tools and visualization
     (Mladenčić et al. Ch. 1 and 11)

II. Predictive DM Techniques
   – Bayesian classifier
     (Kononenko Ch. 9.6)
   – Decision Tree learning
     (Mitchell Ch. 3, Kononenko Ch. 9.1)
   – Classification rule learning
     (Kononenko Ch. 9.2)
   – Classifier Evaluation
     (Bramer Ch. 6)

III. Regression
     (Kononenko Ch. 9.4)

IV. Descriptive DM
   – Predictive vs. descriptive induction
   – Subgroup discovery
   – Association rule learning
     (Kononenko Ch. 9.3)
   – Hierarchical clustering (Kononenko Ch. 12.3)

V. Relational Data Mining
   – RDM and Inductive Logic Programming (Dzeroski & Lavrac Ch. 3, Ch. 4)
   – Propositionalization approaches
   – Relational subgroup discovery
Part I. Introduction

• Data Mining in a Nutshell
• Predictive and descriptive DM techniques
• Data Mining and the KDD process
• DM standards, tools and visualization
Basic Data Mining Task

Input: transaction data table, relational database, text documents, Web pages
Goal: build a classification model, find interesting patterns in data, ...
Data Mining and Machine Learning

- Machine learning techniques
  - classification rule learning
  - subgroup discovery
  - relational data mining and ILP
  - equation discovery
  - inductive databases

- Data mining and decision support integration

- Data mining applications
  - medicine, health care
  - ecology, agriculture
  - knowledge management, virtual organizations
Relational data mining: domain knowledge = relational database
Semantic data mining: domain knowledge = ontologies
Basic DM and DS Tasks

Data Mining

- Input: transaction data table, relational database, text documents, Web pages
- Goal: build a classification model, find interesting patterns in data, ...

Decision Support

- Input: expert knowledge about data and decision alternatives
- Goal: construct decision support model – to support the evaluation and choice of best decision alternatives
Decision support tools: **DEXi**

- **if-then analysis**
- analysis of stability
- **Time analysis**
- **how** explanation
- **why** explanation
DM and DS integration

Data mining

Decision support

patterns model

data

expert knowledge
Basic Text and Web Mining Task

**Input:** text documents, Web pages

**Goal:** text categorization, user modeling, data visualization...

Knowledge discovery from text data and Web

Text/Web Mining

documents

Web pages

model, patterns, …
Text Mining (lectures by D. Mladenić)

Document-Atlas

SEKTbar

Content-Land

Semantic-Graphs

OntoGen

Contexter
Knowledge Technologies: Main research areas & IPS lectures

ICT3

Knowledge Technologies (AI, Intelligent Systems)

Data Mining and Knowledge Discovery
Lavrač

Computational Scientific Discovery
Džeroski

Human Language Technologies
Erjavec

Multiobjective Optimization
Filipič

Text Mining
Mladenić

Decision Support
Bohanec
videolectures.net portal

~ 10,000 lectures
Selected Publications
Part I. Introduction

Data Mining in a Nutshell

• Predictive and descriptive DM techniques
• Data Mining and the KDD process
• DM standards, tools and visualization
What is DM

• Extraction of useful information from data: discovering relationships that have not previously been known

• The viewpoint in this course: Data Mining is the application of Machine Learning techniques to solve real-life data analysis problems
Data Mining in a Nutshell

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>

Given: transaction data table, relational database, text documents, Web pages
Find: a classification model, a set of interesting patterns

knowledge discovery from data

Data Mining model, patterns, …

data
Data Mining in a Nutshell

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>

**Given:** transaction data table, relational database, text documents, Web pages

**Find:** a classification model, a set of interesting patterns

data

knowledge discovery from data

Data Mining

symbolic model
symbolic patterns
explanation

new unclassified instance

classified instance

black box classifier
no explanation
Simplified example: Learning a classification model from contact lens data

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>
Simplified example: Learning a classification model from contact lens data

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>young</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-013</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>pre-presby</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>pre-presby</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>pre-presby</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>

Data Mining
Binary classes (positive vs. negative examples of Target class)
- for Concept learning – classification and class description
- for Subgroup discovery – exploring patterns characterizing groups of instances of target class

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O6-O13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NO</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NO</td>
</tr>
<tr>
<td>O19-O23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NO</td>
</tr>
</tbody>
</table>
Learning from Numeric Class Data

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>LensPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>0</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>8</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>0</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>5</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>0</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>5</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>0</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>0</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>0</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>0</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>0</td>
</tr>
</tbody>
</table>

Numeric class values – regression analysis
## Learning from Unlabeled Data

Unlabeled data - clustering: grouping of similar instances  
- association rule learning

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>
Data Mining: Related areas

Database technology and data warehouses

- efficient storage, access and manipulation of data

- text and Web mining
- soft computing
- pattern recognition
- visualization
- machine learning
- statistics
- databases
Related areas

Statistics, machine learning, pattern recognition and soft computing*

- classification techniques and techniques for knowledge extraction from data

*neural networks, fuzzy logic, genetic algorithms, probabilistic reasoning
Related areas

Text and Web mining

- Web page analysis
- Text categorization
- Acquisition, filtering and structuring of textual information
- Natural language processing
Related areas

Visualization

- visualization of data and discovered knowledge
Point of view in this course

Knowledge discovery using machine learning methods
Data Mining, ML and Statistics

• All three areas have a long tradition of developing inductive techniques for data analysis.
  – reasoning from properties of a data sample to properties of a population

• DM vs. ML - Viewpoint in this course:
  – Data Mining is the application of Machine Learning techniques to hard real-life data analysis problems
Data Mining, ML and Statistics

- All three areas have a long tradition of developing inductive techniques for data analysis.
  - reasoning from properties of a data sample to properties of a population

- **DM vs. Statistics:**
  - **Statistics**
    - Hypothesis testing when certain theoretical expectations about the data distribution, independence, random sampling, sample size, etc. are satisfied
    - Main approach: best fitting all the available data
  - **Data mining**
    - Automated construction of understandable patterns, and structured models
    - Main approach: structuring the data space, heuristic search for decision trees, rules, … covering (parts of) the data space
Part I. Introduction

• Data Mining in a Nutshell
• Predictive and descriptive DM techniques
• Data Mining and the KDD process
• DM standards, tools and visualization
Types of DM tasks

- **Predictive DM:**
  - Classification (learning of rules, decision trees, ...)
  - Prediction and estimation (regression)
  - Predictive relational DM (ILP)

- **Descriptive DM:**
  - Description and summarization
  - Dependency analysis (association rule learning)
  - Discovery of properties and constraints
  - Segmentation (clustering)
  - Subgroup discovery
Predictive vs. descriptive DM

Predictive DM

Descriptive DM
Predictive vs. descriptive DM

**Predictive DM:** Inducing classifiers for solving classification and prediction tasks,
- Classification rule learning, Decision tree learning, ...
- Bayesian classifier, ANN, SVM, ...
- Data analysis through hypothesis generation and testing

**Descriptive DM:** Discovering interesting regularities in the data, uncovering patterns, ... for solving KDD tasks
- Symbolic clustering, Association rule learning, Subgroup discovery, ...
- Exploratory data analysis
Predictive DM formulated as a machine learning task:

• Given a set of labeled training examples (n-tuples of attribute values, labeled by class name)

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1,1</td>
<td>v1,2</td>
<td>v1,3</td>
<td>C1</td>
</tr>
<tr>
<td>v2,1</td>
<td>v2,2</td>
<td>v2,3</td>
<td>C2</td>
</tr>
</tbody>
</table>

• By performing generalization from examples (induction) find a hypothesis (classification rules, decision tree, …) which explains the training examples, e.g. rules of the form:

\[(A_i = v_{i,k}) \& (A_j = v_{j,l}) \& ... \Rightarrow \text{Class} = C_n\]
Predictive DM - Classification

• data are objects, characterized with attributes - they belong to different classes (discrete labels)
• given objects described with attribute values, induce a model to predict different classes
• decision trees, if-then rules, discriminant analysis, ...
# Data mining example

**Input: Contact lens data**

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>young</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>pre-presbyopic</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>pre-presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>pre-presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>
Contact lens data: Decision tree

**Type of task:** prediction and classification

**Hypothesis language:** decision trees

(nodes: attributes, arcs: values of attributes, leaves: classes)
Contact lens data: Classification rules

Type of task: prediction and classification
Hypothesis language: rules \( X \rightarrow C, \) if \( X \) then \( C \)

\( X \) conjunction of attribute values, \( C \) class

tear production=reduced \( \rightarrow \) lenses=NONE
tear production=normal & astigmatism=yes & spect. pre.=hypermetrope \( \rightarrow \) lenses=NONE
tear production=normal & astigmatism=no \( \rightarrow \) lenses=SOFT

tear production=normal & astigmatism=yes & spect. pre.=myope \( \rightarrow \) lenses=HARD

DEFAULT lenses=NONE
Task reformulation: Concept learning problem (positive vs. negative examples of Target class)

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O2</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O3</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O4</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O5</td>
<td>young</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>pre-presbyopic</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O15</td>
<td>pre-presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O16</td>
<td>pre-presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NO</td>
</tr>
<tr>
<td>O17</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O18</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NO</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NO</td>
</tr>
</tbody>
</table>
Contact lens data:
Classification rules in concept learning

Type of task: prediction and classification

Hypothesis language: rules \( X \rightarrow C \), if \( X \) then \( C \)
\( X \) conjunction of attribute values, \( C \) target class

Target class: yes

- tear production=normal & astigmatism=no \( \rightarrow \) lenses=YES
- tear production=normal & astigmatism=yes & spect. pre.=myope \( \rightarrow \) lenses=YES
- else NO
Illustrative example: Customer data

<table>
<thead>
<tr>
<th>Customer</th>
<th>Gender</th>
<th>Age</th>
<th>Income</th>
<th>Spent</th>
<th>BigSpender</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>male</td>
<td>30</td>
<td>214000</td>
<td>18800</td>
<td>yes</td>
</tr>
<tr>
<td>c2</td>
<td>female</td>
<td>19</td>
<td>139000</td>
<td>15100</td>
<td>yes</td>
</tr>
<tr>
<td>c3</td>
<td>male</td>
<td>55</td>
<td>50000</td>
<td>12400</td>
<td>no</td>
</tr>
<tr>
<td>c4</td>
<td>female</td>
<td>48</td>
<td>26000</td>
<td>8600</td>
<td>no</td>
</tr>
<tr>
<td>c5</td>
<td>male</td>
<td>63</td>
<td>191000</td>
<td>28100</td>
<td>yes</td>
</tr>
<tr>
<td>c14</td>
<td>female</td>
<td>61</td>
<td>95000</td>
<td>18100</td>
<td>yes</td>
</tr>
<tr>
<td>c15</td>
<td>male</td>
<td>56</td>
<td>44000</td>
<td>12000</td>
<td>no</td>
</tr>
<tr>
<td>c16</td>
<td>male</td>
<td>36</td>
<td>102000</td>
<td>13800</td>
<td>no</td>
</tr>
<tr>
<td>c17</td>
<td>female</td>
<td>57</td>
<td>215000</td>
<td>29300</td>
<td>yes</td>
</tr>
<tr>
<td>c18</td>
<td>male</td>
<td>33</td>
<td>67000</td>
<td>9700</td>
<td>no</td>
</tr>
<tr>
<td>c19</td>
<td>female</td>
<td>26</td>
<td>95000</td>
<td>11000</td>
<td>no</td>
</tr>
<tr>
<td>c20</td>
<td>female</td>
<td>55</td>
<td>214000</td>
<td>28800</td>
<td>yes</td>
</tr>
</tbody>
</table>
Customer data: Decision trees

- Income
  - ≤ 102000
    - Age
      - ≤ 58: no
      - > 58: yes
  - > 102000
    - yes

- Gender
  - = female
    - Age
      - ≤ 49: no
      - > 49: yes
  - = male
    - no
Predictive DM - Estimation

- often referred to as regression
- data are objects, characterized with attributes (discrete or continuous), classes of objects are continuous (numeric)
- given objects described with attribute values, induce a model to predict the numeric class value
- regression trees, linear and logistic regression, ANN, kNN, ...
**Estimation/regression example:**

**Customer data**

<table>
<thead>
<tr>
<th>Customer</th>
<th>Gender</th>
<th>Age</th>
<th>Income</th>
<th>Spent</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>male</td>
<td>30</td>
<td>214000</td>
<td>18800</td>
</tr>
<tr>
<td>c2</td>
<td>female</td>
<td>19</td>
<td>139000</td>
<td>15100</td>
</tr>
<tr>
<td>c3</td>
<td>male</td>
<td>55</td>
<td>50000</td>
<td>12400</td>
</tr>
<tr>
<td>c4</td>
<td>female</td>
<td>48</td>
<td>26000</td>
<td>8600</td>
</tr>
<tr>
<td>c5</td>
<td>male</td>
<td>63</td>
<td>191000</td>
<td>28100</td>
</tr>
<tr>
<td>c6-c13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c14</td>
<td>female</td>
<td>61</td>
<td>95000</td>
<td>18100</td>
</tr>
<tr>
<td>c15</td>
<td>male</td>
<td>56</td>
<td>44000</td>
<td>12000</td>
</tr>
<tr>
<td>c16</td>
<td>male</td>
<td>36</td>
<td>102000</td>
<td>13800</td>
</tr>
<tr>
<td>c17</td>
<td>female</td>
<td>57</td>
<td>215000</td>
<td>29300</td>
</tr>
<tr>
<td>c18</td>
<td>male</td>
<td>33</td>
<td>67000</td>
<td>9700</td>
</tr>
<tr>
<td>c19</td>
<td>female</td>
<td>26</td>
<td>95000</td>
<td>11000</td>
</tr>
<tr>
<td>c20</td>
<td>female</td>
<td>55</td>
<td>214000</td>
<td>28800</td>
</tr>
</tbody>
</table>
Customer data: regression tree

In the nodes one usually has Predicted value +- st. deviation

Income

≤ 108000

Age

≤ 42.5

16500

> 42.5

26700

> 108000

12000
Predicting algal biomass: regression tree

Month
- Jan.-June
  - Ptot
    - ≤ 9.34
      - 2.97±1.09
    - > 9.34
      - 4.32±2.07
  - > 9.34
    - > 9.1
      - 1.15±0.21
    - ≤ 9.1
      - 2.08 ±0.71

- July - Dec.
  - Si
    - ≤ 10.1
      - 1.28±1.08
    - > 10.1
      - 2.34±1.65
**Descriptive DM: Subgroup discovery example - Customer data**

<table>
<thead>
<tr>
<th>Customer</th>
<th>Gender</th>
<th>Age</th>
<th>Income</th>
<th>Spent</th>
<th>BigSpender</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>male</td>
<td>30</td>
<td>214000</td>
<td>18800</td>
<td>yes</td>
</tr>
<tr>
<td>c2</td>
<td>female</td>
<td>19</td>
<td>139000</td>
<td>15100</td>
<td>yes</td>
</tr>
<tr>
<td>c3</td>
<td>male</td>
<td>55</td>
<td>50000</td>
<td>12400</td>
<td>no</td>
</tr>
<tr>
<td>c4</td>
<td>female</td>
<td>48</td>
<td>26000</td>
<td>8600</td>
<td>no</td>
</tr>
<tr>
<td>c5</td>
<td>male</td>
<td>63</td>
<td>191000</td>
<td>28100</td>
<td>yes</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>c14</td>
<td>female</td>
<td>61</td>
<td>95000</td>
<td>18100</td>
<td>yes</td>
</tr>
<tr>
<td>c15</td>
<td>male</td>
<td>56</td>
<td>44000</td>
<td>12000</td>
<td>no</td>
</tr>
<tr>
<td>c16</td>
<td>male</td>
<td>36</td>
<td>102000</td>
<td>13800</td>
<td>no</td>
</tr>
<tr>
<td>c17</td>
<td>female</td>
<td>57</td>
<td>215000</td>
<td>29300</td>
<td>yes</td>
</tr>
<tr>
<td>c18</td>
<td>male</td>
<td>33</td>
<td>67000</td>
<td>9700</td>
<td>no</td>
</tr>
<tr>
<td>c19</td>
<td>female</td>
<td>26</td>
<td>95000</td>
<td>11000</td>
<td>no</td>
</tr>
<tr>
<td>c20</td>
<td>female</td>
<td>55</td>
<td>214000</td>
<td>28800</td>
<td>yes</td>
</tr>
</tbody>
</table>
Customer data: Subgroup discovery

Type of task: description (pattern discovery)
Hypothesis language: rules $X \Rightarrow Y$, if $X$ then $Y$

$X$ is conjunctions of items, $Y$ is target class

Age $> 52$ & Sex = male $\Rightarrow$ BigSpender = no

Age $> 52$ & Sex = male & Income $\leq 73250$

$\Rightarrow$ BigSpender = no
## Descriptive DM:
Clustering and association rule learning example - Customer data

<table>
<thead>
<tr>
<th>Customer</th>
<th>Gender</th>
<th>Age</th>
<th>Income</th>
<th>Spent</th>
<th>BigSpender</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>male</td>
<td>30</td>
<td>214000</td>
<td>18800</td>
<td>yes</td>
</tr>
<tr>
<td>c2</td>
<td>female</td>
<td>19</td>
<td>139000</td>
<td>15100</td>
<td>yes</td>
</tr>
<tr>
<td>c3</td>
<td>male</td>
<td>55</td>
<td>50000</td>
<td>12400</td>
<td>no</td>
</tr>
<tr>
<td>c4</td>
<td>female</td>
<td>48</td>
<td>26000</td>
<td>8600</td>
<td>no</td>
</tr>
<tr>
<td>c5</td>
<td>male</td>
<td>63</td>
<td>191000</td>
<td>28100</td>
<td>yes</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>c14</td>
<td>female</td>
<td>61</td>
<td>95000</td>
<td>18100</td>
<td>yes</td>
</tr>
<tr>
<td>c15</td>
<td>male</td>
<td>56</td>
<td>44000</td>
<td>12000</td>
<td>no</td>
</tr>
<tr>
<td>c16</td>
<td>male</td>
<td>36</td>
<td>102000</td>
<td>13800</td>
<td>no</td>
</tr>
<tr>
<td>c17</td>
<td>female</td>
<td>57</td>
<td>215000</td>
<td>29300</td>
<td>yes</td>
</tr>
<tr>
<td>c18</td>
<td>male</td>
<td>33</td>
<td>67000</td>
<td>9700</td>
<td>no</td>
</tr>
<tr>
<td>c19</td>
<td>female</td>
<td>26</td>
<td>95000</td>
<td>11000</td>
<td>no</td>
</tr>
<tr>
<td>c20</td>
<td>female</td>
<td>55</td>
<td>214000</td>
<td>28800</td>
<td>yes</td>
</tr>
</tbody>
</table>
## Descriptive DM: Association rule learning example - Customer data

<table>
<thead>
<tr>
<th>Customer</th>
<th>Gender</th>
<th>Age</th>
<th>Income</th>
<th>Spent</th>
<th>BigSpender</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>male</td>
<td>30</td>
<td>214000</td>
<td>18800</td>
<td>yes</td>
</tr>
<tr>
<td>c2</td>
<td>female</td>
<td>19</td>
<td>139000</td>
<td>15100</td>
<td>yes</td>
</tr>
<tr>
<td>c3</td>
<td>male</td>
<td>55</td>
<td>50000</td>
<td>12400</td>
<td>no</td>
</tr>
<tr>
<td>c4</td>
<td>female</td>
<td>48</td>
<td>26000</td>
<td>8600</td>
<td>no</td>
</tr>
<tr>
<td>c5</td>
<td>male</td>
<td>63</td>
<td>191000</td>
<td>28100</td>
<td>yes</td>
</tr>
<tr>
<td>c14</td>
<td>female</td>
<td>61</td>
<td>95000</td>
<td>18100</td>
<td>yes</td>
</tr>
<tr>
<td>c15</td>
<td>male</td>
<td>56</td>
<td>44000</td>
<td>12000</td>
<td>no</td>
</tr>
<tr>
<td>c16</td>
<td>male</td>
<td>36</td>
<td>102000</td>
<td>13800</td>
<td>no</td>
</tr>
<tr>
<td>c17</td>
<td>female</td>
<td>57</td>
<td>215000</td>
<td>29300</td>
<td>yes</td>
</tr>
<tr>
<td>c18</td>
<td>male</td>
<td>33</td>
<td>67000</td>
<td>9700</td>
<td>no</td>
</tr>
<tr>
<td>c19</td>
<td>female</td>
<td>26</td>
<td>95000</td>
<td>11000</td>
<td>no</td>
</tr>
<tr>
<td>c20</td>
<td>female</td>
<td>55</td>
<td>214000</td>
<td>28800</td>
<td>yes</td>
</tr>
</tbody>
</table>
Customer data: Association rules

Type of task: description (pattern discovery)

Hypothesis language: rules $X \Rightarrow Y$, if $X$ then $Y$

$X, Y$ conjunctions of items

1. Age $> 52$ & BigSpender = no $\Rightarrow$ Sex = male
2. Age $> 52$ & BigSpender = no $\Rightarrow$
   Sex = male & Income $\leq 73250$
3. Sex = male & Age $> 52$ & Income $\leq 73250$ $\Rightarrow$
   BigSpender = no
Predictive vs. descriptive DM: Summary from a rule learning perspective

- **Predictive DM:** Induces **rulesets** acting as classifiers for solving classification and prediction tasks
- **Descriptive DM:** Discovers **individual rules** describing interesting regularities in the data

- **Therefore:** Different goals, different heuristics, different evaluation criteria
Relational Data Mining (Inductive Logic Programming) in a Nutshell

Given: a relational database, a set of tables. sets of logical facts, a graph, ...
Find: a classification model, a set of interesting patterns
Relational Data Mining (ILP)

- Learning from multiple tables
- Complex relational problems:
  - temporal data: time series in medicine, traffic control, ...
  - structured data: representation of molecules and their properties in protein engineering, biochemistry, ...
Relational Data Mining (ILP)

<table>
<thead>
<tr>
<th>customer</th>
<th>ID</th>
<th>Zip</th>
<th>Sex</th>
<th>State</th>
<th>Income</th>
<th>Age</th>
<th>Club</th>
<th>Resp</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3478</td>
<td>34677</td>
<td>m</td>
<td>si</td>
<td>60-70</td>
<td>32</td>
<td>men</td>
<td>nr</td>
<td></td>
</tr>
<tr>
<td>3479</td>
<td>43666</td>
<td>f</td>
<td>ma</td>
<td>80-90</td>
<td>45</td>
<td>nm</td>
<td>re</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>order</th>
<th>Customer ID</th>
<th>Order ID</th>
<th>Store ID</th>
<th>Delivery Mode</th>
<th>Paymt Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3478</td>
<td>2140267</td>
<td>12</td>
<td>...</td>
<td>regular</td>
<td>cash</td>
</tr>
<tr>
<td>3478</td>
<td>3446778</td>
<td>12</td>
<td>...</td>
<td>express</td>
<td>check</td>
</tr>
<tr>
<td>3478</td>
<td>4728386</td>
<td>17</td>
<td>...</td>
<td>regular</td>
<td>check</td>
</tr>
<tr>
<td>3479</td>
<td>3233444</td>
<td>17</td>
<td>...</td>
<td>express</td>
<td>credit</td>
</tr>
<tr>
<td>3479</td>
<td>3475886</td>
<td>12</td>
<td>...</td>
<td>regular</td>
<td>credit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>store</th>
<th>Store ID</th>
<th>Size</th>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12</td>
<td>small</td>
<td>franchise</td>
<td>city</td>
<td>rural</td>
</tr>
<tr>
<td>17</td>
<td>large</td>
<td>indep</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Relational representation of customers, orders and stores.
Relational representation of customers, orders and stores.

<table>
<thead>
<tr>
<th>ID</th>
<th>Zip</th>
<th>Sex</th>
<th>Soc St</th>
<th>Income</th>
<th>Age</th>
<th>Club</th>
<th>Resp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3478</td>
<td>34667</td>
<td>m</td>
<td>si</td>
<td>60-70</td>
<td>32</td>
<td>me</td>
<td>nr</td>
</tr>
<tr>
<td>3479</td>
<td>43666</td>
<td>f</td>
<td>ma</td>
<td>80-90</td>
<td>45</td>
<td>nm</td>
<td>re</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Basic table for analysis
<table>
<thead>
<tr>
<th>ID</th>
<th>Zip</th>
<th>Sex</th>
<th>Soc St</th>
<th>Income</th>
<th>Age</th>
<th>Club</th>
<th>Resp</th>
</tr>
</thead>
<tbody>
<tr>
<td>3478</td>
<td>34667</td>
<td>m</td>
<td>si</td>
<td>60-70</td>
<td>32</td>
<td>me</td>
<td>nr</td>
</tr>
<tr>
<td>3479</td>
<td>43666</td>
<td>f</td>
<td>ma</td>
<td>80-90</td>
<td>45</td>
<td>nm</td>
<td>re</td>
</tr>
</tbody>
</table>

Data table presented as logical facts (Prolog format)

```
customer(Id,Zip,Sex,Soc St,Income,Age,Club,Resp).
```

Prolog facts describing data in Table 2:
```
customer(3478,34667,m,si,60-70,32,me,nr).
customer(3479,43666,f,ma,80-90,45,nm,re).
```

Expressing a property of a relation:
```
customer(_,_,f,_,_,_,_,_).
```
Relational Data Mining (ILP)

Data bases:
- Name of relation $p$
- Attribute of $p$
- $n$-tuple $<v_1, ..., v_n> = \text{row in a relational table}$
- relation $p = \text{set of n-tuples} = \text{relational table}$

Logic programming:
- Predicate symbol $p$
- Argument of predicate $p$
- Ground fact $p(v_1, ..., v_n)$
- Definition of predicate $p$
  - Set of ground facts
  - Prolog clause or a set of Prolog clauses

Example predicate definition:

```
good_customer(C) :-
customer(C,_,female,_,_,_,_,_),
order(C,_,_,_,_,creditcard).
```
Part I. Introduction

• Data Mining in a Nutshell
• Predictive and descriptive DM techniques
• Data Mining and the KDD process
• DM standards, tools and visualization
Data Mining and KDD

• KDD is defined as “the process of identifying valid, novel, potentially useful and ultimately understandable models/patterns in data.” *

• Data Mining (DM) is the key step in the KDD process, performed by using data mining techniques for extracting models or interesting patterns from the data.

KDD Process

KDD process of discovering useful knowledge from data

- KDD process involves several phases:
  - data preparation
  - data mining (machine learning, statistics)
  - evaluation and use of discovered patterns

- Data mining is the key step, but represents only 15%-25% of the entire KDD process
MEDIANA – analysis of media research data

- Questionnaires about journal/magazine reading, watching of TV programs and listening of radio programs, since 1992, about 1200 questions. Yearly publication: frequency of reading/listening/watching, distribution w.r.t. Sex, Age, Education, Buying power,..

- Data for 1998, about 8000 questionnaires, covering lifestyle, spare time activities, personal viewpoints, reading/listening/watching of media (yes/no/how much), interest for specific topics in media, social status

- good quality, “clean” data

- table of n-tuples (rows: individuals, columns: attributes, in classification tasks selected class)
MEDIANA – media research pilot study

- Patterns uncovering regularities concerning:
  - Which other journals/magazines are read by readers of a particular journal/magazine?
  - What are the properties of individuals that are consumers of a particular media offer?
  - Which properties are distinctive for readers of different journals?
- Induced models: description (association rules, clusters) and classification (decision trees, classification rules)
Simplified association rules

Finding profiles of readers of the Delo daily newspaper

1. reads_Marketing_magazine 116 →
   reads_Delo 95 (0.82)
2. reads_Financial_News (Finance) 223 → reads_Delo 180 (0.81)
3. reads_Views (Razgledi) 201 → reads_Delo 157 (0.78)
4. reads_Money (Denar) 197 → reads_Delo 150 (0.76)
5. reads_Vip 181 → reads_Delo 134 (0.74)

Interpretation: Most readers of Marketing magazine, Financial News, Views, Money and Vip read also Delo.
Simplified association rules

1. reads_Sara 332 \(\Rightarrow\) reads_Slovenske novice 211 (0.64)
2. reads_Ljubezenske zgodbe 283 \(\Rightarrow\)
   reads_Slovenske novice 174 (0.61)
3. reads_Dolenjski list 520 \(\Rightarrow\)
   reads_Slovenske novice 310 (0.6)
4. reads_Omama 154 \(\Rightarrow\) reads_Slovenske novice 90 (0.58)
5. reads_Delavska enotnost 177 \(\Rightarrow\)
   reads_Slovenske novice 102 (0.58)

Most of the readers of Sara, Love stories, Dolenjska new, Omama in Workers new read also Slovenian news.
Simplified association rules

1. reads_Sportske novosti 303 \( \rightarrow \)
   reads_Slovenski delnicar 164 (0.54)
2. reads_Sportske novosti 303 \( \rightarrow \)
   reads_Salomonov oglasnik 155 (0.51)
3. reads_Sportske novosti 303 \( \rightarrow \)
   reads_Lady 152 (0.5)

More than half of readers of Sports news reads also Slovenian shareholders magazine, Solomon advertisements and Lady.
Finding reader profiles: decision tree for classifying people into readers and non-readers of a teenage magazine Antena.
Part I. Introduction

- Data Mining in a Nutshell
- Predictive and descriptive DM techniques
- Data Mining and the KDD process
- DM standards, tools and visualization
CRISP-DM

• Cross-Industry Standard Process for DM
• A collaborative, 18-months partially EC founded project started in July 1997
• NCR, ISL (Clementine), Daimler-Benz, OHRA (Dutch health insurance companies), and SIG with more than 80 members
• DM from art to engineering
• Views DM more broadly than Fayyad et al. (actually DM is treated as KDD process):
CRISP Data Mining Process

DM Tasks

- Business Understanding
- Data Understanding
- Data Preparation
- Modelling
- Evaluation
- Deployment
DM tools

Tools (Software) for Data Mining and Knowledge Discovery

Email new submissions and changes to editor@kdnuggets.com

- **Suites** supporting multiple discovery tasks and data preparation
- **Classification** -- for building a classification model
  Approach: Multiple | Decision tree | Rules | Neural network | Bayesian | Other
- **Clustering** - for finding clusters or segments
- **Statistics, Estimation and Regression**
- **Links and Associations** - for finding links, dependency networks, and associations
- **Sequential Patterns** - tools for finding sequential patterns
- **Visualization** - scientific and discovery-oriented visualization
- **Text and Web Mining**
- **Deviation and Fraud Detection**
- **Reporting and Summarization**
- **Data Transformation and Cleaning**
- **OLAP and Dimensional Analysis**
Public DM tools

- WEKA - Waikato Environment for Knowledge Analysis
- KNIME - Konstanz Information Miner
- R – Bioconductor, ...
- Orange, Orange4WS, ClowdFlows
Visualization

• can be used on its own (usually for description and summarization tasks)
• can be used in combination with other DM techniques, for example
  – visualization of decision trees
  – cluster visualization
  – visualization of association rules
  – subgroup visualization
Data visualization: Scatter plot
DB Miner: Association rule visualization
MineSet: Decision tree visualization
Orange: Visual programming and subgroup discovery visualization
Part I: Summary

- KDD is the overall process of discovering useful knowledge in data
  - many steps including data preparation, cleaning, transformation, pre-processing

- Data Mining is the data analysis phase in KDD
  - DM takes only 15%-25% of the effort of the overall KDD process
    - employing techniques from machine learning and statistics

- Predictive and descriptive induction have different goals: classifier vs. pattern discovery

- Many application areas, many powerful tools available
Part II. Predictive DM techniques

• Naive Bayesian classifier
• Decision tree learning
• Classification rule learning
• Classifier evaluation
Bayesian methods

- Bayesian methods – simple but powerful classification methods
  - Based on Bayesian formula
    \[ p(H \mid D) = \frac{p(D \mid H)}{p(D)} p(H) \]

- Main methods:
  - Naive Bayesian classifier
  - Semi-naïve Bayesian classifier
  - Bayesian networks *

* Out of scope of this course
Naïve Bayesian classifier

- Probability of class, for given attribute values

$$p(c_j | v_1...v_n) = p(c_j) \cdot \frac{p(v_1...v_n | c_j)}{p(v_1...v_n)}$$

- For all $C_j$ compute probability $p(C_j)$, given values $v_i$ of all attributes describing the example which we want to classify (assumption: conditional independence of attributes, when estimating $p(C_j)$ and $p(C_j | v_i)$)

$$p(c_j | v_1...v_n) \approx p(c_j) \cdot \prod_i \frac{p(c_j | v_i)}{p(c_j)}$$

- Output $C_{MAX}$ with maximal posterior probability of class:

$$C_{MAX} = \arg \max_{c_j} p(c_j | v_1...v_n)$$
Naïve Bayesian classifier

\[ p(c_j | v_1...v_n) = \frac{p(c_j \cdot v_1...v_n)}{p(v_1...v_n)} = \frac{p(v_1...v_n | c_j) \cdot p(c_j)}{p(v_1...v_n)} = \]

\[ = \prod_i p(v_i | c_j) \cdot p(c_i) \quad \frac{p(c_j)}{p(v_1...v_n)} = \prod_i \frac{p(c_j | v_i) \cdot p(v_i)}{p(c_j)} = \]

\[ = p(c_j) \cdot \prod_i \frac{p(v_i)}{p(v_1...v_n)} \prod_i \frac{p(c_j | v_i)}{p(c_j)} \approx p(c_j) \cdot \prod_i \frac{p(c_j | v_i)}{p(c_j)} \]
Semi-naïve Bayesian classifier

• Naive Bayesian estimation of probabilities (reliable)

\[
p(c_j | v_i) \cdot p(c_j | v_k) \frac{p(c_j)}{p(c_j)}\]

• Semi-naïve Bayesian estimation of probabilities (less reliable)

\[
p(c_j | v_i, v_k) \frac{p(c_j)}{p(c_j)}\]
Probability estimation

• Relative frequency:
\[ p(c_j) = \frac{n(c_j)}{N}, \quad p(c_j | v_i) = \frac{n(c_j, v_i)}{n(v_i)} \quad \text{for } j = 1 \ldots k, \text{ for } k \text{ classes} \]

• Prior probability: Laplace law
\[ p(c_j) = \frac{n(c_j) + 1}{N + k} \]

• m-estimate:
\[ p(c_j) = \frac{n(c_j) + m \cdot p_a(c_j)}{N + m} \]
Probability estimation: intuition

- Experiment with N trials, n successful
- Estimate probability of success of next trial
- **Relative frequency**: n/N
  - reliable estimate when number of trials is large
  - Unreliable when number of trials is small, e.g., 1/1=1

- **Laplace**: (n+1)/(N+2), (n+1)/(N+k), k classes
  - Assumes uniform distribution of classes

- **m-estimate**: (n+m.p_a)/(N+m)
  - Prior probability of success p_a, parameter m
    (weight of prior probability, i.e., number of ‘virtual’ examples)
Explanation of Bayesian classifier

• Based on information theory
  – Expected number of bits needed to encode a message = optimal code length -\log p for a message, whose probability is p (*)

• Explanation based of the sum of information gains of individual attribute values \( v_i \) (Kononenko and Bratko 1991, Kononenko 1993)

\[
- \log( p(c_j | v_1...v_n)) = \\
= - \log( p(c_j)) - \sum_{i=1}^{n} (- \log p(c_j) + \log( p(c_j | v_i))
\]

* log p denotes binary logarithm
Example of explanation of semi-naïve Bayesian classifier

Hip surgery prognosis
Class = no ("no complications", most probable class, 2 class problem)

<table>
<thead>
<tr>
<th>Attribute value</th>
<th>For decision (bit)</th>
<th>Against (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age = 70-80</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td>Sex = Female</td>
<td></td>
<td>-0.19</td>
</tr>
<tr>
<td>Mobility before injury = Fully mobile</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>State of health before injury = Other</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td>Mechanism of injury = Simple fall</td>
<td></td>
<td>-0.08</td>
</tr>
<tr>
<td>Additional injuries = None</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Time between injury and operation &gt; 10 days</td>
<td>0.42</td>
<td>-0.3</td>
</tr>
<tr>
<td>Fracture classification acc. To Garden = Garden III</td>
<td></td>
<td>-0.3</td>
</tr>
<tr>
<td>Fracture classification acc. To Pauwels = Pauwels III</td>
<td></td>
<td>-0.14</td>
</tr>
<tr>
<td>Transfusion = Yes</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td>Antibiotic profilaxies = Yes</td>
<td></td>
<td>-0.32</td>
</tr>
<tr>
<td>Hospital rehabilitation = Yes</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>General complications = None</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td><strong>Combination:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time between injury and examination &lt; 6 hours</td>
<td>0.21</td>
<td></td>
</tr>
<tr>
<td>AND Hospitalization time between 4 and 5 weeks</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Combination:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Therapy = Artroplastic AND anticoagulant therapy = Yes</td>
<td>0.63</td>
<td></td>
</tr>
</tbody>
</table>
Visualization of information gains for/against $C_i$
Naïve Bayesian classifier

• Naïve Bayesian classifier can be used
  – when we have sufficient number of training examples
    for reliable probability estimation

• It achieves good classification accuracy
  – can be used as ‘gold standard’ for comparison with
    other classifiers

• Resistant to noise (errors)
  – Reliable probability estimation
  – Uses all available information

• Successful in many application domains
  – Web page and document classification
  – Medical diagnosis and prognosis, …
## Improved classification accuracy due to using m-estimate

<table>
<thead>
<tr>
<th></th>
<th>Primary tumor</th>
<th>Breast cancer</th>
<th>thyroid</th>
<th>Rheumatology</th>
</tr>
</thead>
<tbody>
<tr>
<td>#instan</td>
<td>339</td>
<td>288</td>
<td>884</td>
<td>355</td>
</tr>
<tr>
<td>#class</td>
<td>22</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>#attrib</td>
<td>17</td>
<td>10</td>
<td>15</td>
<td>32</td>
</tr>
<tr>
<td>#values</td>
<td>2</td>
<td>2.7</td>
<td>9.1</td>
<td>9.1</td>
</tr>
<tr>
<td>majority</td>
<td>25%</td>
<td>80%</td>
<td>56%</td>
<td>66%</td>
</tr>
<tr>
<td>entropy</td>
<td>3.64</td>
<td>0.72</td>
<td>1.59</td>
<td>1.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Relative freq.</th>
<th>m-estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary tumor</td>
<td>48.20%</td>
<td>52.50%</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>77.40%</td>
<td>79.70%</td>
</tr>
<tr>
<td>hepatitis</td>
<td>58.40%</td>
<td>90.00%</td>
</tr>
<tr>
<td>lymphography</td>
<td>79.70%</td>
<td>87.70%</td>
</tr>
</tbody>
</table>
Part II. Predictive DM techniques

• Naïve Bayesian classifier
• Decision tree learning
• Classification rule learning
• Classifier evaluation
### Illustrative example: Contact lenses data

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>young</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>pre-presby</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>pre-presby</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>pre-presby</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>
Decision tree for contact lenses recommendation

1. **tears production**:
   - reduced: **NONE**
   - normal:
     - no: **SOFT**
     - yes:
       - spect. pre.:
         - myope: **HARD**
         - hypermetrope: **NONE**
Decision tree for contact lenses recommendation

- tear prod.
  - reduced
    - NONE [N=12, S+H=0]
  - normal
    - astigmatism
      - no
        - SOFT [S=5, H+N=1]
      - yes
        - spect. pre.
          - myope
            - HARD [H=3, S+N=2]
          - hypermetrope
            - NONE [N=2, S+H=1]
# PlayGolf: Training examples

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayGolf</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Decision tree representation for PlayGolf

- each internal node is a test of an attribute
- each branch corresponds to an attribute value
- each path is a conjunction of attribute values
- each leaf node assigns a classification
Decision tree representation for PlayGolf

Decision trees represent a disjunction of conjunctions of constraints on the attribute values of instances:

\[
\begin{align*}
( \text{Outlook}=\text{Sunny} & \land \text{Humidity}=\text{Normal} ) \\
\lor \quad ( \text{Outlook}=\text{Overcast} ) \\
\lor \quad ( \text{Outlook}=\text{Rain} \land \text{Wind}=\text{Weak} )
\end{align*}
\]
PlayGolf: Other representations

- Logical expression for PlayGolf=Yes:
  - \((\text{Outlook}=\text{Sunny} \land \text{Humidity}=\text{Normal}) \lor (\text{Outlook}=\text{Overcast}) \lor (\text{Outlook}=\text{Rain} \land \text{Wind}=\text{Weak})\)

- Converting a tree to if-then rules
  - **IF** Outlook=\text{Sunny} \land \text{Humidity}=\text{Normal} **THEN** PlayGolf=Yes
  - **IF** Outlook=\text{Overcast} **THEN** PlayGolf=Yes
  - **IF** Outlook=\text{Rain} \land \text{Wind}=\text{Weak} **THEN** PlayGolf=Yes
  - **IF** Outlook=\text{Sunny} \land \text{Humidity}=\text{High} **THEN** PlayGolf=No
  - **IF** Outlook=\text{Rain} \land \text{Wind}=\text{Strong} **THEN** PlayGolf=No
PlayGolf: Using a decision tree for classification

Is Saturday morning OK for playing golf?
Outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong
PlayGolf = No, because Outlook=Sunny $\land$ Humidity=High
Appropriate problems for decision tree learning

- Classification problems: classify an instance into one of a discrete set of possible categories (medical diagnosis, classifying loan applicants, ...)

- Characteristics:
  - instances described by attribute-value pairs (discrete or real-valued attributes)
  - target function has discrete output values (boolean or multi-valued, if real-valued then regression trees)
  - disjunctive hypothesis may be required
  - training data may be noisy (classification errors and/or errors in attribute values)
  - training data may contain missing attribute values
Learning of decision trees

• ID3 (Quinlan 1979), CART (Breiman et al. 1984), C4.5, WEKA, ...
  – create the root node of the tree
  – if all examples from S belong to the same class Cj
    • then label the root with Cj
  – else
    • select the ‘most informative’ attribute $A$ with values $v_1, v_2, \ldots, v_n$
    • divide training set $S$ into $S_1, \ldots, S_n$ according to values $v_1, \ldots, v_n$
    • recursively build sub-trees $T_1, \ldots, T_n$ for $S_1, \ldots, S_n$
Search heuristics in ID3

- Central choice in ID3: Which attribute to test at each node in the tree? The attribute that is most useful for classifying examples.
- Define a statistical property, called **information gain**, measuring how well a given attribute separates the training examples w.r.t their target classification.
- First define a measure commonly used in information theory, called **entropy**, to characterize the (im)purity of an arbitrary collection of examples.
Entropy

- **S** - training set, **C\(_1\),...,C\(_N\)** - classes
- **Entropy** \( E(S) \) – measure of the impurity of training set \( S \)

\[
E(S) = - \sum_{c=1}^{N} p_c \cdot \log_2 p_c
\]

- **Entropy in binary classification problems**

\[
E(S) = - p_+ \log_2 p_+ - p_- \log_2 p_-
\]

\( p_c \) - prior probability of class \( C_c \)

(relative frequency of \( C_c \) in \( S \))
Entropy

- $E(S) = - p_+ \log_2 p_+ - p_- \log_2 p_-$

- The entropy function relative to a Boolean classification, as the proportion $p_+$ of positive examples varies between 0 and 1
Entropy – why?

- **Entropy** $E(S) = \text{expected amount of information (in bits) needed to assign a class to a randomly drawn object in } S \text{ (under the optimal, shortest-length code)}$
- **Why?**
- Information theory: optimal length code assigns $- \log_2 p$ bits to a message having probability $p$
- So, in binary classification problems, the expected number of bits to encode $+$ or $-$ of a random member of $S$ is:

$$p_+ (- \log_2 p_+) + p_-( - \log_2 p_-) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$
PlayGolf: Entropy

- Training set $S$: 14 examples (9 pos., 5 neg.)
- Notation: $S = [9+, 5-]$  
- $E(S) = - p_+ \log_2 p_+ - p_- \log_2 p_-$
- Computing entropy, if probability is estimated by relative frequency

$$E(S) = - \left( \frac{|S_+|}{|S|} \cdot \log \frac{|S_+|}{|S|} \right) - \left( \frac{|S_-|}{|S|} \cdot \log \frac{|S_-|}{|S|} \right)$$

- $E([9+, 5-]) = - (9/14) \log_2(9/14) - (5/14) \log_2(5/14) = 0.940$
PlayGolf: Entropy

- $E(S) = - p_+ \log_2 p_+ - p_- \log_2 p_-$
- $E(9+,5-) = -(9/14) \log_2 (9/14) - (5/14) \log_2 (5/14) = 0.940$

Outlook?
- Sunny
  - {D1,D2,D8,D9,D11} [2+, 3-] $E=0.970$
  - Overcast
    - {D3,D7,D12,D13} [4+, 0-] $E=0$
  - Rain
    - {D4,D5,D6,D10,D14} [3+, 2-] $E=0.970$

Humidity?
- High [3+, 4-] $E=0.985$
- Normal [6+, 1-] $E=0.592$

Wind?
- Weak [6+, 2-] $E=0.811$
- Strong [3+, 3-] $E=1.00$
Information gain search heuristic

• **Information gain** measure is aimed to minimize the number of tests needed for the classification of a new object.

• **Gain(S,A)** – expected reduction in entropy of S due to sorting on A

\[
Gain (S, A) = E(S) - \sum_{v \in \text{Values} (A)} \frac{|S_v|}{|S|} \cdot E(S_v)
\]

• **Most informative** attribute: \( \max \text{Gain}(S,A) \)
Information gain search heuristic

• Which attribute is more informative, A1 or A2?

\[ \text{Gain}(S,A1) = 0.94 - (8/14 \times 0.811 + 6/14 \times 1.00) = 0.048 \]

\[ \text{Gain}(S,A2) = 0.94 - 0 = 0.94 \quad \text{A2 has max Gain} \]
### PlayGolf: Information gain

\[
Gain (S, A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot E(S_v)
\]

- \(Values(Wind) = \{\text{Weak, Strong}\}\)

\[
\begin{align*}
\text{Wind?} & \quad \text{Weak} & \quad \text{Strong} \\
[6+, 2-] & \quad E=0.811 & \quad [3+, 3-] & \quad E=1.00 \\
S &= [9+, 5-], \quad E(S) = 0.940 \\
S_{\text{weak}} &= [6+, 2-], \quad E(S_{\text{weak}}) = 0.811 \\
S_{\text{strong}} &= [3+, 3-], \quad E(S_{\text{strong}}) = 1.0 \\
\text{Gain}(S, Wind) &= E(S) - \frac{8}{14}E(S_{\text{weak}}) - \frac{6}{14}E(S_{\text{strong}}) = 0.940 - \\
&\quad (\frac{8}{14}) \times 0.811 - (\frac{6}{14}) \times 1.0 = \mathbf{0.048}
\end{align*}
\]
PlayGolf: Information gain

• Which attribute is the best?
  – Gain(S, Outlook)=0.246 MAX!
  – Gain(S, Humidity)=0.151
  – Gain(S, Wind)=0.048
  – Gain(S, Temperature)=0.029
PlayGolf: Information gain

• Which attribute should be tested here?
  
  – Gain($S_{\text{sunny}}$, Humidity) = 0.97-(3/5)0-(2/5)0 = 0.970 \textbf{MAX!}

  – Gain($S_{\text{sunny}}$, Temperature) = 0.97-(2/5)0-(2/5)1-(1/5)0 = 0.570

  – Gain($S_{\text{sunny}}$, Wind) = 0.97-(2/5)1-(3/5)0.918 = 0.019
Probability estimates

• **Relative frequency**:
  - problems with small samples

\[
p(Class \mid Cond) = \frac{n(Class \cdot Cond)}{n(Cond)}
\]

\[
[6+,1-] (7) = \frac{6}{7}
\]

\[
[2+,0-] (2) = \frac{2}{2} = 1
\]

• **Laplace estimate**:
  - assumes uniform prior distribution of k classes

\[
p(Class \mid Cond) = \frac{n(Class \cdot Cond) + 1}{n(Cond) + k}
\]

\[
k = 2
\]

\[
[6+,1-] (7) = \frac{6+1}{7+2} = \frac{7}{9}
\]

\[
[2+,0-] (2) = \frac{2+1}{2+2} = \frac{3}{4}
\]
Heuristic search in ID3

• **Search bias:** Search the space of decision trees from simplest to increasingly complex (greedy search, no backtracking, prefer small trees)

• **Search heuristics:** At a node, select the attribute that is most useful for classifying examples, split the node accordingly

• **Stopping criteria:** A node becomes a leaf
  – if all examples belong to same class $C_j$, label the leaf with $C_j$
  – if all attributes were used, label the leaf with the most common value $C_k$ of examples in the node

• **Extension to ID3:** handling noise - tree pruning
Pruning of decision trees

- Avoid overfitting the data by tree pruning
- Pruned trees are
  - less accurate on training data
  - more accurate when classifying unseen data
Sources of imperfection

1. Random errors (noise) in training examples
   - erroneous attribute values
   - erroneous classification
2. Too sparse training examples (incompleteness)
3. Inappropriate/insufficient set of attributes (inexactness)
4. Missing attribute values in training examples
Handling noise – Tree pruning

• Handling imperfect data
  – handling imperfections of type 1-3
    • pre-pruning (stopping criteria)
    • post-pruning / rule truncation
  – handling missing values

• Pruning avoids perfectly fitting noisy data: relaxing the completeness (fitting all +) and consistency (fitting all -) criteria in ID3
Prediction of breast cancer recurrence: Tree pruning

**Degree of malignancy (Degree_of_malign)**
- < 3
- ≥ 3

**Tumor size (Tumor_size)**
- < 15
- ≥ 15
  - Age
    - < 40
      - no_recur 4
      - recurrence 1
    - ≥ 40
      - no_recur 4

**Involved nodes (Involved_nodes)**
- < 3
- ≥ 3
  - no_recur 30
    - recurrence 18
  - no_recur 27
    - recurrence 10
  - no_recur 125
    - recurrence 39
  - no_recur 4

Accuracy and error

- **Accuracy**: percentage of correct classifications
  - on the training set
  - on unseen instances

- **How accurate is a decision tree when classifying unseen instances**
  - An estimate of accuracy on unseen instances can be computed, e.g., by averaging over 4 runs:
    - split the example set into training set (e.g. 70%) and test set (e.g. 30%)
    - induce a decision tree from training set, compute its accuracy on test set

- **Error = 1 - Accuracy**
- **High error may indicate data overfitting**
Overfitting and accuracy

- Typical relation between tree size and accuracy

- Question: how to prune optimally?
Avoiding overfitting

• How can we avoid overfitting?
  – Pre-pruning (forward pruning): stop growing the tree e.g., when data split not statistically significant or too few examples are in a split
  – Post-pruning: grow full tree, then post-prune

  – forward pruning considered inferior (myopic)
  – post pruning makes use of sub trees
How to select the “best” tree

- Measure performance over training data (e.g., pessimistic post-pruning, Quinlan 1993)
- Measure performance over separate validation data set (e.g., reduced error pruning, Quinlan 1987)
  - until further pruning is harmful DO:
    - for each node evaluate the impact of replacing a subtree by a leaf, assigning the majority class of examples in the leaf, if the pruned tree performs no worse than the original over the validation set
    - greedily select the node whose removal most improves tree accuracy over the validation set
- MDL: minimize size(tree)+size(misclassifications(tree))
Selected decision/regression tree learners

- Decision tree learners
  - ID3 (Quinlan 1979)
  - CART (Breiman et al. 1984)
  - Assistant (Cestnik et al. 1987)
  - C4.5 (Quinlan 1993), C5 (See5, Quinlan)
  - J48 (available in WEKA)

- Regression tree learners, model tree learners
  - M5, M5P (implemented in WEKA)
Features of C4.5

- Implemented as part of the WEKA data mining workbench
- Handling noisy data: post-pruning
- Handling incompletely specified training instances: ‘unknown’ values (?)
  - in learning assign conditional probability of value v: \( p(v|C) = \frac{p(v,C)}{p(C)} \)
  - in classification: follow all branches, weighted by prior prob. of missing attribute values
Other features of C4.5

• Binarization of attribute values
  – for continuous values select a boundary value maximally increasing the informativity of the attribute: sort the values and try every possible split (done automatically)
  – for discrete values try grouping the values until two groups remain *

• ‘Majority’ classification in NULL leaf (with no corresponding training example)
  – if an example ‘falls’ into a NULL leaf during classification, the class assigned to this example is the majority class of the parent of the NULL leaf

* the basic C4.5 doesn’t support binarisation of discrete attributes, it supports grouping
Part II. Predictive DM techniques

• Naïve Bayesian classifier
• Decision tree learning
• Classification rule learning
• Classifier evaluation
Rule Learning in a Nutshell

**knowledge discovery from data**

Rule learning

Model: a set of rules
Patterns: individual rules

data

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>young</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>pre-presbyc hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
<td></td>
</tr>
<tr>
<td>O15</td>
<td>pre-presbyc hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>O16</td>
<td>pre-presbyc hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>O17</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>presbyopic hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
<td></td>
</tr>
</tbody>
</table>

**Given:** transaction data table, relational database (a set of objects, described by attribute values)

**Find:** a classification model in the form of a set of rules; or a set of interesting patterns in the form of individual rules
Rule set representation

- Rule base is a disjunctive set of conjunctive rules
- Standard form of rules:
  IF Condition THEN Class
  Class IF Conditions
  Class ← Conditions

  IF Outlook=Sunny \land Humidity=Normal THEN PlayGolf=Yes
  IF Outlook=Overcast THEN PlayGolf=Yes
  IF Outlook=Rain \land Wind=Weak THEN PlayGolf=Yes

- Form of CN2 rules:
  IF Conditions THEN MajClass [ClassDistr]

- Rule base: \{R1, R2, R3, \ldots, DefaultRule\}
Data mining example
Input: Contact lens data

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>young</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>pre-presbyo</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>pre-presbyo</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>pre-presbyo</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>
Contact lens data: Classification rules

**Type of task:** prediction and classification

**Hypothesis language:** rules $X \Rightarrow C$, if $X$ then $C$

$X$ conjunction of attribute values, $C$ class

tear production=reduced $\rightarrow$ lenses=NONE

tear production=normal & astigmatism=yes & spect. pre.=hypermetrope $\rightarrow$ lenses=NONE

tear production=normal & astigmatism=no $\rightarrow$ lenses=SOFT

tear production=normal & astigmatism=yes & spect. pre.=myope $\rightarrow$ lenses=HARD

DEFAULT lenses=NONE
Rule learning

- Two rule learning approaches:
  - Learn decision tree, convert to rules
  - Learn set/list of rules
    - Learning an unordered set of rules
    - Learning an ordered list of rules
- Heuristics, overfitting, pruning
Contact lenses: convert decision tree to an unordered rule set

- Tear production = reduced => Lenses = NONE [S=0, H=0, N=12]
- Tear production = normal & Astigmatism = yes & Spect. pre. = hypermetrope => Lenses = NONE [S=0, H=1, N=2]
- Tear production = normal & Astigmatism = no => Lenses = SOFT [S=5, H=0, N=1]
- Tear production = normal & Astigmatism = yes & Spect. pre. = myope => Lenses = HARD [S=0, H=3, N=2]
- DEFAULT Lenses = NONE

Order independent rule set (may overlap)
Contact lenses: convert decision tree to decision list

IF tear production=reduced THEN lenses=NONE
ELSE /*tear production=normal*/
    IF astigmatism=no THEN lenses=SOFT
    ELSE /*astigmatism=yes*/
        IF spect. pre.=myope THEN lenses=HARD
        ELSE /*spect.pre.=hypermetrope*/
            lenses=NONE

Ordered (order dependent) rule list
Converting decision tree to rules, and rule post-pruning (Quinlan 1993)

• Very frequently used method, e.g., in C4.5 and J48

• Procedure:
  – grow a full tree (allowing overfitting)
  – convert the tree to an equivalent set of rules
  – prune each rule independently of others
  – sort final rules into a desired sequence for use
Concept learning: Task reformulation for rule learning: (pos. vs. neg. examples of Target class)

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O2</td>
<td>young</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O3</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O4</td>
<td>young</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O5</td>
<td>young</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>pre-presbyopic</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O15</td>
<td>pre-presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O16</td>
<td>pre-presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NO</td>
</tr>
<tr>
<td>O17</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O18</td>
<td>presbyopic</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NO</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>presbyopic</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NO</td>
</tr>
</tbody>
</table>
Original covering algorithm
(AQ, Michalski 1969,86)

Given examples of N classes $C_1, \ldots, C_N$
for each class $C_i$
do
  – $E_i := P_i \cup N_i$ ($P_i$ pos., $N_i$ neg.)
  – $\text{RuleBase}(C_i) := \text{empty}$
  – repeat {learn-set-of-rules}
    • learn-one-rule $R$ covering some positive examples and no negatives
    • add $R$ to $\text{RuleBase}(C_i)$
    • delete from $P_i$ all pos. ex. covered by $R$
  – until $P_i = \text{empty}$
Covering algorithm

Positive examples

Negative examples
Covering algorithm

Rule1: Cl=+ ← Cond2 AND Cond3
Covering algorithm

Rule 1: $Cl = + \leftarrow Cond2 \text{ AND } Cond3$

Positive examples

Negative examples
Covering algorithm

Rule1: Cl=+ ← Cond2 AND Cond3

Rule2: Cl=+ ← Cond8 AND Cond6

Positive examples

Negative examples
### PlayGolf: Training examples

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Heuristics for learn-one-rule: PlayGolf example

PlayGolf = yes [9+,5-] (14)
PlayGolf = yes  ← Wind=weak  [6+,2-] (8)
  ← Wind=strong [3+,3-] (6)
  ← Humidity=normal [6+,1-] (7)
  ← …

PlayGolf = yes  ← Humidity=normal
  Outlook=sunny [2+,0-] (2)
  ← …

Estimating rule accuracy (rule precision) with the probability that a covered example is positive
A(Class ← Cond) = p(Class| Cond)

Estimating the probability with the relative frequency of covered pos. ex. / all covered ex.
[6+,1-] (7) = 6/7, [2+,0-] (2) = 2/2 = 1
Probability estimates

• **Relative frequency**:  
  - problems with small samples
    
\[
p(Class \mid Cond) = \frac{n(Class \cdot Cond)}{n(Cond)}
\]

\[
[6+,1-] (7) = 6/7
\]
\[
[2+,0-] (2) = 2/2 = 1
\]

• **Laplace estimate**:  
  - assumes uniform prior distribution of k classes

\[
p(Class \mid Cond) = \frac{n(Class \cdot Cond) + 1}{n(Cond) + k}
\]

\[
[6+,1-] (7) = 6+1 / 7+2 = 7/9
\]
\[
[2+,0-] (2) = 2+1 / 2+2 = 3/4
\]

\[
\]
Learn-one-rule: search heuristics

- Assume a two-class problem
- Two classes (+,-), learn rules for + class (Cl).
- Search for specializations R’ of a rule R = Cl ← Cond from the RuleBase.
- Specialization R’ of rule R = Cl ← Cond has the form R’ = Cl ← Cond & Cond’
- Heuristic search for rules: find the ‘best’ Cond’ to be added to the current rule R, such that rule accuracy is improved, e.g., such that Acc(R’) > Acc(R)
  - where the expected classification accuracy can be estimated as $A(R) = p(Cl|\text{Cond})$
Learn-one-rule: Greedy vs. beam search

• learn-one-rule by greedy general-to-specific search, at each step selecting the `best’ descendant, no backtracking
  – e.g., the best descendant of the initial rule  
    PlayGolf = yes ←
  – is rule PlayGolf = yes ← Humidity=normal

• beam search: maintain a list of k best candidates at each step; descendants (specializations) of each of these k candidates are generated, and the resulting set is again reduced to k best candidates
Learn-one-rule as search: PlayGolf example

PlayGolf = yes
IF

PlayGolf = yes
IF Wind=weak

PlayGolf = yes
IF Wind=strong

PlayGolf = yes
IF Humidity=normal

PlayGolf = yes
IF Humidity=normal, Wind=weak

PlayGolf = yes
IF Humidity=normal, Wind=strong

PlayGolf = yes
IF Humidity=normal, Outlook=sunny

PlayGolf = yes
IF Humidity=normal, Outlook=rain

...
Learn-one-rule as heuristic search: PlayGolf example

- **PlayGolf = yes**
  - **IF Wind=weak**
    - **[6,2]** (8)
  - **IF Wind=strong**
    - **[3,3]** (6)
  - **IF Humidity=normal, Wind=weak**
  - **IF Humidity=normal, Wind=strong**
  - **IF Humidity=normal, Outlook=sunny**
    - **[2,0]** (2)
  - **IF Humidity=normal, Outlook=rain**
    - **[3,4]** (7)
  - **IF Humidity=high**
    - **[6,1]** (7)
  - **IF Humidity=normal**
    - **[9,5]** (14)
What is “high” rule accuracy (rule precision)?

• Rule evaluation measures:
  – aimed at maximizing classification accuracy
  – minimizing Error = 1 - Accuracy
  – avoiding overfitting
• BUT: Rule accuracy/precision should be traded off against the “default” accuracy/precision of the rule $Cl \leftarrow true$
  – 68% accuracy is OK if there are 20% examples of that class in the training set, but bad if there are 80%
• Relative accuracy (relative precision)
  – $RAcc(Cl \leftarrow Cond) = p(Cl | Cond) - p(Cl)$
Weighted relative accuracy

• If a rule covers a single example, its accuracy/precision is either 0% or 100%
  – maximising relative accuracy tends to produce many overly specific rules

• **Weighted relative accuracy**
  \[
  \text{WRAcc}(\text{Cl} \leftarrow \text{Cond}) = p(\text{Cond}) \cdot [p(\text{Cl} | \text{Cond}) - p(\text{Cl})]
  \]

• WRAcc is a fundamental rule evaluation measure:
  – WRAcc can be used if you want to assess both accuracy and significance
  – WRAcc can be used if you want to compare rules with different heads and bodies
Learn-one-rule: search heuristics

- Assume two classes (+,-), learn rules for + class (Cl). Search for specializations of one rule \( R = Cl \leftarrow \text{Cond} \) from RuleBase.

- Expected **classification accuracy**: \( A(R) = p(\text{Cl|Cond}) \)

- **Informativity** (info needed to specify that example covered by Cond belongs to Cl): \( I(R) = -\log_2 p(\text{Cl|Cond}) \)

- **Accuracy gain** (increase in expected accuracy): \( AG(R',R) = p(\text{Cl|Cond'}) - p(\text{Cl|Cond}) \)

- **Information gain** (decrease in the information needed): \( IG(R',R) = \log_2 p(\text{Cl|Cond'}) - \log_2 p(\text{Cl|Cond}) \)

- **Weighted** measures favoring more general rules: \( WAG, WIG \)
  \[ WAG(R',R) = \frac{p(\text{Cond'})}{p(\text{Cond})} \cdot (p(\text{Cl|Cond'}) - p(\text{Cl|Cond})) \]

- **Weighted relative accuracy** trades off coverage and relative accuracy \( WRAcc(R) = p(\text{Cond}).(p(\text{Cl|Cond}) - p(\text{Cl})) \)
Ordered set of rules: if-then-else rules

• rule  Class IF Conditions is learned by first determining Conditions and then Class

• **Notice:** mixed sequence of classes C1, …, Cn in RuleBase

• **But:** ordered execution when classifying a new instance: rules are sequentially tried and the first rule that `fires’ (covers the example) is used for classification

• **Decision list** \{R1, R2, R3, …, D\}: rules Ri are interpreted as **if-then-else** rules

• If no rule fires, then DefaultClass (majority class in \(E_{\text{cur}}\))
Sequential covering algorithm (similar as in Mitchell’s book)

- RuleBase := empty
- E_{cur} := E
- repeat
  - learn-one-rule R
  - RuleBase := RuleBase U R
  - E_{cur} := E_{cur} - \{examples covered and correctly classified by R\} \hspace{1cm} \text{(DELETE ONLY POS. EX.!)}
  - until performance(R, E_{cur}) < \text{ThresholdR}
- RuleBase := sort RuleBase by performance(R,E)
- return RuleBase
Learn ordered set of rules (CN2, Clark and Niblett 1989)

- RuleBase := empty
- $E_{\text{cur}} := E$
- repeat
  - learn-one-rule $R$
  - RuleBase := RuleBase U $R$
  - $E_{\text{cur}} := E_{\text{cur}} \setminus \{\text{all examples covered by } R\}$
    (NOT ONLY POS. EX.!!)
- until performance($R, E_{\text{cur}}$) < ThresholdR
- RuleBase := sort RuleBase by performance($R, E$)
- RuleBase := RuleBase U DefaultRule($E_{\text{cur}}$)
Learn-one-rule: Beam search in CN2

• Beam search in CN2 learn-one-rule algo.: 
  – construct BeamSize of best rule bodies (conjunctive conditions) that are statistically significant 
  – BestBody - min. entropy of examples covered by Body 
  – construct best rule $R := \text{Head} \leftarrow \text{BestBody}$ by adding majority class of examples covered by BestBody in rule Head 

• performance $(R, E_{\text{cur}})$: - Entropy($E_{\text{cur}}$) 
  – performance$(R, E_{\text{cur}}) < \text{Threshold}_R$ (neg. num.) 
  – Why? Ent. $> t$ is bad, Perf. = -Ent $< -t$ is bad
Variations

- Sequential vs. simultaneous covering of data (as in TDIDT): choosing between attribute-values vs. choosing attributes
- Learning rules vs. learning decision trees and converting them to rules
- Pre-pruning vs. post-pruning of rules
- What statistical evaluation functions to use
- Probabilistic classification
Probabilistic classification

- In the ordered case of standard CN2 rules are interpreted in an IF-THEN-ELSE fashion, and the first fired rule assigns the class.
- In the unordered case all rules are tried and all rules which fire are collected. If a clash occurs, a probabilistic method is used to resolve the clash.
- A simplified example:
  1. tear production=reduced => lenses=NONE [S=0,H=0,N=12]
  2. tear production=normal & astigmatism=yes & spect. pre.=hypermetrope => lenses=NONE [S=0,H=1,N=2]
  3. tear production=normal & astigmatism=no => lenses=SOFT [S=5,H=0,N=1]
  4. tear production=normal & astigmatism=yes & spect. pre.=myope => lenses=HARD [S=0,H=3,N=2]
  5. DEFAULT lenses=NONE

Suppose we want to classify a person with normal tear production and astigmatism. Two rules fire: rule 2 with coverage [S=0,H=1,N=2] and rule 4 with coverage [S=0,H=3,N=2]. The classifier computes total coverage as [S=0,H=4,N=4], resulting in probabilistic classification into class H with probability 0.5 and N with probability 0.5. In this case, the clash can not be resolved, as both probabilities are equal.
Part II. Predictive DM techniques

• Naïve Bayesian classifier
• Decision tree learning
• Classification rule learning
• Classifier evaluation
Classifier evaluation

- Accuracy and Error
- n-fold cross-validation
- Confusion matrix
- ROC
Evaluating hypotheses

• Use of induced hypotheses
  – discovery of new patterns, new knowledge
  – classification of new objects

• Evaluating the quality of induced hypotheses
  – Accuracy, Error = 1 - Accuracy
  – classification accuracy on testing examples = percentage of correctly classified instances
    • split the example set into training set (e.g. 70%) to induce a concept, and test set (e.g. 30%) to test its accuracy
    • more elaborate strategies: 10-fold cross validation, leave-one-out, ...
  – comprehensibility (compactness)
  – information contents (information score), significance
A method for accuracy estimation of classifiers

Partition set D into n disjoint, almost equally-sized folds $T_i$ where $U_i T_i = D$

for $i = 1, \ldots, n$ do
  – form a training set out of n-1 folds: $D_i = D \setminus T_i$
  – induce classifier $H_i$ from examples in $D_i$
  – use fold $T_i$ for testing the accuracy of $H_i$

Estimate the accuracy of the classifier by averaging accuracies over 10 folds $T_i$
• Partition

T₁  T₂  T₃  D
Partition

Train

\[ D \setminus T_1 = D_1 \]
\[ D \setminus T_2 = D_2 \]
\[ D \setminus T_3 = D_3 \]
• Partition

\[ D \setminus T_1 = D_1 \]
\[ D \setminus T_2 = D_2 \]
\[ D \setminus T_3 = D_3 \]

• Train
• Partition

• Train

\[ D \setminus T_1 = D_1 \]
\[ D \setminus T_2 = D_2 \]
\[ D \setminus T_3 = D_3 \]

• Test

\[ T_1 \]
\[ T_2 \]
\[ T_3 \]
Confusion matrix and rule (in)accuracy

• Accuracy of a classifier is measured as TP+TN / N.
• Suppose two rules are both 80% accurate on an evaluation dataset, are they always equally good?
  – e.g., Rule 1 correctly classifies 40 out of 50 positives and 40 out of 50 negatives; Rule 2 correctly classifies 30 out of 50 positives and 50 out of 50 negatives
  – on a test set which has more negatives than positives, Rule 2 is preferable;
  – on a test set which has more positives than negatives, Rule 1 is preferable; unless…
  – …the proportion of positives becomes so high that the ‘always positive’ predictor becomes superior!
• Conclusion: classification accuracy is not always an appropriate rule quality measure
Confusion matrix

<table>
<thead>
<tr>
<th>Predicted positive</th>
<th>Predicted negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive examples</td>
<td>True positives</td>
</tr>
<tr>
<td>Negative examples</td>
<td>False negatives</td>
</tr>
<tr>
<td>Positive examples</td>
<td>False positives</td>
</tr>
<tr>
<td>Negative examples</td>
<td>True negatives</td>
</tr>
</tbody>
</table>

- also called *contingency table*

### Classifier 1

<table>
<thead>
<tr>
<th>Predicted positive</th>
<th>Predicted negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive examples</td>
<td>40</td>
</tr>
<tr>
<td>Negative examples</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>50</td>
</tr>
</tbody>
</table>

### Classifier 2

<table>
<thead>
<tr>
<th>Predicted positive</th>
<th>Predicted negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive examples</td>
<td>30</td>
</tr>
<tr>
<td>Negative examples</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicted positive</th>
<th>Predicted negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive examples</td>
<td>20</td>
</tr>
<tr>
<td>Negative examples</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>70</td>
</tr>
</tbody>
</table>

100
**ROC space**

- **True positive rate** = \( \frac{\text{#true pos.}}{\text{#pos.}} \)
  - \( \text{TPr}_1 = \frac{40}{50} = 80\% \)
  - \( \text{TPr}_2 = \frac{30}{50} = 60\% \)

- **False positive rate** = \( \frac{\text{#false pos.}}{\text{#neg.}} \)
  - \( \text{FPr}_1 = \frac{10}{50} = 20\% \)
  - \( \text{FPr}_2 = \frac{0}{50} = 0\% \)

- **ROC space** has
  - FPr on X axis
  - TPr on Y axis
The ROC space
The ROC convex hull
Summary of evaluation

- 10-fold cross-validation is a standard classifier evaluation method used in machine learning.
- ROC analysis is very natural for rule learning and subgroup discovery.
  - can take costs into account
  - here used for evaluation
  - also possible to use as search heuristic
Part III. Numeric prediction

- Baseline
- Linear Regression
- Regression tree
- Model Tree
- kNN
<table>
<thead>
<tr>
<th>Regression</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong>: attribute-value description</td>
<td></td>
</tr>
<tr>
<td><strong>Target variable</strong>:</td>
<td><strong>Target variable</strong>:</td>
</tr>
<tr>
<td>Continuous</td>
<td>Categorical (nominal)</td>
</tr>
<tr>
<td><strong>Evaluation</strong>: cross validation, separate test set, …</td>
<td></td>
</tr>
<tr>
<td><strong>Error</strong>:</td>
<td><strong>Error</strong>:</td>
</tr>
<tr>
<td>MSE, MAE, RMSE, …</td>
<td>1-accuracy</td>
</tr>
<tr>
<td><strong>Algorithms</strong>:</td>
<td><strong>Algorithms</strong>:</td>
</tr>
<tr>
<td>Linear regression, regression trees, …</td>
<td>Decision trees, Naïve Bayes, …</td>
</tr>
<tr>
<td><strong>Baseline predictor</strong>:</td>
<td><strong>Baseline predictor</strong>:</td>
</tr>
<tr>
<td>Mean of the target variable</td>
<td>Majority class</td>
</tr>
</tbody>
</table>
Example

- data about 80 people: Age and Height
Test set

<table>
<thead>
<tr>
<th>Age</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.85</td>
</tr>
<tr>
<td>10</td>
<td>1.4</td>
</tr>
<tr>
<td>35</td>
<td>1.7</td>
</tr>
<tr>
<td>70</td>
<td>1.6</td>
</tr>
</tbody>
</table>
Baseline numeric model

- Average of the target variable

![Graph showing the relationship between age and height, with average predictor indicated.](image-url)
Baseline numeric predictor

- Average of the target variable is 1.63

<table>
<thead>
<tr>
<th>Age</th>
<th>Height</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.85</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>1.7</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>1.6</td>
<td></td>
</tr>
</tbody>
</table>
Linear Regression Model

Height = 0.0056 * Age + 1.4181
Regression tree

Height = 1.3932
LM 1 (5/23.737%)

Height = 1.4025
LM 2 (4/13.419%)

Age
<=4
>=4

Height = 1.4644
LM 3 (8/45.928%)

>=6.5
>6.5

Age
<=12.5
>12.5

Height = 1.7096
LM 4 (63/44.833%)

0 50 100
Age
0 1 1.5 2
Height

Blue: Height
Pink: Prediction
Model tree

Age

<=12.5

LM 1 (17/15.516%)

Height =
0.0333 * Age
+ 1.1366

>12.5

LM 2 (63/44.833%)

Height =
0.0011 * Age
+ 1.6692

Age

Height

Height

Prediction

0 20 40 60 80 100

Age

Height

Height

Prediction

0 20 40 60 80 100

Age
kNN – K nearest neighbors

- Looks at K closest examples (by age) and predicts the average of their target variable
- \( K=3 \)
Which predictor is the best?

<table>
<thead>
<tr>
<th>Age</th>
<th>Height</th>
<th>Baseline</th>
<th>Linear regression</th>
<th>Regression tree</th>
<th>Model tree</th>
<th>kNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.85</td>
<td>1.63</td>
<td>1.43</td>
<td>1.39</td>
<td>1.20</td>
<td>1.01</td>
</tr>
<tr>
<td>10</td>
<td>1.4</td>
<td>1.63</td>
<td>1.47</td>
<td>1.46</td>
<td>1.47</td>
<td>1.51</td>
</tr>
<tr>
<td>35</td>
<td>1.7</td>
<td>1.63</td>
<td>1.61</td>
<td>1.71</td>
<td>1.71</td>
<td>1.67</td>
</tr>
<tr>
<td>70</td>
<td>1.6</td>
<td>1.63</td>
<td>1.81</td>
<td>1.71</td>
<td>1.75</td>
<td>1.81</td>
</tr>
</tbody>
</table>
## Evaluating numeric prediction

<table>
<thead>
<tr>
<th>Performance measure</th>
<th>Formula</th>
</tr>
</thead>
</table>
| mean-squared error                | \[
\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}
\] |
| root mean-squared error           | \[
\sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}}
\] |
| mean absolute error               | \[
\frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{n}
\] |
| relative squared error            | \[
\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(a_1 - \overline{a})^2 + \ldots + (a_n - \overline{a})^2}, \text{ where } \overline{a} = \frac{1}{n} \sum_i a_i
\] |
| root relative squared error       | \[
\sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(a_1 - \overline{a})^2 + \ldots + (a_n - \overline{a})^2}}
\] |
| relative absolute error           | \[
\frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{|a_1 - \overline{a}| + \ldots + |a_n - \overline{a}|}
\] |
| correlation coefficient           | \[
\frac{S_{PA}}{\sqrt{S_p S_A}}, \text{ where } S_{PA} = \frac{\sum_i (p_i - \overline{p})(a_i - \overline{a})}{n-1},
\] |
|                                  | \[
S_p = \frac{\sum_i (p_i - \overline{p})^2}{n-1}, \text{ and } S_A = \frac{\sum_i (a_i - \overline{a})^2}{n-1}
\] |
Course Outline

I. Introduction
- Data Mining in a Nutshell
- Predictive and descriptive DM techniques
- Data Mining and KDD process
- DM standards, tools and visualization
  (Mladenić et al. Ch. 1 and 11)

II. Predictive DM Techniques
- Bayesian classifier
  (Kononenko Ch. 9.6)
- Decision Tree learning
  (Mitchell Ch. 3, Kononenko Ch. 9.1)
- Classification rule learning
  (Kononenko Ch. 9.2)
- Classifier Evaluation
  (Bramer Ch. 6)

III. Regression
  (Kononenko Ch. 9.4)

IV. Descriptive DM
- Predictive vs. descriptive induction
- Subgroup discovery
- Association rule learning
  (Kononenko Ch. 9.3)
- Hierarchical clustering (Kononenko Ch. 12.3)

V. Relational Data Mining
- RDM and Inductive Logic Programming (Dzeroski & Lavrac Ch. 3, Ch. 4)
- Propositionalization approaches
- Relational subgroup discovery
Part IV. Descriptive DM techniques

- Predictive vs. descriptive induction
- Subgroup discovery
- Association rule learning
- Hierarchical clustering
Predictive vs. descriptive induction

- **Predictive induction**: Inducing classifiers for solving classification and prediction tasks,
  - Classification rule learning, Decision tree learning, ...
  - Bayesian classifier, ANN, SVM, ...
  - Data analysis through hypothesis generation and testing

- **Descriptive induction**: Discovering interesting regularities in the data, uncovering patterns, ... for solving KDD tasks
  - Symbolic clustering, Association rule learning, Subgroup discovery, ...
  - Exploratory data analysis
Descriptive DM

- Often used for preliminary explanatory data analysis
- User gets feel for the data and its structure
- Aims at deriving descriptions of characteristics of the data
- Visualization and descriptive statistical techniques can be used
Descriptive DM

• Description
  – Data description and summarization: describe elementary and aggregated data characteristics (statistics, …)
  – Dependency analysis:
    • describe associations, dependencies, …
    • discovery of properties and constraints

• Segmentation
  – Clustering: separate objects into subsets according to distance and/or similarity (clustering, SOM, visualization, …)
  – Subgroup discovery: find unusual subgroups that are significantly different from the majority (deviation detection w.r.t. overall class distribution)
Predictive vs. descriptive induction: A rule learning perspective

- **Predictive induction**: Induces *rulesets* acting as classifiers for solving classification and prediction tasks

- **Descriptive induction**: Discovers *individual rules* describing interesting regularities in the data

- **Therefore**: Different goals, different heuristics, different evaluation criteria
Supervised vs. unsupervised learning: A rule learning perspective

- **Supervised learning**: Rules are induced from labeled instances (training examples with class assignment) - usually used in **predictive induction**

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>
Supervised vs. unsupervised learning: A rule learning perspective

- **Supervised learning:** Rules are induced from labeled instances (training examples with class assignment) - usually used in **predictive induction**
- **Unsupervised learning:** Rules are induced from unlabeled instances (training examples with no class assignment) - usually used in **descriptive induction**

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>
Supervised vs. unsupervised learning: A rule learning perspective

• **Supervised learning:** Rules are induced from labeled instances (training examples with class assignment) - usually used in **predictive induction**

• **Unsupervised learning:** Rules are induced from unlabeled instances (training examples with no class assignment) - usually used in **descriptive induction**

• **Exception: Subgroup discovery**
  DisCOVERS **individual rules** describing interesting regularities in the data from **labeled** examples
### Task reformulation: Binary Class Values

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>hypermetrope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>YES</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NO</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NO</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NO</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NO</td>
</tr>
</tbody>
</table>

**Binary classes (positive vs. negative examples of Target class)**
- for Concept learning – classification and class description
- for Subgroup discovery – exploring patterns characterizing groups of instances of target class
Subgroup Discovery

- A task in which individual interpretable patterns in the form of rules are induced from data, labeled by a predefined property of interest.
- SD algorithms learn several independent rules that describe groups of target class examples
  - subgroups must be large and significant
Part IV. Descriptive DM techniques

- Predictive vs. descriptive induction
- Subgroup discovery
- Association rule learning
- Hierarchical clustering
Subgroup Discovery

Task definition (Kloesgen, Wrobel 1997)

**Given:** a population of individuals and a target class label (the property of individuals we are interested in)

**Find:** population subgroups that are statistically most `interesting`, e.g., are as large as possible and have most unusual statistical (distributional) characteristics w.r.t. the target class (property of interest)
Subgroup interestingness

Interestingness criteria:

– As large as possible
– Class distribution as different as possible from the distribution in the entire data set
– Significant
– Surprising to the user
– Non-redundant
– Simple
– Useful - actionable
Classification versus Subgroup Discovery

- **Classification** *(predictive induction)* - constructing sets of classification rules
  - aimed at learning a model for classification or prediction
  - rules are dependent

- **Subgroup discovery** *(descriptive induction)* – constructing individual subgroup describing rules
  - aimed at finding interesting patterns in target class examples
    - large subgroups *(high target class coverage)*
    - with significantly different distribution of target class examples *(high TP/FP ratio, high significance, high WRAcc)*
  - each rule (pattern) is an independent chunk of knowledge
Classification versus Subgroup discovery
Subgroup discovery task

Task definition for a use case of finding and characterizing population subgroups with high risk for coronary heart disease (CHD)

- **Given**: a population of individuals and a property of interest (target class, e.g. CHD)
- **Find**: `most interesting’ descriptions of population subgroups
  - are as large as possible (high target class coverage)
  - have most unusual distribution of the target property (high TP/FP ratio, high significance)
Subgroup Discovery: Medical Use Case

- Find and characterize population subgroups with high risk for coronary heart disease (CHD) (Gamberger, Lavrač, Krstačić)
- **A1** for males: principal risk factors
  \[ \text{CHD} \leftrightarrow \text{pos. fam. history} \, \& \, \text{age} > 46 \]
- **A2** for females: principal risk factors
  \[ \text{CHD} \leftrightarrow \text{bodyMassIndex} > 25 \, \& \, \text{age} > 63 \]
- **A1, A2** (anamnestic info only), **B1, B2** (an. and physical examination), **C1** (an., phy. and ECG)
- **A1**: supporting factors (found by statistical analysis):
  psychosocial stress, as well as cigarette smoking, hypertension and overweight
Subgroup visualization

Subgroups of patients with CHD risk

[Gamberger, Lavrač & Wettschereck, IDAMAP2002]
Subgroups vs. classifiers

• Classifiers:
  – Classification rules aim at pure subgroups
  – A set of rules forms a domain model

• Subgroups:
  – Rules describing subgroups aim at significantly higher proportion of positives
  – Each rule is an independent chunk of knowledge

• Link
  – SD can be viewed as cost-sensitive classification
  – Instead of \( FN_{cost} \) we aim at increased \( TP_{profit} \)
Classification Rule Learning for Subgroup Discovery: Deficiencies

- Only first few rules induced by the covering algorithm have sufficient support (coverage)
- Subsequent rules are induced from smaller and strongly biased example subsets (pos. examples not covered by previously induced rules), which hinders their ability to detect population subgroups
- ‘Ordered’ rules are induced and interpreted sequentially as a if-then-else decision list
CN2-SD: Adapting CN2 Rule Learning to Subgroup Discovery

- Weighted covering algorithm
- Weighted relative accuracy (WRAcc) search heuristics, with added example weights
- Probabilistic classification
- Evaluation with different interestingness measures
CN2-SD: CN2 Adaptations

- General-to-specific search (beam search) for best rules
- Rule quality measure:
  - CN2: Laplace: \( \text{Acc}(\text{Class} \leftarrow \text{Cond}) = \)
    \[ = \frac{n_c+1}{n_{\text{rule}}+k} \]
  - CN2-SD: Weighted Relative Accuracy
    \( \text{WRAcc}(\text{Class} \leftarrow \text{Cond}) = \)
    \[ p(\text{Cond}) \left( p(\text{Class}|\text{Cond}) - p(\text{Class}) \right) \]
- Weighted covering approach (example weights)
- Significance testing (likelihood ratio statistics)
- Output: Unordered rule sets (probabilistic classification)
CN2-SD: Weighted Covering

- Standard covering approach: covered examples are deleted from current training set

- Weighted covering approach:
  - weights assigned to examples
  - covered pos. examples are re-weighted: in all covering loop iterations, store count i how many times (with how many rules induced so far) a pos. example has been covered: \( w(e,i) \), \( w(e,0)=1 \)
  - Additive weights: \( w(e,i) = 1/(i+1) \)
    - \( w(e,i) \) – pos. example \( e \) being covered \( i \) times
Subgroup Discovery

Positive examples

Negative examples
Subgroup Discovery

Positive examples

Rule1: Cl=+ ← Cond6 AND Cond2

Negative examples
Subgroup Discovery

Positive examples

Negative examples

Rule2: Cl=+ ← Cond3 AND Cond4
Subgroup Discovery

Positive examples

Negative examples
CN2-SD: Weighted WRAcc Search Heuristic

• Weighted relative accuracy (WRAcc) search heuristics, with added example weights

\[
\text{WRAcc}(\text{Cl} \leftarrow \text{Cond}) = p(\text{Cond}) \cdot (p(\text{Cl} | \text{Cond}) - p(\text{Cl}))
\]

  increased coverage, decreased # of rules, approx. equal accuracy (PKDD-2000)

• In WRAcc computation, probabilities are estimated with relative frequencies, adapt:

\[
\text{WRAcc}(\text{Cl} \leftarrow \text{Cond}) = \frac{n'(\text{Cond})}{N'} \cdot \left( \frac{n'(\text{Cl}.\text{Cond})}{n'(\text{Cond})} - \frac{n'(\text{Cl})}{N'} \right)
\]

  – N’ : sum of weights of examples
  – n’(Cond) : sum of weights of all covered examples
  – n’(Cl.Cond) : sum of weights of all correctly covered examples
Subgroup visualization

The CHD task: Find, characterize and visualize population subgroups with high CHD risk (large enough, distributionally unusual, most actionable)
Induced subgroups and their statistical characterization

Subgroup A2 for female patients:

High-CHD-risk IF

body mass index over 25 kg/m² (typically 29)

AND

age over 63 years

Supporting characteristics (computed using \( \chi^2 \) statistical significance test) are: positive family history and hypertension. Women in this risk group typically have slightly increased LDL cholesterol values and normal but decreased HDL cholesterol values.
SD algorithms in the Orange DM Platform

- SD (Gamberger & Lavrač, JAIR 2002)
- APRIORI-SD (Kavšek & Lavrač, AAI 2006)
- CN2-SD (Lavrač et al., JMLR 2004): Adapting CN2 classification rule learner to Subgroup Discovery
  - Weighted covering algorithm
  - Weighted relative accuracy (WRAcc) search heuristics, with added example weights
SD algorithms in Orange and Orange4WS

- **Orange**
  - classification and subgroup discovery algorithms
  - data mining workflows
  - visualization
  - developed at FRI, Ljubljana

- **Orange4WS** (Podpečan 2010)
  - Web service oriented
  - supports workflows and other Orange functionality
  - includes also
    - WEKA algorithms
    - relational data mining
    - semantic data mining with ontologies
  - Web-based platform is under construction
Current platform and workflow developments

- CrowdFlows browser-based DM platform (Kranjc et al. 2012)
- Semantic Subgroup Discovery workflows (Vavpetič et al., 2012)
Part IV. Descriptive DM techniques

- Predictive vs. descriptive induction
- Subgroup discovery
- Association rule learning
- Hierarchical clustering
Association Rule Learning

**Rules:** $X \Rightarrow Y$, if $X$ then $Y$

$X$ and $Y$ are itemsets (records, conjunction of items), where items/features are binary-valued attributes.

**Given:** Transactions

<table>
<thead>
<tr>
<th></th>
<th>i1</th>
<th>i2</th>
<th>...</th>
<th>i50</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>t2</td>
<td>0</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

**Find:** A set of association rules in the form $X \Rightarrow Y$

**Example:** Market basket analysis

beer & coke $\Rightarrow$ peanuts & chips ($0.05$, $0.65$)

- Support: $Sup(X,Y) = \frac{#XY}{#D} = p(XY)$
- Confidence: $Conf(X,Y) = \frac{#XY}{#X} = \frac{Sup(X,Y)}{Sup(X)} = \frac{p(XY)}{p(X)} = p(Y|X)$
Association Rule Learning: Examples

• Market basket analysis
  – beer & coke ⇒ peanuts & chips (5%, 65%)
    (IF beer AND coke THEN peanuts AND chips)
  – Support 5%: 5% of all customers buy all four items
  – Confidence 65%: 65% of customers that buy beer and coke also buy peanuts and chips

• Insurance
  – mortgage & loans & savings ⇒ insurance (2%, 62%)
  – Support 2%: 2% of all customers have all four
  – Confidence 62%: 62% of all customers that have mortgage, loan and savings also have insurance
Association rule learning

- \( X \Rightarrow Y \ldots \) IF \( X \) THEN \( Y \), where \( X \) and \( Y \) are itemsets
- Intuitive meaning: transactions that contain \( X \) tend to contain \( Y \)
- **Items** - binary attributes (features) m,f,headache, muscle pain, arthrotic, arthritic, spondylotic, spondylitic, stiff_less_1_hour
- **Example transactions** – itemsets formed of patient records
  
<table>
<thead>
<tr>
<th>i1</th>
<th>i2</th>
<th>...</th>
<th>...</th>
<th>i50</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Association rules**
  
  - spondylitic \( \Rightarrow \) arthritic & stiff_gt_1_hour \[5\%, 70\%\]
  - arthrotic & spondylotic \( \Rightarrow \) stiff_less_1_hour \[20\%, 90\%\]
Association Rule Learning

**Given:** a set of transactions $D$

**Find:** all association rules that hold on the set of transactions that have

- user defined minimum support, i.e., support $> \text{MinSup}$, and
- user defined minimum confidence, i.e., confidence $> \text{MinConf}$

It is a form of exploratory data analysis, rather than hypothesis verification.
Searching for the associations

- Find all large itemsets
- Use the large itemsets to generate association rules
- If XY is a large itemset, compute
  \[ r = \frac{\text{support}(XY)}{\text{support}(X)} \]
- If \( r > \text{MinConf} \), then \( X \Rightarrow Y \) holds
  (support > MinSup, as XY is large)
Large itemsets

• Large itemsets are itemsets that appear in at least MinSup transaction
• All subsets of a large itemset are large itemsets (e.g., if A,B appears in at least MinSup transactions, so do A and B)
• This observation is the basis for very efficient algorithms for association rules discovery (linear in the number of transactions)
Association vs. Classification rules

- Exploration of dependencies
- Different combinations of dependent and independent attributes
- Complete search (all rules found)

Focused prediction
- Predict one attribute (class) from the others
- Heuristic search (subset of rules found)
Part IV. Descriptive DM techniques

• Predictive vs. descriptive induction
• Subgroup discovery
• Association rule learning
• Hierarchical clustering
Hierarchical clustering

- **Algorithm** (agglomerative hierarchical clustering):
  
  Each instance is a cluster;
  
  repeat
  
  find *nearest* pair $C_i$ in $C_j$;
  
  fuse $C_i$ in $C_j$ in a new cluster
  
  $C_r = C_i \cup C_j$;
  
  determine *dissimilarities* between
  
  $C_r$ and other clusters;
  
  until one cluster left;

- **Dendogram:**
Hierarchical clustering

- Fusing the nearest pair of clusters

- Minimizing intra-cluster similarity
- Maximizing inter-cluster similarity

- Computing the dissimilarities from the “new” cluster
Hierarchical clustering: example

a) sample problem

b) dissimilarity matrix

c) dissimilarity matrix after ‘fusing’ elements x and y

d) dissimilarity matrix after ‘fusing’ elements w and v

e) dissimilarity matrix after ‘fusing’ cluster (x,y) and element z

f) dendrogram
Results of clustering

A dendogram of resistance vectors

Part V: Relational Data Mining

What is RDM

• Propositionalization techniques
• Semantic Data Mining
• Inductive Logic programming
• Learning as search in Inductive Logic Programming
Relational Data Mining (Inductive Logic Programming) in a nutshell

Relational Data Mining

Given: a relational database, a set of tables. sets of logical facts, a graph, ...

Find: a classification model, a set of interesting patterns
Relational Data Mining (ILP)

- Learning from multiple tables
  - patient records connected with other patient and demographic information
- Complex relational problems:
  - temporal data: time series in medicine, ...
  - structured data: representation of molecules and their properties in protein engineering, biochemistry, ...
Sample ILP problem: East-West trains

1. TRAINS GOING EAST

1. 

2. 

3. 

4. 

5. 

2. TRAINS GOING WEST

1. 

2. 

3. 

4. 

5.
Relational data representation

![Diagram of train and objects]

<table>
<thead>
<tr>
<th>CAR</th>
<th>TRAIN</th>
<th>SHAPE</th>
<th>LENGTH</th>
<th>ROOF</th>
<th>WHEELS</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>t1</td>
<td>rectangle</td>
<td>short</td>
<td>none</td>
<td>2</td>
</tr>
<tr>
<td>c2</td>
<td>t1</td>
<td>rectangle</td>
<td>long</td>
<td>none</td>
<td>3</td>
</tr>
<tr>
<td>c3</td>
<td>t1</td>
<td>rectangle</td>
<td>short</td>
<td>peaked</td>
<td>2</td>
</tr>
<tr>
<td>c4</td>
<td>t1</td>
<td>rectangle</td>
<td>long</td>
<td>none</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Relational data representation
Part V: Relational Data Mining

- What is RDM
- Propositionalization techniques
- Semantic Data Mining
- Inductive Logic programming
- Learning as search in Inductive Logic Programming
Propositionalization in a nutshell

Transform a multi-relational (multiple-table) representation to a propositional representation (single table)

Proposed in ILP systems
LINUS (Lavrac et al. 1991, 1994), 1BC (Flach and Lachiche 1999), ...
Propositionalization in a nutshell

Main propositionalization step: first-order feature construction

f1(T):- hasCar(T,C), clength(C,short).

f2(T):- hasCar(T,C), hasLoad(C,L), loadShape(L, circle)

f3(T) :- ....

Propositional learning:

t(T) ← f1(T), f4(T)

Relational interpretation:

eastbound(T) ← hasShortCar(T), hasClosedCar(T).
Relational Data Mining through Propositionalization

Step 1

Propositionalization

Relational representation of customers, orders and stores.
Relational Data Mining through Propositionalization

Step 1

1. constructing relational features
2. constructing a propositional table
Relational Data Mining through Propositionalization

Step 1
Propositionalization

Relational representation of customers, orders and stores.

Step 2
Data Mining

model, patterns, …
Relational Data Mining through Propositionalization

Step 1

Propositionalization

Relational representation of customers, orders and stores.

Step 2

Data Mining

target(A) :-
  'Doctor'(A), 'Italy'(A).

target(A) :-
  'Public'(A), 'Gold'(A).

target(A) :-
  'Poland'(A), 'Deposit'(A), 'Gold'(A).

target(A) :-
  'Germany'(A), 'Insurance'(A).

target(A) :-
  'Service'(A), 'Germany'(A).

patterns (set of rules)
RSD Lessons learned

Efficient propositionalization can be applied to individual-centered, multi-instance learning problems:

- one free global variable (denoting an individual, e.g. molecule M)
- one or more structural predicates: (e.g. has_atom(M,A)), each introducing a new existential local variable (e.g. atom A), using either the global variable (M) or a local variable introduced by other structural predicates (A)
- one or more utility predicates defining properties of individuals or their parts, assigning values to variables

feature121(M):- hasAtom(M,A), atomType(A,21)
feature235(M):- lumo(M,Lu), lessThr(Lu,-1.21)
mutagenic(M):- feature121(M), feature235(M)
Relational Data Mining in Orange4WS

- service for propositionalization through efficient first-order feature construction (Železny and Lavrač, MLJ 2006)
  
  \[ f_{121}(M) \colon \text{hasAtom}(M, A), \text{atomType}(A, 21) \]
  
  \[ f_{235}(M) \colon \text{lumo}(M, \text{Lu}), \text{lessThr}(\text{Lu}, 1.21) \]

- subgroup discovery using CN2-SD
  
  mutagenic(M) ← feature121(M), feature235(M)
Part V: Relational Data Mining

- What is RDM
- Propositionalization techniques
- Inductive Logic programming
- Learning as search in Inductive Logic Programming
What is Semantic Data Mining

- Ontology-driven (semantic) data mining is an emerging research topic
- Semantic Data Mining (SDM) - a new term denoting:
  - the new challenge of mining semantically annotated resources, with ontologies used as background knowledge to data mining
  - approaches with which semantic data are mined
What is Semantic Data Mining

SDM task definition

Given:
- transaction data table, relational database, text documents, Web pages, ...
- one or more domain ontologies

Find: a classification model, a set of patterns
Semantic Data Mining in Orange4WS

• Exploiting semantics in data mining
  – Using domain ontologies as background knowledge for data mining

• Semantic data mining technology: a two-step approach
  – Using propositionalization through first-order feature construction
  – Using subgroup discovery for rule learning

• Implemented in the SEGS algorithm
Using domain ontologies (e.g. Gene Ontology) as background knowledge for Data Mining

Gene Ontology

12093 biological process
1812 cellular components
7459 molecular functions

Joint work with
Igor Trajkovski
and Filip Zelezny
Using domain ontologies (e.g. Gene Ontology) as background knowledge for Data Mining

First-order features, describing gene properties and relations between genes, can be viewed as generalisations of individual genes
First order feature construction

First order features with support > min_support

\[
\begin{align*}
\text{f}(7, A) & : \text{function}(A, 'GO:0046872'). \\
\text{f}(8, A) & : \text{function}(A, 'GO:0004871'). \\
\text{f}(11, A) & : \text{process}(A, 'GO:0007165'). \\
\text{f}(14, A) & : \text{process}(A, 'GO:0044267'). \\
\text{f}(15, A) & : \text{process}(A, 'GO:0050874'). \\
\text{f}(20, A) & : \text{function}(A, 'GO:0004871'), \text{process}(A, 'GO:0050874'). \\
\text{f}(26, A) & : \text{component}(A, 'GO:0016021'). \\
\text{f}(29, A) & : \text{function}(A, 'GO:0046872'), \text{component}(A, 'GO:0016020'). \\
\text{f}(122, A) & : \text{interaction}(A, B), \text{function}(B, 'GO:0004872'). \\
\text{f}(223, A) & : \text{interaction}(A, B), \text{function}(B, 'GO:0004871'), \text{process}(B, 'GO:0009613'). \\
\text{f}(224, A) & : \text{interaction}(A, B), \text{function}(B, 'GO:0016787'), \text{component}(B, 'GO:0043231').
\end{align*}
\]
# Propositionalization

<table>
<thead>
<tr>
<th></th>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
<th>f5</th>
<th>f6</th>
<th>...</th>
<th>...</th>
<th>fn</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>g2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>g3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>g4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>g5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>g1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>g2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>g3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>g4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Propositional learning: subgroup discovery

<table>
<thead>
<tr>
<th></th>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
<th>f5</th>
<th>f6</th>
<th>...</th>
<th>...</th>
<th>fn</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>g2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>g3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>g4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>g5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>g6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>g7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>g8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>g9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

diffexp(A) :- interaction(A,B) & function(B,'GO:0004871')

Over-expressed
IF f2 and f3
[4,0]
Subgroup Discovery

diff. exp. genes

Not diff. exp. genes
Subgroup Discovery

In RSD (using propositional learner CN2-SD):

Quality of the rules = Coverage \times Precision

*Coverage = \text{sum of the covered weights}

*Precision = \text{purity of the covered genes}
RSD naturally uses gene weights in its procedure for repetitive subgroup generation, via its heuristic rule evaluation: weighted relative accuracy.
Semantic Data Mining in two steps

- **Step 1:** Construct relational logic features of genes such as
  \[ \text{interaction}(g, G) \land \text{function}(G, \text{protein\_binding}) \]
  
  \((g \text{ interacts with another gene whose functions include protein binding})\)

  and propositional table construction with features as attributes

- **Step 2:** Using these features to discover and describe subgroups of genes that are differentially expressed (e.g., belong to class DIFF.EXP. of top 300 most differentially expressed genes) in contrast with RANDOM genes (randomly selected genes with low differential expression).

- Sample subgroup description:
  \[ \text{diffexp}(A) : \text{- interaction}(A, B) \land \text{function}(B, \text{GO:0004871}) \land \text{process}(B, \text{GO:0009613}) \]
Summary: SEGS, using the RSD approach

• The SEGS approach enables to discover new medical knowledge from the combination of gene expression data with public gene annotation databases

• The SEGS approach proved effective in several biomedical applications (JBI 2008, …)
  • The work on semantic data mining - using ontologies as background knowledge for subgroup discovery with SEGS - was done in collaboration with I. Trajkovski, F. Železny and J. Tolar

• Recent work: Semantic subgroup discovery implemented in Orange4WS
Semantic subgroup discovery with SEGS

- SEGS workflow is implemented in the Orange4WS data mining environment

- SEGS is also implemented also as a Web applications
  (Trajkovski et al., IEEE TSMC 2008, Trajkovski et al., JBI 2008)
From SEGS to SDM-SEGS: Generalizing SEGS

- SDM-SEGS: a general semantic data mining system generalizing SEGS
- Discovers subgroups both for ranked and labeled data
- Exploits input ontologies in OWL format
- Is also implemented in Orange4WS
Semantic Data Mining

- Semantic subgroup discovery (Vavpetič et al., 2012)
Part V: Relational Data Mining

- What is RDM
- Propositionalization techniques
- Semantic Data Mining
- Inductive Logic programming
- Learning as search in Inductive Logic Programming
Sample ILP problem: Logic programming

\[ E^+ = \{ \text{sort}([2,1,3],[1,2,3]) \} \]
\[ E^- = \{ \text{sort}([2,1],[1]), \text{sort}([3,1,2],[2,1,3]) \} \]

\[ B : \text{definitions of } \text{permutation/2 and } \text{sorted/1} \]

- **Predictive ILP**

  \[ \text{sort}(X,Y) \leftarrow \text{permutation}(X,Y), \text{sorted}(Y). \]

- **Descriptive ILP**

  \[ \text{sorted}(Y) \leftarrow \text{sort}(X,Y). \]
  \[ \text{permutation}(X,Y) \leftarrow \text{sort}(X,Y) \]
  \[ \text{sorted}(X) \leftarrow \text{sort}(X,X) \]
Sample ILP problem: Knowledge discovery

\[ E^+ = \{ \text{daughter}(\text{mary}, \text{ann}), \text{daughter}(\text{eve}, \text{tom}) \} \]
\[ E^- = \{ \text{daughter}(\text{tom}, \text{ann}), \text{daughter}(\text{eve}, \text{ann}) \} \]

\[ B = \{ \text{mother}(\text{ann}, \text{mary}), \text{mother}(\text{ann}, \text{tom}), \text{father}(\text{tom}, \text{eve}), \text{father}(\text{tom}, \text{ian}), \text{female}(\text{ann}), \text{female}(\text{mary}), \text{female}(\text{eve}), \text{male}(\text{pat}), \text{male}(\text{tom}), \text{parent}(X, Y) \leftarrow \text{mother}(X, Y), \text{parent}(X, Y) \leftarrow \text{father}(X, Y) \} \]
Sample relational problem: Knowledge discovery

- $E^+ = \{\text{daughter(mary, ann), daughter(eve, tom)}\}$
  $E^- = \{\text{daughter(tom, ann), daughter(eve, ann)}\}$
- $B = \{\text{mother(ann, mary), mother(ann, tom), father(tom, eve), father(tom, ian), female(ann), female(mary), female(eve), male(pat), male(tom), parent(X, Y) \leftarrow mother(X, Y), parent(X, Y) \leftarrow father(X, Y)}\}$

- **Predictive ILP** - Induce a definite clause
  
  daughter(X, Y) $\leftarrow$ female(X), parent(Y, X).

  or a set of definite clauses

  daughter(X, Y) $\leftarrow$ female(X), mother(Y, X).
  daughter(X, Y) $\leftarrow$ female(X), father(Y, X).

- **Descriptive ILP** - Induce a set of (general) clauses

  $\leftarrow$ daughter(X, Y), mother(X, Y).
  female(X)$\leftarrow$ daughter(X, Y).
  mother(X, Y); father(X, Y) $\leftarrow$ parent(X, Y).
Basic Relational Data Mining and ILP learning tasks

Predictive RDM

Descriptive RDM
Predictive ILP

**Given:**
- A set of observations
  - positive examples $E^+$
  - negative examples $E^-$
- background knowledge $B$
- hypothesis language $L_H$
- covers relation

**Find:**
A hypothesis $H \in L_H$, such that (given $B$) $H$ covers all positive and no negative examples

**In logic, find $H$ such that**
- $\forall e \in E^+: B \land H \models e$ ($H$ is complete)
- $\forall e \in E^- : B \land H \not\models e$ ($H$ is consistent)

**In ILP, $E$ are ground facts, $B$ and $H$ are (sets of) definite clauses**
Predictive ILP

• **Given:**
  - A set of observations
    - positive examples $E^+$
    - negative examples $E^-$
  - background knowledge $B$
  - hypothesis language $L_H$
  - covers relation
  - quality criterion

• **Find:**
A hypothesis $H \in L_H$, such that (given $B$) $H$ is optimal w.r.t. some quality criterion, e.g., max. predictive accuracy $A(H)$

*(instead of finding a hypothesis $H \in L_H$, such that (given $B$) $H$ covers all positive and no negative examples)*
Descriptive ILP

• **Given:**
  – A set of observations (positive examples $E^+$)
  – background knowledge $B$
  – hypothesis language $L_H$
  – covers relation

• **Find:**
  Maximally specific hypothesis $H \in L_H$, such that (given $B$) $H$ covers all positive examples

• In logic, find $H$ such that $\forall c \in H$, $c$ is true in some preferred model of $B \cup E$ (e.g., least Herbrand model $M(B \cup E)$)

• In ILP, $E$ are ground facts, $B$ are (sets of) general clauses
Sample problem: East-West trains

1. TRAINS GOING EAST

   1.
   2.
   3.
   4.
   5.

2. TRAINS GOING WEST

   1.
   2.
   3.
   4.
   5.
RDM knowledge representation (database)

### LOAD_TABLE

<table>
<thead>
<tr>
<th>LOAD</th>
<th>CAR</th>
<th>OBJECT</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>l1</td>
<td>c1</td>
<td>circle</td>
<td>1</td>
</tr>
<tr>
<td>l2</td>
<td>c2</td>
<td>hexagon</td>
<td>1</td>
</tr>
<tr>
<td>l3</td>
<td>c3</td>
<td>triangle</td>
<td>1</td>
</tr>
<tr>
<td>l4</td>
<td>c4</td>
<td>rectangle</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### TRAIN_TABLE

<table>
<thead>
<tr>
<th>CAR</th>
<th>TRAIN</th>
<th>SHAPE</th>
<th>LENGTH</th>
<th>ROOF</th>
<th>WHEELS</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>t1</td>
<td>rectangle</td>
<td>short</td>
<td>none</td>
<td>2</td>
</tr>
<tr>
<td>c2</td>
<td>t1</td>
<td>rectangle</td>
<td>long</td>
<td>none</td>
<td>3</td>
</tr>
<tr>
<td>c3</td>
<td>t1</td>
<td>rectangle</td>
<td>short</td>
<td>peaked</td>
<td>2</td>
</tr>
<tr>
<td>c4</td>
<td>t1</td>
<td>rectangle</td>
<td>long</td>
<td>none</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### CAR_TABLE

CAR  | TRAIN | SHAPE    | LENGTH | ROOF   | WHEELS |
-----|-------|----------|--------|--------|--------|
  c1 | t1    | rectangle| short  | none   | 2      |
  c2 | t1    | rectangle| long   | none   | 3      |
  c3 | t1    | rectangle| short  | peaked | 2      |
  c4 | t1    | rectangle| long   | none   | 2      |
| ... | ...   | ...      | ...    | ...    | ...    |
ER diagram for East-West trains

- **Train**
  - Direction
  - Has
  - Car
    - Shape
    - Length
    - Roof
    - Wheels
  - Load
    - Number
    - Object

**Relationships**:
- 1:
- M:
- 1:
- 1:
ILP representation:

Da

- Example:
  eastbound(t1).

- Background theory:
  car(t1,c1).
  rectangle(c1).
  short(c1).
  none(c1).
  two_wheels(c1).
  load(c1,l1).
  circle(l1).
  one_load(l1).

  car(t1,c2).
  rectangle(c2).
  long(c2).
  none(c2).
  three_wheels(c2).
  load(c2,l2).
  hexagon(l2).
  one_load(l2).

  car(t1,c3).
  rectangle(c3).
  short(c3).
  peaked(c3).
  two_wheels(c3).
  load(c3,l3).
  triangle(l3).
  one_load(l3).

  car(t1,c4).
  rectangle(c4).
  long(c4).
  none(c4).
  two_wheels(c4).
  load(c4,l4).
  rectangle(l4).
  three_loads(l4).

- Hypothesis (predictive ILP):
  eastbound(T) :- car(T,C),short(C),not none(C).
ILP representation: Datalog

- Example:
  eastbound(t1):-
  car(t1,c1),rectangle(c1),short(c1),none(c1),two_wheels(c1),
  load(c1,l1),circle(l1),one_load(l1),
  car(t1,c2),rectangle(c2),long(c2),none(c2),three_wheels(c2),
  load(c2,l2),hexagon(l2),one_load(l2),
  car(t1,c3),rectangle(c3),short(c3),peaked(c3),two_wheels(c3),
  load(c3,l3),triangle(l3),one_load(l3),
  car(t1,c4),rectangle(c4),long(c4),none(c4),two_wheels(c4),
  load(c4,l4),rectangle(l4),three_load(l4).

- Background theory: empty

- Hypothesis:
  eastbound(T):-car(T,C),short(C),not none(C).
ILP representation

• Example:
  
  eastbound([c(rectangle,
              c(rectangle,long,none,3,l(hexagon,1)),
              c(rectangle,short,peaked,2,l(triangle,1)),
              c(rectangle,long,none,2,l(rectangle,3))]).

• Background theory: member/2, arg/3

• Hypothesis:
  
  eastbound(T):-member(C,T),arg(2,C,short), not arg(3,C,none).
Propositionalization in ILP (LINUS)

- Example: learning family relationships

<table>
<thead>
<tr>
<th>Training examples</th>
<th>Background knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>daughter(sue,eve). (+)</td>
<td>parent(eve,sue). female(ann).</td>
</tr>
<tr>
<td>daughter(ann,pat). (+)</td>
<td>parent(ann,tom). female(sue).</td>
</tr>
<tr>
<td>daughter(tom,ann). (-)</td>
<td>parent(pat,ann). female(eve).</td>
</tr>
<tr>
<td>daughter(eve,ann). (-)</td>
<td>parent(tom,sue).</td>
</tr>
</tbody>
</table>

- Transformation to propositional form:

<table>
<thead>
<tr>
<th>Class</th>
<th>Variables</th>
<th>Propositional features</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
<td>f(X)</td>
</tr>
<tr>
<td>θ</td>
<td>sue</td>
<td>eve</td>
</tr>
<tr>
<td>θ</td>
<td>ann</td>
<td>pat</td>
</tr>
<tr>
<td>θ</td>
<td>tom</td>
<td>ann</td>
</tr>
<tr>
<td>θ</td>
<td>eve</td>
<td>ann</td>
</tr>
</tbody>
</table>

- Result of propositional rule learning:
  \[ \text{Class} = \theta \text{ if } (\text{female}(X) = \text{true}) \land (\text{parent}(Y,X) = \text{true}) \]

- Transformation to program clause form:
  \[ \text{daughter}(X,Y) \leftarrow \text{female}(X),\text{parent}(Y,X) \]
First-order feature construction

• All the expressiveness of ILP is in the features
• Given a way to construct (or choose) first-order features, body construction in ILP becomes propositional
  – idea: learn non-determinate clauses with LINUS by saturating background knowledge (performing systematic feature construction in a given language bias)
Declarative bias for first-order feature construction

- In ILP, features involve interactions of local variables
- Features should define properties of individuals (e.g. trains, molecules) or their parts (e.g., cars, atoms)
- Feature construction in LINUS, using the following language bias:
  - one free global variable (denoting an individual, e.g. train)
  - one or more structural predicates: (e.g., has_car(T,C)), each introducing a new existential local variable (e.g. car, atom), using either the global variable (train, molecule) or a local variable introduced by other structural predicates (car, load)
  - one or more utility predicates defining properties of individuals or their parts: no new variables, just using variables
  - all variables should be used
  - parameter: max. number of predicates forming a feature
Sample first-order features

• The following rule has two features ‘has a short car’ and ‘has a closed car’:
  
  eastbound(T):-hasCar(T,C1),clength(C1,short),
  hasCar(T,C2),not croof(C2,none).

• The following rule has one feature ‘has a short closed car’:
  
  eastbound(T):-hasCar(T,C),clength(C,short),
  not croof(C,none).

• Equivalent representation:
  
  eastbound(T):-hasShortCar(T),hasClosedCar(T).

  hasShortCar(T):-hasCar(T,C),clength(C,short).
  hasClosedCar(T):-hasCar(T,C),not croof(C,none).
LINUS revisited

- Standard LINUS:
  - transforming an ILP problem to a propositional problem
  - apply background knowledge predicates

- Revisited LINUS:
  - Systematic first-order feature construction in a given language bias

- Too many features?
  - use a relevancy filter (Gamberger and Lavrac)
LINUS revisited:
Example: East-West trains

Rules induced by CN2, using 190 first-order features with up to two utility predicates:

eastbound(T):-
  hasCarHasLoadSingleTriangle(T),
  not hasCarLongJagged(T),
  not hasCarLongHasLoadCircle(T).

westbound(T):-
  not hasCarEllipse(T),
  not hasCarShortFlat(T),
  not hasCarPeakedTwo(T).

Meaning:

eastbound(T):-
  hasCar(T,C1),hasLoad(C1,L1),lshape(L1,tria),lnumber(L1,1),
  not (hasCar(T,C2),clength(C2,long),croof(C2,jagged)),
  not (hasCar(T,C3),hasLoad(C3,L3),clength(C3,long),lshape(L3,circ)).

westbound(T):-
  not (hasCar(T,C1),cshape(C1,ellipse)),
  not (hasCar(T,C2),clength(C2,short),croof(C2,flat)),
  not (hasCar(T,C3),croof(C3,peak),cwheels(C3,2)).
Relational Data Mining in Orange4WS and ClowdFlows

- service for propositionalization through efficient first-order feature construction (Železny and Lavrač, MLJ 2006)

  \[ f_{121}(M) : - \text{hasAtom}(M,A), \text{atomType}(A,21) \]

  \[ f_{235}(M) : - \text{lumo}(M,Lu), \text{lessThr}(Lu,1.21) \]

- subgroup discovery using CN2-SD

  \[ \text{mutagenic}(M) \leftarrow \text{feature121}(M), \text{feature235}(M) \]
Part V: Relational Data Mining

- What is RDM
- Propositionalization techniques
- Semantic Data Mining
- Inductive Logic programming

Learning as search in Inductive Logic Programming
Learning as search

• **Structuring the state space**: Representing a partial order of hypotheses (e.g. rules) as a graph
  – nodes: concept descriptions (hypotheses/rules)
  – arcs defined by specialization/generalization operators: an arc from parent to child exists if-and-only-if parent is a proper most specific generalization of child

• **Specialization operators**: e.g., adding conditions:
  \[ s(A=a_2 \land B=b_1) = \{A=a_2 \land B=b_1 \land D=d_1, A=a_2 \land B=b_1 \land D=d_2 \} \]

• **Generalization operators**: e.g., dropping conditions:
  \[ g(A=a_2 \land B=b_1) = \{A=a_2, B=b_1\} \]

• **Partial order of hypotheses defines a lattice** (called a refinement graph)
Learn-one-rule as search - Structuring the hypothesis space: PlayGolf example

PlayGolf = yes
IF

PlayGolf = yes
IF Wind=weak

PlayGolf = yes
IF Wind=strong

PlayGolf = yes
IF Humidity=normal

PlayGolf = yes
IF Humidity=high

PlayGolf = yes
IF Humidity=normal, Wind=weak

PlayGolf = yes
IF Humidity=normal, Wind=strong

PlayGolf = yes
IF Humidity=normal, Outlook=sunny

PlayGolf = yes
IF Humidity=normal, Outlook=rain
Learn-one-rule as heuristic search: PlayGolf example

PlayGolf = yes  IF [9+,5−] (14)

PlayGolf = yes  IF Wind=weak
[6+,2−] (8)

PlayGolf = yes  IF Wind=strong
[3+,3−] (6)

PlayGolf = yes  IF Humidity=normal
[6+,1−] (7)

PlayGolf = yes  IF Humidity=high
[3+,4−] (7)

PlayGolf = yes  IF Humidity=normal, Wind=weak

PlayGolf = yes  IF Humidity=normal, Wind=strong

PlayGolf = yes  IF Humidity=normal, Outlook=sunny
[2+,0−] (2)
Learning as search
(Mitchell’s version space model)

• Hypothesis language $L_H$ defines the state space
• How to structure the hypothesis space $L_H$?
• How to move from one hypothesis to another?

• The version space: region between $S$ (maximally specific) and $G$ (maximally general) complete and consistent concept descriptions

```
• too general

• too specific

• more general

• more specific

complete and consistent
```
Learning as search

- Search/move by applying generalization and specialization

- **Prune generalizations:**
  - if $H$ covers example $e$ then all generalizations of $H$ will also cover $e$ (prune using neg. ex.)

- **Prune specializations:**
  - if $H$ does not cover example $e$, no specialization will cover $e$ (prune using if $H$ pos. ex.)
Learning as search: Learner’s ingredients

- structure of the search space (specialization and generalization operators)

- search strategy
  - depth-first
  - breath-first
  - heuristic search (best first, hill-climbing, beam search)

- search heuristics
  - measure of attribute ‘informativity’
  - measure of ‘expected classification accuracy’ (relative frequency, Laplace estimate, m-estimate), ...

- stopping criteria (consistency, completeness, statistical significance, …)
Learn-one-rule: search heuristics

- Assume a two-class problem
- Two classes (+,-), learn rules for + class (Cl).
- Search for specializations $R'$ of a rule $R = Cl \leftarrow Cond$ from the RuleBase.
- Specializarion $R'$ of rule $R = Cl \leftarrow Cond$
  has the form $R' = Cl \leftarrow Cond & Cond'$
- Heuristic search for rules: find the ‘best’ $Cond'$ to be added to the current rule $R$, such that rule accuracy is improved, e.g., such that $Acc(R') > Acc(R)$
  - where the expected classification accuracy can be estimated as $A(R) = p(Cl|Cond)$
Learn-one-rule – Search strategy: Greedy vs. beam search

- learn-one-rule by greedy general-to-specific search, at each step selecting the `best’ descendant, no backtracking
  - e.g., the best descendant of the initial rule
    PlayGolf = yes ←
  - is rule PlayGolf = yes ← Humidity=normal

- beam search: maintain a list of k best candidates at each step; descendants (specializations) of each of these k candidates are generated, and the resulting set is again reduced to k best candidates
ILP as search of program clauses

• An ILP learner can be described by
  – the **structure of the space of clauses**
    • based on the generality relation
    • Let C and D be two clauses. C is more general than D (C ⊨ D) iff
      covers(D) ⊆ covers(C)
    • Example: p(X,Y) ← r(Y,X) is more general than
      p(X,Y) ← r(Y,X), q(X)
  – its **search strategy**
    • uninformed search (depth-first, breadth-first, iterative deepening)
    • heuristic search (best-first, hill-climbing, beam search)
  – its **heuristics**
    • for directing search
    • for stopping search (quality criterion)
ILP as search of program clauses

- **Semantic generality**
  Hypothesis $H_1$ is semantically more general than $H_2$ w.r.t. background theory $B$ if and only if $B \cup H_1 \models H_2$

- **Syntactic generality or $\theta$-subsumption**
  (most popular in ILP)
  - Clause $c_1 \theta$-subsumes $c_2$ ($c_1 \geq \theta c_2$) if and only if $\exists \theta : c_1 \theta \subseteq c_2$
  - Hypothesis $H_1 \geq \theta H_2$ if and only if $\forall c_2 \in H_2$ exists $c_1 \in H_1$ such that $c_1 \geq \theta c_2$

- **Example**
  
  \begin{align*}
  c_1 &= \text{daughter}(X,Y) \leftarrow \text{parent}(Y,X) \\
  c_2 &= \text{daughter}(\text{mary},\text{ann}) \leftarrow \text{female}(\text{mary}), \\
  & \quad \text{parent}(\text{ann},\text{mary}), \\
  & \quad \text{parent}(\text{ann},\text{tom}).
  \end{align*}

  $c_1 \theta$-subsumes $c_2$ under $\theta = \{X/\text{mary}, Y/\text{ann}\}$
The role of subsumption in ILP

• Generality ordering for hypotheses
• Pruning of the search space:
  – generalization
    • if C covers a neg. example then its generalizations need not be considered
  – specialization
    • if C doesn’t cover a pos. example then its specializations need not be considered
• Top-down search of refinement graphs
• Bottom-up search of the hypo. space by
  – building least general generalizations, and
  – inverting resolutions
Structuring the hypothesis space

flies(X) \leftarrow \text{bird(X)}

flies(X) \leftarrow \text{bird(X), normal(X)}

too general

too specific

more general

more specific
Two strategies for learning

• General-to-specific
  – if Θ-subsumption is used then refinement operators

• Specific-to-general search
  – if Θ-subsumption is used then lgg-operator or generalization operator
ILP as search of program clauses

- Two strategies for learning
  - Top-down search of refinement graphs
  - Bottom-up search
    - building least general generalizations
    - inverting resolution (CIGOL)
    - inverting entailment (PROGOL)
More general (induction)

More specific
Generality ordering of clauses

<table>
<thead>
<tr>
<th>Training examples</th>
<th>Background knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>daughter(mary,ann). ⊕</td>
<td>parent(ann,mary).</td>
</tr>
<tr>
<td>daughter(eve,tom). ⊕</td>
<td>parent(ann,tom).</td>
</tr>
<tr>
<td>daughter(tom,ann). ⊗</td>
<td>parent(tom,eve).</td>
</tr>
<tr>
<td>daughter(eve,ann). ⊗</td>
<td>parent(tom,ian).</td>
</tr>
<tr>
<td>daughter(X,Y) ← X=Y</td>
<td></td>
</tr>
<tr>
<td>daughter(X,Y) ← female(X)</td>
<td></td>
</tr>
<tr>
<td>daughter(X,Y) ← parent(Y,X)</td>
<td></td>
</tr>
<tr>
<td>daughter(X,Y) ← parent(X,Z)</td>
<td></td>
</tr>
</tbody>
</table>

Part of the refinement graph for the family relations problem.
Greedy search of the best clause

<table>
<thead>
<tr>
<th>Training examples</th>
<th>Background knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>daughter(mary,ann). ⊕ parent(ann,mary).</td>
<td>female(ann.).</td>
</tr>
<tr>
<td>daughter(eve,tom). ⊕ parent(ann,tom).</td>
<td>female(mary).</td>
</tr>
<tr>
<td>daughter(tom,ann). ⊖ parent(tom,eve).</td>
<td>female(eve).</td>
</tr>
<tr>
<td>daughter(eve,ann). ⊖ parent(tom,ian).</td>
<td></td>
</tr>
</tbody>
</table>
FOIL

- **Language:** function-free normal programs
  - recursion, negation, new variables in the body, no functors, no constants (original)
- **Algorithm:** covering
- **Search heuristics:** weighted info gain
- **Search strategy:** hill climbing
- **Stopping criterion:** encoding length restriction
- **Search space reduction:** types, in/out modes
determinate literals
- **Ground background knowledge, extensional coverage**
- **Implemented in C**
Part V: Summary

- RDM extends DM by allowing multiple tables describing structured data
- Complexity of representation and therefore of learning is determined by one-to-many links
- Many RDM problems are individual-centred and therefore allow strong declarative bias
Advanced Topics

Text mining: An introduction

- Document clustering and outlier detection
- Wordification approach to relational data mining
### Background: Data mining

**Given:** transaction data table, a set of text documents, ...

**Find:** a classification model, a set of interesting patterns

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Spect. presc.</th>
<th>Astigm.</th>
<th>Tear prod.</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>17</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O2</td>
<td>23</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O3</td>
<td>22</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O4</td>
<td>27</td>
<td>myope</td>
<td>yes</td>
<td>normal</td>
<td>HARD</td>
</tr>
<tr>
<td>O5</td>
<td>19</td>
<td>myope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O6-O13</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>35</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>43</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>39</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O17</td>
<td>54</td>
<td>myope</td>
<td>no</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O18</td>
<td>62</td>
<td>myope</td>
<td>no</td>
<td>normal</td>
<td>NONE</td>
</tr>
<tr>
<td>O19-O23</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>56</td>
<td>hypermetrope</td>
<td>yes</td>
<td>normal</td>
<td>NONE</td>
</tr>
</tbody>
</table>
Data mining: Task reformulation

<table>
<thead>
<tr>
<th>Person</th>
<th>Young</th>
<th>Myope</th>
<th>Astigm.</th>
<th>Reduced tear</th>
<th>Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>O2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td>O3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>O4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td>O5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>O6-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td>O15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>O16</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>NO</td>
</tr>
<tr>
<td>O17</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>O18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>NO</td>
</tr>
<tr>
<td>O19-O23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O24</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>NO</td>
</tr>
</tbody>
</table>

Binary features and class values
Text mining:
Words/terms as binary features

<table>
<thead>
<tr>
<th>Document</th>
<th>Word1</th>
<th>Word2</th>
<th>…</th>
<th>WordN</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td>d3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td>d5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d6-d13</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>d14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td>d15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d16</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>NO</td>
</tr>
<tr>
<td>d17</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>NO</td>
</tr>
<tr>
<td>d19-d23</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>d24</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>NO</td>
</tr>
</tbody>
</table>

Instances = documents
Words and terms = Binary features
Text Mining from unlabeled data

<table>
<thead>
<tr>
<th>Document</th>
<th>Word1</th>
<th>Word2</th>
<th>...</th>
<th>WordN</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td>d3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td>d5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d6-d13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>d14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td>d15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d16</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>NO</td>
</tr>
<tr>
<td>d17</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>NO</td>
</tr>
<tr>
<td>d18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>NO</td>
</tr>
<tr>
<td>d19-d23</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>d24</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>NO</td>
</tr>
</tbody>
</table>

Unlabeled data - clustering: grouping of similar instances
- association rule learning
Text mining

Step 1

BoW vector construction

1. BoW features construction
2. Table of BoW vectors construction

Step 2

Data Mining

model, patterns, clusters,
Text Mining

• Feature construction
  – StopWords elimination
  – Stemming or lemmatization
  – Term construction by frequent N-Grams construction
  – Terms obtained from thesaurus (e.g., WordNet)

• BoW vector construction

• Mining of BoW vector table
  – Feature selection, Document similarity computation
  – Text mining: Categorization, Clustering, Summarization, ...
Stemming and Lemmatization

- Different forms of the same word usually problematic for text data analysis
  - because they have different spelling and similar meaning (e.g. learns, learned, learning,…)
  - usually treated as completely unrelated words

- Stemming is a process of transforming a word into its stem
  - cutting off a suffix (eg., smejala -> smej)

- Lemmatization is a process of transforming a word into its normalized form
  - replacing the word, most often replacing a suffix (eg., smejala -> smejati)
Bag-of-Words document representation

Journal of Artificial Intelligence Research

JAIR is a refereed journal, covering all areas of Artificial Intelligence, which is distributed free of charge over the internet. Each volume of the journal is also published by Morgan Kaufman....
Word weighting

- In bag-of-words representation each word is represented as a separate variable having numeric weight.
- The most popular weighting schema is normalized word frequency TFIDF:

\[
tfidf(w) = tf \cdot \log\left(\frac{N}{df(w)}\right)
\]

- \( tf(w) \) – term frequency (number of word occurrences in a document)
- \( df(w) \) – document frequency (number of documents containing the word)
- \( N \) – number of all documents
- \( tfidf(w) \) – relative importance of the word in the document

The word is more important if it appears several times in a target document
The word is more important if it appears in less documents
Cosine similarity between document vectors

- Each document D is represented as a vector of TF-IDF weights
- Similarity between two vectors is estimated by the similarity between their vector representations (cosine of the angle between the two vectors):

\[
\text{Similarity } (D_1, D_2) = \frac{\sum_i x_{1i}x_{2i}}{\sqrt{\sum_j x_j^2} \sqrt{\sum_k x_k^2}}
\]
Advanced Topics

- Text mining: An introduction
- Document clustering and outlier detection
- Wordification approach to relational data mining
Document clustering

- Clustering is a process of finding natural groups in data in an unsupervised way (no class labels pre-assigned to documents)
- Document similarity is used
- Most popular clustering methods:
  - K-Means clustering
  - Agglomerative hierarchical clustering
  - EM (Gaussian Mixture)
  - ...
Document clustering with OntoGen
ontogen.ijs.si

Slide adapted from D. Mladenić, JSI
Using OntoGen for clustering PubMed articles on autism

Work by Petrič et al. 2009
K-Means clustering in OntoGen

OntoGen uses k-Means clustering for semi-automated topic ontology construction

• Given:
  – set of documents (eg., word-vectors with TFIDF),
  – distance measure (eg., cosine similarity)
  – K - number of groups

• For each group initialize its centroid with a random document

• While not converging
  – each document is assigned to the nearest group (represented by its centroid)
  – for each group calculate new centroid (group mass point, average document in the group)
Detecting outlier documents

- By classification noise detection on a domain pair dataset, assuming two separate document corpora A and C
Outlier detection for cross-domain knowledge discovery

2-dimensional projection of documents (about autism (red) and calcineurin (blue). Outlier documents are bolded for the user to easily spot them.

Our research has shown that most domain bridging terms appear in outlier documents. (Lavrač, Sluban, Grčar, Juršič 2010)
Using OntoGen for outlier document identification

Text corpus

Outlier Identification

Concept A'

A U C

Concept C'

Slide adapted from D. Mladenić, JSI
NoiseRank: Ensemble-based noise and outlier detection

- Misclassified document detection by an ensemble of diverse classifiers (e.g., Naive Bayes, Random Forest, SVM, ... classifiers)
- Ranking of misclassified documents by “voting” of classifiers
# NoiseRank on news articles

Articles on Kenyan elections: local vs. Western media

<table>
<thead>
<tr>
<th>Rank</th>
<th>Class</th>
<th>ID</th>
<th>Detected by:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WE</td>
<td>352</td>
<td>___<em>Bayes___RF100___RF500___SVM___SVMEasy___SatFilt</em></td>
</tr>
<tr>
<td>2</td>
<td>LO</td>
<td>25</td>
<td>___<em>Bayes___RF100___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>3</td>
<td>LO</td>
<td>101</td>
<td>___<em>Bayes___RF100___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>4</td>
<td>LO</td>
<td>173</td>
<td>___<em>Bayes___RF100___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>5</td>
<td>WE</td>
<td>348</td>
<td>___<em>Bayes___RF100___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>6</td>
<td>WE</td>
<td>326</td>
<td>___<em>Bayes___RF100___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>7</td>
<td>WE</td>
<td>357</td>
<td>___<em>Bayes___RF100___RF500___SVM___SVMEasy___SatFilt</em></td>
</tr>
<tr>
<td>8</td>
<td>WE</td>
<td>410</td>
<td>___<em>Bayes___RF100___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>9</td>
<td>LO</td>
<td>21</td>
<td>___<em>RF100___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>10</td>
<td>LO</td>
<td>4</td>
<td>___<em>Bayes___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>11</td>
<td>LO</td>
<td>68</td>
<td>___<em>RF100___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>12</td>
<td>LO</td>
<td>162</td>
<td>___<em>Bayes___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>13</td>
<td>WE</td>
<td>358</td>
<td>_<em><strong>Bayes___RF100___RF500___SVM</strong></em></td>
</tr>
<tr>
<td>14</td>
<td>WE</td>
<td>464</td>
<td>___<em>RF100___RF500___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>15</td>
<td>LO</td>
<td>153</td>
<td>___<em>Bayes___SVM___SVMEasy</em></td>
</tr>
<tr>
<td>16</td>
<td>LO</td>
<td>201</td>
<td>___<em>RF100___RF500___SatFilt</em></td>
</tr>
<tr>
<td>17</td>
<td>WE</td>
<td>238</td>
<td>___<em>RF100___RF500___SVM</em></td>
</tr>
<tr>
<td>18</td>
<td>WE</td>
<td>364</td>
<td>___<em>Bayes___RF500___SVM</em></td>
</tr>
<tr>
<td>19</td>
<td>WE</td>
<td>370</td>
<td>___<em>Bayes___RF100___SVM</em></td>
</tr>
<tr>
<td>20</td>
<td>WE</td>
<td>379</td>
<td>___<em>RF100___RF500___SVMEasy</em></td>
</tr>
</tbody>
</table>
NoiseRank on news articles

- **Article 352: Out of topic**
The article was later indeed removed from the corpus used for further linguistic analysis, since it is not about Kenya(ns) or the socio-political climate but about British tourists or expatriates’ misfortune.

- **Article 173: Guest journalist**
Wrongly classified because it could be regarded as a “Western article” among the local Kenyan press. The author does not have the cultural sensitivity or does not follow the editorial guidelines requiring to be careful when mentioning words like tribe in negative contexts. One could even say that he has a kind of “Western” writing style.
Advanced Topics

• Text mining: An introduction
• Document clustering and outlier
Wordification approach to relational data mining
Motivation

• Develop a RDM technique inspired by text mining
• Using a large number of simple, easy to understand features (words)
• Improved scalability, handling large datasets
• Used as a preprocessing step to propositional learners
Wordification Methodology

• Transform a relational database to a document corpus
  • For each individual (row) in the main table, concatenate words generated for the main table with words generated for the other tables, linked through external keys
Wordification Methodology

- One individual of the main data table in the relational database ~ one text document
- Features (attribute values) ~ the words of this document
- Individual words (called word-items or witems) are constructed as combinations of:
  \[ [\text{table name}]_[\text{attribute name}]_[\text{value}] \]
- n-grams are constructed to model feature dependencies:
  \[ [\text{witem}_1]_[\text{witem}_2]_..._[\text{witem}_n] \]
Wordification Methodology

• Transform a relational database to a document corpus
• Construct BoW vectors with TF-IDF weights on words
  (optional: Perform feature selection)
• Apply text mining or propositional learning on BoW table
### Wordification

<table>
<thead>
<tr>
<th>TRAIN</th>
<th>CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>trainID</td>
<td>eastbound</td>
</tr>
<tr>
<td>t1</td>
<td>east</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>t5</td>
<td>west</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CAR</th>
<th>[carID, shape, roof, wheels, train]</th>
</tr>
</thead>
<tbody>
<tr>
<td>c11</td>
<td>rectangle, none, 2, t1</td>
</tr>
<tr>
<td>c12</td>
<td>rectangle, peaked, 3, t1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>c51</td>
<td>rectangle, none, 2, t5</td>
</tr>
<tr>
<td>c52</td>
<td>hexagon, flat, 2, t5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**t1:** [car_roof_none, car_shape_rectangle, car_wheels_2, car_roof_none__car_shape_rectangle, car_roof_none__car_wheels_2, car_shape_rectangle__car_wheels_2, car_roof_peaked, car_shape_rectangle, car_wheels_3, car_roof_peaked__car_shape_rectangle, car_roof_peaked__car_wheels_3, car_shape_rectangle__car_wheels_3], east
Wordification

t1: [car_roof_none, car_shape_rectangle, car_wheels_2, car_roof_none__car_shape_rectangle, car_roof_none__car_wheels_2, car_shape_rectangle__car_wheels_2, car_roof_peaked, car_shape_rectangle, car_wheels_3, car_roof_peaked__car_shape_rectangle, car_roof_peaked__car_wheels_3, car_shape_rectangle__car_wheels_3], east

t5: [car_roof_none, car_shape_rectangle, car_wheels_2, car_roof_none__car_shape_rectangle, car_roof_none__car_wheels_2, car_shape_rectangle__car_wheels_2, car_roof_flat, car_shape_hexagon, car_wheels_2, car_roof_flat__car_shape_hexagon, car_roof_flat__car_wheels_2, car_shape_hexagon__car_wheels_2], west

TF-IDF calculation for BoW vector construction:

<table>
<thead>
<tr>
<th>car_shape_rectangle</th>
<th>car_roof_peaked</th>
<th>car_wheels_3</th>
<th>car_roof_peaked__car_shape_rectangle</th>
<th>car_shape_rectangle__car_wheels_3</th>
<th>...</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>0.000</td>
<td>0.693</td>
<td>0.693</td>
<td>0.693</td>
<td></td>
<td>east</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>t5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td>west</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
TF-IDF weights

• No explicit use of existential variables in features, TF-IDF instead

• The weight of a word indicates how relevant is the feature for the given individual

• The TF-IDF weights can then be used either for filtering words with low importance or for using them directly by a propositional learner (e.g. J48)
Experiments

- Cross-validation experiments on 8 relational datasets: Trains (in two variants), Carcinogenesis, Mutagenenesis with 42 and 188 examples, IMDB, and Financial.
- Results (using J48 for propositional learning)
Experiments

• Cross-validation experiments on 8 relational datasets: Trains (in two variants), Carcinogenesis, Mutagenensis with 42 and 188 examples, IMDB, and Financial.

• Results (using J48 for propositional learning)
  – first applying Friedman test to rank the algorithms,
  – then post-hoc test Nemenyi test to compare multiple algorithms to each other
Experiments

• Cross-validation experiments on 8 relational datasets: Trains (in two variants), Carcinogenesis, Mutagenensis with 42 and 188 examples, IMDB, and Financial.

Results (using J48 for propositional learning):

- Measure = CA
  - CD = 1.77
  - Wordification (1.9)
  - AlephFeaturize (2.5)
  - RSD (2.7)
  - ReIF (2.9)

- Measure = run-time
  - CD = 1.77
  - Wordification (1.0)
  - AlephFeaturize (2.9)
  - RSD (3.0)
  - ReIF (3.1)
<table>
<thead>
<tr>
<th>Domain</th>
<th>Algorithm</th>
<th>J48-Accuracy [%]</th>
<th>J48-AUC</th>
<th>Run-time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trains</td>
<td>Wordification</td>
<td>55.00</td>
<td>0.51</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>ReIF</td>
<td>65.00</td>
<td>0.65</td>
<td>1.04</td>
</tr>
<tr>
<td></td>
<td>RSD</td>
<td>65.00</td>
<td>0.68</td>
<td>0.53</td>
</tr>
<tr>
<td></td>
<td>AlephFeaturize</td>
<td>75.00</td>
<td>0.82</td>
<td>0.40</td>
</tr>
<tr>
<td>Trains</td>
<td>Wordification</td>
<td>95.00</td>
<td>0.91</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>ReIF</td>
<td>65.00</td>
<td>0.62</td>
<td>1.06</td>
</tr>
<tr>
<td></td>
<td>RSD</td>
<td>50.00</td>
<td>0.53</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>AlephFeaturize</td>
<td>85.00</td>
<td>0.74</td>
<td>0.38</td>
</tr>
<tr>
<td>Mutagenesis42</td>
<td>Wordification</td>
<td>97.62</td>
<td>0.93</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>ReIF</td>
<td>80.95</td>
<td>0.59</td>
<td>2.11</td>
</tr>
<tr>
<td></td>
<td>RSD</td>
<td>97.62</td>
<td>0.93</td>
<td>2.63</td>
</tr>
<tr>
<td></td>
<td>AlephFeaturize</td>
<td>97.62</td>
<td>0.93</td>
<td>2.07</td>
</tr>
<tr>
<td>Mutagenesis188</td>
<td>Wordification</td>
<td>95.74</td>
<td>0.90</td>
<td>1.65</td>
</tr>
<tr>
<td></td>
<td>ReIF</td>
<td>75.53</td>
<td>0.79</td>
<td>7.76</td>
</tr>
<tr>
<td></td>
<td>RSD</td>
<td>94.15</td>
<td>0.91</td>
<td>10.10</td>
</tr>
<tr>
<td></td>
<td>AlephFeaturize</td>
<td>87.23</td>
<td>0.88</td>
<td>19.27</td>
</tr>
<tr>
<td>IMDB</td>
<td>Wordification</td>
<td>84.34</td>
<td>0.79</td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td>ReIF</td>
<td>79.52</td>
<td>0.73</td>
<td>32.49</td>
</tr>
<tr>
<td></td>
<td>RSD</td>
<td>73.49</td>
<td>0.47</td>
<td>4.33</td>
</tr>
<tr>
<td></td>
<td>AlephFeaturize</td>
<td>73.49</td>
<td>0.47</td>
<td>4.96</td>
</tr>
<tr>
<td>Carcinogenesis</td>
<td>Wordification</td>
<td>61.09</td>
<td>0.62</td>
<td>1.79</td>
</tr>
<tr>
<td></td>
<td>ReIF</td>
<td>54.71</td>
<td>0.53</td>
<td>16.44</td>
</tr>
<tr>
<td></td>
<td>RSD</td>
<td>58.05</td>
<td>0.56</td>
<td>9.29</td>
</tr>
<tr>
<td></td>
<td>AlephFeaturize</td>
<td>55.32</td>
<td>0.49</td>
<td>104.70</td>
</tr>
<tr>
<td>Financial</td>
<td>Wordification</td>
<td>86.75</td>
<td>0.48</td>
<td>4.65</td>
</tr>
<tr>
<td></td>
<td>ReIF</td>
<td>97.00</td>
<td>0.91</td>
<td>260.93</td>
</tr>
<tr>
<td></td>
<td>RSD</td>
<td>86.75</td>
<td>0.48</td>
<td>533.68</td>
</tr>
<tr>
<td></td>
<td>AlephFeaturize</td>
<td>86.75</td>
<td>0.48</td>
<td>525.86</td>
</tr>
</tbody>
</table>
Use Case: IMDB

- **IMDB subset**: Top 250 and bottom 100 movies
- Movies, actors, movie genres, directors, director genres
- Wordification methodology applied
- Association rules learned on BoW vector table
Use Case: IMDB

goodMovie ← director_genre_drama, movie_genre_thriller, director_name_AlfredHitchcock. (Support: 5.38% Confidence: 100.00%)

movie_genre_drama ← goodMovie, actor_name_RobertDeNiro. (Support: 3.59% Confidence: 100.00%)

director_name_AlfredHitchcock ← actor_name_AlfredHitchcock. (Support: 4.79% Confidence: 100.00%)

director_name_StevenSpielberg ← goodMovie, movie_genre_adventure, actor_name_TedGrossman. (Support: 1.79% Confidence: 100.00%)
Wordification implemented in ClowdFlows

- Propositionalization through wordification, available at http://clowdflows.org/workflow/1455/

June 28, 2013
DAISY, Konstanz
Evaluation implemented in ClowdFlows

- Wordification and propositionalization algorithms comparison, available at http://clowdflows.org/workflow/1456/
Summary

– Wordification methodology
– Implemented in ClowdFlows
– Allows for solving non-standard RDM tasks, including RDM clustering, word cloud visualization, association rule learning, topic ontology construction, outlier detection, …