Data Mining in ClowdFlows and TextFlows

Nada Lavrač, Anže Vavpetič, Matej Martinc

Jožef Stefan Institute
Ljubljana, Slovenia

Comtrade Seminar, 31.5.2017
Jožef Stefan Institute (JSI)
Ljubljana, Slovenia

• Jožef Stefan Institute (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)

• Jožef Stefan International Postgraduate School
  (founded in 2004)
  – Offers four MSc and PhD programs (in English):
    - information and communication technologies
    - nanotechnologies, ecotechnologies, sensor technologies

• Department of Knowledge Technologies
  Head: Nada Lavrač, Staff: about 30 researchers, 10 students
Department of Knowledge Technologies
Course Outline

9:00-10:30 **Nada Lavrač: Data mining essentials**
   decision trees, rules, regression, data mining platforms
10:30-11:00 coffee break
11:00-12:30 **Anže Vavpetič: ClowdFlows essentials**
   creating a package, a widget, widget types, basic DM workflows
12:30-13:00 Questions and discussion
13:00-14:00 Lunch break

14:00-14:45 **Nada Lavrač: Pattern Mining**
   Subgroup discovery, relational DM, association rules
14:45-15:30 **Anže Vavpetič: Relational DM in ClowdFlows**
15:30-16:00 coffee break
16:00-16:45 **Nada Lavrač: Text Mining**
   From data mining to text mining
16:45-17:30 **Matej Martinc: Text Mining in TextFlows**
17:30-18:00 Questions, discussion and conclusions
Part I. Data Mining Essentials

Data Mining and the KDD process

- Introduction to Data Mining
- Data Mining platforms
- Predictive Data Mining techniques (classification, regression) and classifier evaluation
What is Data Mining

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

• **Data Mining (DM)** – techniques and applications of **Machine Learning**, aimed at solving real-life data analysis problems

• **Knowledge discovery in databases (KDD)** – the process of knowledge discovery
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
What is Data Mining

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

• Data Mining (DM) – techniques and applications of Machine Learning, aimed at solving real-life data analysis problems

• Knowledge discovery in databases (KDD) – the process of knowledge discovery

• Big data – data and techniques for dealing with very large volumes of data, possibly dynamic data streams
The 4 Vs of Big Data

**Volume**
- Scale of Data
- World population: 7 billion people
- 6 billion people have cell phones
- The New York Stock Exchange captures 1 TB of trade information during each trading session
- 40 zettabytes (43 trillion gigabytes) of data will be created by 2020, an increase of 300 times from 2005
- Most companies in the U.S. have at least 100 terabytes (100,000 gigabytes) of data stored
- It’s estimated that 2.5 quintillion bytes (2.3 trillion gigabytes) of data are created each day

**Variety**
- Different forms of data
- 4 billion plus hours of video are watched on YouTube each month
- 400 million tweets are sent per day by about 200 million monthly active users
- 30 billion pieces of content are shared on Facebook every month
- 420 million wearable, wireless health monitors

**Velocity**
- Analysis of streaming data
- Modern cars have close to 100 sensors that monitor items such as fuel level and tire pressure
- By 2016, it is projected there will be 18.9 billion network connections — almost 2.5 connections per person on earth
- By 2015, 4.4 million IT jobs will be created globally to support big data, with 1.9 million in the United States

**Veracity**
- Uncertainty of data
- 27% of respondents in one survey were unsure of how much of their data was inaccurate
- Poor data quality costs the US economy around $3.1 trillion a year

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can those massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety, and Veracity**

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adopt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, GES
Given: transaction data table, relational database, text documents, Web pages
Find: a classification model, a set of interesting patterns
Data Mining

Given: transaction data table, relational database, text documents, Web pages
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>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>
## 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>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>
# Pattern discovery in 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>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>

**Rule:**

IF  
Tear prod. = reduced  
THEN  
Lenses = NONE
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-O13</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>O14</td>
<td>pre-presbyc</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>pre-presbyc</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>pre-presbyc</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>

Diagram:

- **tear prod.**
  - reduced:
    - NONE
  - normal:
    - astigmatism
      - no:
        - SOFT
      - yes:
        - spect. pre.
          - myope:
            - HARD
          - hypermetrope:
            - NONE
Decision tree classification model learned from contact lens data

Nodes: attributes
Arcs: values of attributes
Leaves: classes

- **tear prod.**
  - reduced
    - **NONE**
  - normal
    - no
      - **SOFT**
    - yes
      - **spect. pre.**
        - myope
          - **HARD**
        - hypermetrope
          - **NONE**
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>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>

lenses=NONE ← tear production=red
lenses=NONE ← tear production=normal AND astigmatism=yes AND spect. pre.=hypermetrope
lenses=SOFT ← tear production=normal AND astigmatism=no
lenses=HARD ← tear production=normal AND astigmatism=yes AND spect. pre.=myope
lenses=NONE ←
Classification rules model learned from contact lens data

lenses=NONE  ← tear production=reduced
lenses=NONE  ← tear production=normal AND astigmatism=yes AND spect. pre.=hypermetrope
lenses=SOFT  ← tear production=normal AND astigmatism=no
lenses=HARD  ← tear production=normal AND astigmatism=yes AND spect. pre.=myope
lenses=NONE  ←
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 tasks
  - classification and class description
  - “one vs. all” multi-class learning
- for Subgroup discovery tasks
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
Why learn and use symbolic models

**Given:** the learned classification model
(a decision tree or a set of rules)

**Find:**
- the class label for a new unlabeled instance
- use the model for the explanation of classifications of new data instances
- use the discovered patterns for data exploration
First Generation Data Mining

- **First machine learning algorithms for**
  - Decision tree and rule learning in 1970s and early 1980s by Quinlan, Michalski et al., Breiman et al., …

- **Characterized by**
  - Learning from data stored in a single data table
  - Relatively small set of instances and attributes

- **Lots of ML research followed in 1980s**
  - Numerous conferences ICML, ECML, … and ML sessions at AI conferences IJCAI, ECAI, AAAI, …
  - Extended set of learning tasks and algorithms addressed
Second Generation Data Mining

- **Developed since 1990s:**
  - Focused on data mining tasks characterized by large datasets described by large numbers of attributes
Second Generation Data Mining

• **Developed since 1990s:**
  – Focused on data mining tasks characterized by large datasets described by large numbers of attributes

  ![Data Mining Process Diagram](image)

  – New conferences on practical aspects of data mining and knowledge discovery: KDD, PKDD, ...
  – New learning tasks and efficient learning algorithms:
    • Learning predictive models: Bayesian network learning, relational data mining, statistical relational learning, SVMs, ...
    • Learning descriptive patterns: association rule learning, subgroup discovery, ...
Second Generation Data Mining Platforms

Orange, WEKA, KNIME, RapidMiner, …
Second Generation Data Mining Platforms

Orange, WEKA, KNIME, RapidMiner, …

– include numerous data mining algorithms
– enable data and model visualization
– like Orange, Taverna, WEKA, KNIME, RapidMiner, also enable complex workflow construction
Towards Third Generation Data Mining

- **Orange4WS** (Podpečan et al. 2009), **ClowdFlows** (Kranjc et al. 2012) and **TextFlows** (Perovšek et al. 2016)
  - are service oriented (DM algorithms as web services)
  - user-friendly HCI: canvas for workflow construction
  - include functionality of standard data mining platforms
    - WEKA algorithms, implemented as Web services
  - Include new functionality
    - relational data mining
    - semantic data mining
    - NLP processing and text mining
  - enable simplified construction of Web services from available algorithms
  - ClowdFlows and TextFlows run in a browser – enables data mining, workflow construction and sharing on the web
ClowdFlows platform

- Large algorithm repository
  - Relational data mining
  - All Orange algorithms
  - WEKA algorithms as web services
  - Data and results visualization
  - Text analysis
  - Social network analysis
  - Analysis of big data streams

- Large workflow repository
  - Enables access to our technology heritage
ClowdFlows platform

- Large repository of algorithms
- Large repository of workflows

Example workflow: Propositionalization with RSD available in ClowdFlows at http://clowdflows.org/workflow/611/
TextFlows

• Motivation:
  – Develop an online text mining platform for composition, execution and sharing of text mining workflows

• TextFlows platform – fork of ClowdFlows.org:
  – Web-based user interface
  – Visual programming
  – Big roster of existing workflow (mostly data mining) components
  – Cloud-based service-oriented architecture
“Big Data” Use Case

• Real-time analysis of big data streams

• Example: news monitoring by graph visualization (graph of CNN RSS feeds) http://clowdflows.org/streams/data/31/1
Part I. Data Mining Essentials

- Data Mining and the KDD process
- Introduction to Data Mining
- Data Mining platforms
- Predictive Data Mining techniques (classification, regression) and classifier evaluation
Decision tree classifier

- **tear prod.**
  - reduced: NONE
  - normal:
    - **astigmatism**
      - no: SOFT
      - yes: **spect. pre.**
        - myope: HARD
        - hypermetrope: NONE

Decision tree learning algorithm

- ID3 (Quinlan 1979), CART (Breiman et al. 1984), C4.5, J48 in 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 v1, v2, ... vn
    - divide training set S into S1,...,Sn according to values v1,...,vn
    - recursively build sub-trees T1,...,Tn for S1,...,Sn
Decision tree search heuristics

• Central choice in decision tree algorithms: 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, **C₁,...,Cₙ** - 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 **Cₖ**

(relative frequency of **Cₖ** 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** $E(S) =$ expected amount of information (in bits) needed to assign a class to a randomly drawn object in $S$ (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:

$$p_+ (- \log_2 p_+) + p_- (- \log_2 p_-) = - p_+ \log_2 p_+ - p_- \log_2 p_-$$
Entropy – example calculation

• 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
Information gain
search heuristic

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

• \( \text{Gain}(S, A) \) – expected reduction in entropy of \( S \) due to sorting on \( A \)

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

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

• Which attribute is more informative, A1 or A2?

- Gain(S,A1) = 0.94 – (8/14 × 0.811 + 6/14 × 1.00) = 0.048
- Gain(S,A2) = 0.94 – 0 = 0.94
  A2 has max Gain
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
Handling noise – Tree pruning

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_malig

< 3

≥ 3

Tumor_size

< 15

≥ 15

Age

no_recur 125
recurrence 39

Involved_nodes

< 3

≥ 3

no_recur 30
recurrence 18

no_recur 27
recurrence 10

< 40

≥ 40

no_recur 4
recurrence 1

no_recur 4

no_recur 4

no_rec 4
rec1
Pruned decision tree for contact lenses recommendation

- **Tear Production**
  - Reduced: NONE
  - Normal:
    - No: SOFT
      - [S=5, H+N=1]
    - Yes:
      - Spectacle Prescription
        - Myope: HARD
          - [H=3, S+N=2]
        - Hypermetrope: NONE
          - [N=2, S+H=1]
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
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 and J48

- 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
Classifier evaluation

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

• **Evaluating the quality of induced models**
  – 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
n-fold cross validation

- 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** $i = 1, ..., n$ **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

• 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
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
<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>:</td>
<td><strong>Evaluation</strong>: cross validation, separate test set, …</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 regression problem

- data about 80 people: Age and Height

<table>
<thead>
<tr>
<th>Age</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1.03</td>
</tr>
<tr>
<td>5</td>
<td>1.19</td>
</tr>
<tr>
<td>6</td>
<td>1.26</td>
</tr>
<tr>
<td>9</td>
<td>1.39</td>
</tr>
<tr>
<td>15</td>
<td>1.69</td>
</tr>
<tr>
<td>19</td>
<td>1.67</td>
</tr>
<tr>
<td>22</td>
<td>1.86</td>
</tr>
<tr>
<td>25</td>
<td>1.85</td>
</tr>
<tr>
<td>41</td>
<td>1.59</td>
</tr>
<tr>
<td>48</td>
<td>1.60</td>
</tr>
<tr>
<td>54</td>
<td>1.90</td>
</tr>
<tr>
<td>71</td>
<td>1.82</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
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
Baseline numeric predictor

- Average of the target variable is 1.63
Linear Regression Model

Height = 0.0056 \times \text{Age} + 1.4181
Regression tree

Height = 1.3932
LM 1 (5/23.737%)
Age ≤ 4
LM 2 (4/13.419%
Age > 4

Height = 1.4025

Height = 1.7096
LM 3 (8/45.928%)
Age ≤ 6.5
LM 4 (63/44.833%)
Age > 6.5

Age

Height

Height

Prediction

Age

0
0.5
1
1.5
2
0 50 100

Height

0 0.5 1 1.5 2

0 50 100

Height

Height

Prediction

Age

0 50 100

Height

0 0.5 1 1.5 2

0 50 100

Height
Model tree

- Age
  - $\leq 12.5$
    - LM 1 (17/15.516%)
      - Height = $0.0333 \times \text{Age} + 1.1366$
  - $> 12.5$
    - LM 2 (63/44.833%)
      - Height = $0.0011 \times \text{Age} + 1.6692$

Age vs. Height:
- Kohonen self-organizing map (SOM) with prediction overlay.
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               | \[
\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(a_1 - \bar{a})^2 + \ldots + (a_n - \bar{a})^2}, \text{ where } \bar{a} = \frac{1}{n} \sum_i a_i
\]
| root relative squared error          | \[
\sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(a_1 - \bar{a})^2 + \ldots + (a_n - \bar{a})^2}}
\]
| relative absolute error              | \[
\frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{|a_1 - \bar{a}| + \ldots + |a_n - \bar{a}|}
\]
| correlation coefficient              | \[
\frac{S_{PA}}{\sqrt{S_p S_A}}, \text{ where } S_{PA} = \frac{\sum_i (p_i - \bar{p})(a_i - \bar{a})}{n-1},
\]
|                                      | \[
S_p = \frac{\sum_i (p_i - \bar{p})^2}{n-1}, \text{ and } S_A = \frac{\sum_i (a_i - \bar{a})^2}{n-1}
\]
Rule Learning

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</td>
<td>hypermetrope</td>
<td>no</td>
<td>normal</td>
<td>SOFT</td>
</tr>
<tr>
<td>O15</td>
<td>pre-presbyc</td>
<td>hypermetrope</td>
<td>yes</td>
<td>reduced</td>
<td>NONE</td>
</tr>
<tr>
<td>O16</td>
<td>pre-presbyc</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>

**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

- Form of CN2 rules:
  IF Conditions THEN MajClass [ClassDistr]
- Rule base: \{R1, R2, R3, \ldots, DefaultRule\}
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>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>
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.)

– RuleBase($C_i$) := empty

– repeat \{learn-set-of-rules\}
  
  • **learn-one-rule** $R$ covering some positive examples and no negatives
  
  • add $R$ to 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

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

Positive examples

Negative examples
Covering algorithm

Positive examples

Negative examples

Rule 1: Cl=+ $\iff$ Cond2 AND Cond3
Covering algorithm

Positive examples

Rule 1: \( Cl = + \) \( \iff \) Cond2 AND Cond3

Negative examples

Rule 2: \( Cl = + \) \( \iff \) Cond8 AND Cond6
Probability estimates

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

- **Laplace estimate**: assumes uniform prior distribution of \( k \) classes
  \[ p\left(\text{Class} \mid \text{Cond}\right) = \frac{n(\text{Class} \cdot \text{Cond}) + 1}{n(\text{Cond}) + k} \]

- Examples:
  - \([6+,1-] \) (7) = \( \frac{6}{7} \)
  - \([2+,0-] \) (2) = \( \frac{2}{2} = 1 \)
  - \([6+,1-] \) (7) = \( \frac{6+1}{7+2} = \frac{7}{9} \)
  - \([2+,0-] \) (2) = \( \frac{2+1}{2+2} = \frac{3}{4} \)
Learn-one-rule: search heuristics

- Assume a two-class problem
- Two classes (+,-), learn rules for + class (Cl).
- Search for specializations $R'$ of a rule $R = Cl \leftarrow Cond$ 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: 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
    \[ \text{lenses}=\text{NONE} \leftarrow \]
  – is rule \[ \text{lenses}=\text{NONE} \leftarrow \text{tear production}=\text{reduced} \]

• 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
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 \( Cl \leftarrow \text{true} \)
  – 68% accuracy is OK if there are 20% examples of that class in the training set, but bad if there are 80%

• Relative accuracy \((\text{relative precision})\)
  – \( \text{RAcc}(Cl \leftarrow \text{Cond}) = p(\text{Cl} | \text{Cond}) - p(\text{Cl}) \)
Learn-one-rule: search heuristics

- Assume two classes (+,-), learn rules for + class (Cl). Search for specializations of one rule R = Cl ← Cond from RuleBase.
- Expected **classification accuracy**: \( A(R) = p(Cl|Cond) \)
- **Informativity** (info needed to specify that example covered by Cond belongs to Cl): \( I(R) = - \log_2 p(Cl|Cond) \)
- **Accuracy gain** (increase in expected accuracy): \( AG(R',R) = p(Cl|Cond') - p(Cl|Cond) \)
- **Information gain** (decrease in the information needed): \( IG(R',R) = \log_2 p(Cl|Cond') - \log_2 p(Cl|Cond) \)
- **Weighted** measures favoring more general rules: WAG, WIG
  \( WAG(R',R) = p(Cond')/p(Cond) \cdot (p(Cl|Cond') - p(Cl|Cond)) \)
- **Weighted relative accuracy** trades off coverage and relative accuracy \( WRAcc(R) = p(Cond) \cdot (p(Cl|Cond) - p(Cl)) \)
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

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

- RuleBase := empty
- E_{cur} := E
- **repeat**
  - learn-one-rule R
  - RuleBase := RuleBase U R
  - E_{cur} := E_{cur} - \{all examples covered by R\}
    (NOT ONLY POS. EX.!!)
- **until** performance(R, E_{cur}) < ThresholdR
- RuleBase := sort RuleBase by performance(R,E)
- RuleBase := RuleBase U DefaultRule(E_{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 $(R, E_{\text{cur}}) : - \text{Entropy}(E_{\text{cur}})$
  - performance$(R, E_{\text{cur}}) < \text{Threshold}_R$ (neg. num.)
  - Why? Ent. $> t$ is bad, Perf. $= -\text{Ent} < -t$ 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

- Best performing rule learning algorithm: Ripper
- JRip implementation of Ripper in WEKA, available in ClowdFlows
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.
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
  \[ p(c_j \mid v_1\ldots v_n) = p(c_j) \frac{p(v_1\ldots v_n \mid c_j)}{p(v_1\ldots v_n)} \]

- 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 \( p(C_j \mid v_i) \))
  \[ p(c_j \mid v_1\ldots v_n) \approx p(c_j) \prod_{i} \frac{p(c_j \mid v_i)}{p(c_j)} \]

- Output \( C_{\text{MAX}} \) with maximal posterior probability of class:
  \[ C_{\text{MAX}} = \arg\max_{C_j} p(c_j \mid v_1\ldots v_n) \]
Semi-naïve Bayesian classifier

• Naive Bayesian estimation of probabilities (reliable)

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

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

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

• Relative frequency:

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

• Prior probability: Laplace law

\[ p(c_j) = \frac{n(c_j) + 1}{N + k} \]

• m-estimate *

* Out of scope of this course
Probability estimates

• Relative frequency :
  – problems with small samples

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

\[
\begin{align*}
[6+,1-] (7) &= \frac{6}{7} \\
[2+,0-] (2) &= \frac{2}{2} = 1
\end{align*}
\]

• Laplace estimate :
  – assumes uniform prior distribution of k classes

\[
= \frac{n(Class, Cond) + 1}{n(Cond) + k} \quad k = 2
\]

\[
\begin{align*}
[6+,1-] (7) &= \frac{6+1}{7+2} = \frac{7}{9} \\
[2+,0-] (2) &= \frac{2+1}{2+2} = \frac{3}{4}
\end{align*}
\]
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
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, …