Data Mining
as Part of Knowledge Discovery in Databases (KDD)

Presented by Naci Akkøk
as part of INF4180/3180, Advanced Database Systems, fall 2003
(based on slightly modified foils of Dr. Denise Ecklund from 6 November 2002)

Lecture Contents

- KDD: Knowledge Discovery in Databases
  - Definition and Applications
- OLAP: On-Line Analytical Processing
- Architectures for OLAP and KDD
- KDD Life Cycle
- Data Mining
  - Mechanisms
  - Implications for the Database System

Literature:
Elmasri & Navathe - Section 26.2
Garcia-Molina, Ullman, & Widom - Section 20.6
Definition - KDD

- Knowledge Discovery in Databases (KDD)
  - "the non-trivial extraction of implicit, previously unknown and potentially useful knowledge from data"

- Why?
  - To find trends and correlations in existing databases, that help you change structures and processes in human organizations
    - to be more effective
    - to save time
    - to make more money
    - to improve product quality
    - to etc.

Applications - KDD

- Marketing
  - Find the most important segmentation for classifying my customers
  - Predict future sales for a specific product

- Product Maintenance
  - Define a service maintenance contract of interest to a majority of my customers

- Human Resource Management
  - Define an employee compensation package to increase employee retention to at least 5 years of service
  - For each university department, predict the number of new students that will major in that subject area

- Finance
  - Detect fraudulent use of credit cards
Where SQL Stops Short

- Successful KDD uses the entire tool hierarchy
  - Creates new knowledge in the form of predictions, classifications, associations, time-based patterns
  - Extends the DB schema to a limited, multi-dimensional view for Decision Support Systems (DSS)
  - DB schema describes the structure you already know

OnLine Analytical Processing (OLAP)

- OLAP
  - "the dynamic synthesis, analysis, and consolidation of large volumes of multi-dimensional data"
  - Focuses on multi-dimensional relationships among existing data records
- Differs from extended SQL for data warehouses
  - DW operations: roll-up, drill-down, slice, dice, pivot
  - OLAP packages add application-specific analysis
    - Finance – depreciation, currency conversion, ...
    - Building regulations – usable space, electrical loading, ...
    - Computer Capacity Planning – disk storage requirements, failure rate estimation...
OnLine Analytic Processing (OLAP)

- OLAP differs from data mining
  - OLAP tools provide quantitative analysis of multi-dimensional data relationships
  - Data mining tools create and evaluate a set of possible problem solutions (and rank them)
    - Ex: Propose 3 marketing strategies and order them based on marketing cost and likely sales income

- Three system architectures are used for OLAP
  - Relational OLAP (ROLAP)
  - Multi-dimensional OLAP (MOLAP)
  - Managed Query Environment (MQE)

Relational OLAP

- Runtime Steps:
  - Translate client request into 1 or more SQL queries
  - Present SQL queries to a back-end RDBMS
  - Convert result relations into multi-dimensional datacubes
  - Execute the analysis application and return the results

- Advantages:
  - No special data storage structures, use standard relational storage structures
  - Uses most current data from the OLTP server

- Disadvantages:
  - Typically accesses only one RDBMS => no data integration over multiple DBSs
  - Conversion from flat relations to memory-based datacubes is slow and complex
  - Poor performance if large amounts of data are retrieved from the RDBMS
Multi-dimensional OLAP

- Preprocessing Steps:
  - Extract data from multiple data sources
  - Store as a data warehouse, using custom storage structures

- Runtime Steps:
  - Access datacubes through special index structures
  - Execute the analysis application and returns the results

- Advantages:
  - Special, multi-dimensional storage structures give good retrieval performance
  - Warehouse integrates "clean" data from multiple data sources

- Disadvantages:
  - Inflexible, multi-dimensional storage structures support only one application well
  - Requires people and software to maintain the data warehouse

Managed Query Environment (MQE)

- Client Runtime Steps:
  - Fetch data from MOLAP Server, or RDBMS directly
  - Build memory-based data structures, as required
  - Execute the analysis application

- Advantages:
  - Distributes workload to the clients, offloading the servers
  - Simple to install, and maintain => reduced cost

- Disadvantages:
  - Provides limited analysis capability (i.e., client is less powerful than a server)
  - Lots of redundant data stored on the client systems
  - Client-defined and cached datacubes can cause inconsistent data
  - Uses lots of network bandwidth
KDD System Architecture

- Need to mine all types of data sources
- DM tools require multi-dimensional input data
- General KDD systems also hosts OLAP tools, SQL interface, and SQL for DWs (e.g., nD SQL)

Typical KDD Deployment Architecture

- Runtime Steps:
  - Submit knowledge request
  - Fetch datacubes from the warehouse
  - Execute knowledge acquisition tools
  - Return findings to the client for “display”

- Advantages:
  - Data warehouse provides "clean", maintained, multi-dimensional data
  - Data retrieval is typically efficient
  - Data warehouse can be used by other applications
  - Easy to add new KDD tools

- Disadvantages:
  - KDD is limited to data selected for inclusion in the warehouse
  - If DW is not available, use MOLAP server or provide warehouse on KDD server
KDD Life Cycle

Data Preparation Phases

- Commonly part of data warehouse preparation
- Additional preparation for KDD

Data Selection and Extraction
Data Cleaning
Data Enrichment
Data Coding
Data Mining
Result Reporting

Data preparation and result reporting are 80% of the work!
Based on results you may decide to:
- Get more data from internal data sources
- Get additional data from external sources
- Recode the data

Data Enrichment

- Integrating additional data from external sources
- Sources are public and private agencies
  - Government, Credit bureau, Research lab, …
- Typical data examples:
  - Average income by city, occupation, or age group
  - Percentage of homeowners and car owners by …
  - A person’s credit rating, debt level, loan history, …
  - Geographical density maps (for population, occupations, …)
- New data extends each record from internal sources
- Database issues:
  - More heterogenous data formats to be integrated
  - Understand the semantics of the external source data
Data Coding

- Goal: to streamline the data for effective and efficient processing by the target KDD application
- Steps:
  1) Delete records with many missing values
     - But …in fraud detection, missing values are indicators of the behavior you want to discover!
  2) Delete extra attributes
     - Ex: delete customer names if looking at customer classes
  3) Code detailed data values into categories or ranges based on the types of knowledge you want to discover
     - Ex: divide specific ages into age ranges, 0-10, 11-20, …
       map home addresses to regional codes
       convert homeownership to "yes" or "no"
       convert purchase date to a month number starting from Jan. 1990

The Data Mining Process

- Based on the questions being asked and the required "form" of the output
  1) Select the data mining mechanisms you will use
  2) Make sure the data is properly coded for the selected mechanisms
     - Ex. A tool may accept numeric input only
  3) Perform rough analysis using traditional tools
     - Create a naive prediction using statistics, e.g., averages
     - The data mining tools must do better than the naive prediction or you are not learning more than the obvious!
  4) Run the tool and examine the results
**Data Mining - Mechanisms**

<table>
<thead>
<tr>
<th>Purpose/Use</th>
<th>Knowledge Type</th>
<th>Mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>To define classes and predict future behavior of existing instances.</td>
<td>Predictive Modeling</td>
<td>Decision Trees, Neural Networks, Regression Analysis, Genetic Algorithms</td>
</tr>
<tr>
<td>To define classes and categorize new instances based on the classes.</td>
<td>Database Segmentation</td>
<td>K-nearest Neighbors, Neural Networks, Kohonen Maps</td>
</tr>
<tr>
<td>To identify a cause and predict the effect.</td>
<td>Link Analysis</td>
<td>Negative Association, Association Discovery, Sequential Pattern Discovery, Matching Time Sequence Discovery</td>
</tr>
<tr>
<td>To define classes and detect new and old instances that lie outside the classes.</td>
<td>Deviation Detection</td>
<td>Statistics, Visualization, Genetic Algorithms</td>
</tr>
</tbody>
</table>

**Configuring the KDD Server**

- Data mining mechanisms are not application-specific, they depend on the target knowledge type
- The application area impacts the type of knowledge you are seeking, so the application area guides the selection of data mining mechanisms that will be hosted on the KDD server.

**To configure a KDD server**

- Select an application area
- Select (or build) a data source
- Select N knowledge types (types of questions you will ask)
- For each knowledge type Do
  - Select 1 or more mining tools for that knowledge type

**Example:**

Application area: marketing
Data Source: data warehouse of current customer info
Knowledge types: classify current customers, predict future sales, predict new customers
Data Mining Tools: decision trees, and neural networks

©2002 Denise Ecklund

KDD-9
Data Mining Example - Database Segmentation

- Given: a coded database with 1 million records on subscriptions to five types of magazines
- Goal: to define a classification of the customers that can predict what types of new customers will subscribe to which types of magazines

<table>
<thead>
<tr>
<th>Client#</th>
<th>Age</th>
<th>Income</th>
<th>Credit</th>
<th>Car Owner</th>
<th>House Owner</th>
<th>Home Area</th>
<th>Car Magaz</th>
<th>House Magaz</th>
<th>Sports Magaz</th>
<th>Music Magaz</th>
<th>Comic Magaz</th>
</tr>
</thead>
<tbody>
<tr>
<td>2303</td>
<td>20</td>
<td>18.5</td>
<td>17.8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2309</td>
<td>25</td>
<td>36.0</td>
<td>26.6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2313</td>
<td>42</td>
<td>24.0</td>
<td>20.8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2327</td>
<td>31</td>
<td>48.2</td>
<td>24.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2328</td>
<td>56</td>
<td>94.0</td>
<td>40.0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2330</td>
<td>43</td>
<td>18.0</td>
<td>7.0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2333</td>
<td>22</td>
<td>36.3</td>
<td>15.8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Fall 2003 presented by Naci Akkøk, based upon originals by Denise Ecklund, ©2002

Decision Trees for DB Segmentation

- Which attribute(s) best predicts which magazine(s) a customer subscribes to (sensitivity analysis)
  - Attributes: age, income, credit, car-owner, house-owner, area
- Classify people who subscribe to a car magazine

Only 1% of the people over 44.5 years of age buys a car magazine
Age > 44.5 ...... 99%
Age <= 44.5 ...... 100%
No one over 48.5 years of age buys a car magazine
Age > 48.5 ...... 100%
Age <= 48.5 ...... 92%
62% of the people under 44.5 years of age buys a car magazine
Income > 34.5 ...... 100%
Income <= 34.5 ...... 47%
Age <= 31.5 ...... 46%
Age > 31.5 ...... 0%
Everyone in this DB who has an income of less than $34, 500 AND buys a car magazine is less than 31.5 years of age

Target advertising segment for car magazines

Fall 2003 presented by Naci Akkøk, based upon originals by Denise Ecklund, ©2002

©2002 Denise Ecklund
Neural Networks

- Input nodes are connected to output nodes by a set of hidden nodes and edges
- Inputs describe DB instances
- Outputs are the categories we want to recognize
- Hidden nodes assign weights to each edge so they represent the weight of relationships between the input and the output of a large set of training data

<table>
<thead>
<tr>
<th>Car</th>
<th>House</th>
<th>Sports</th>
<th>Music</th>
<th>Comic</th>
</tr>
</thead>
</table>

Input Nodes

Hidden Layer Nodes

Output Nodes

Training and Mining

- Training Phase (learning)
  - Code all DB data as 1’s and 0’s
  - Set all edge weights to prob = 0
  - Input each coded database record
  - Check that the output “is correct”
  - The system adjusts the edge weights to get the correct answer

- Mining Phase (recognition)
  - Input a new instance coded as 1’s and 0’s
  - Output is the classification of the new instance

- Issues:
  - Training sample must be large to get good results
  - Network is a “black box”, it does not tell “why” an instance gives a particular output (no theory)
An Explosion of Mining Results

- Data Mining tools can output thousands of rules and associations (collectively called *patterns*)

When is a pattern *interesting*? Metrics of *interestingness*

1) If it can be understood by humans

   - **Simplicity**: *Rule Length* for
     \[(A \Rightarrow B)\] is the number of conjunctive conditions or the number of attributes in the rule

2) The pattern is strong

   - **Confidence**: *Rule Strength* for
     \[(A \Rightarrow B)\] is the conditional probability that A implies B
     \[
     \frac{\#\text{recs with } A \& B}{\#\text{recs with } A}
     \]

3) It is potentially useful for your business needs

   - **Support**: *Support* for \[(A \Rightarrow B)\] is the number or percentage of DB records that include A and B

4) It is novel

   - **Novelty**: *Rule Uniqueness* for
     \[(A \Rightarrow B)\] means no other rule includes or implies this rule

---

Genetic Algorithms

- Based on Darwin’s theory of ”survival of the fittest”
  - Living organisms reproduce, individuals evolve/mutate, individuals survive or die based on fitness
- The output of a genetic algorithm is the set of ”fittest solutions” that will survive in a particular environment
- The input is an initial set of possible solutions
- The process
  - Produce the next generation (by a cross-over function)
  - Evolve solutions (by a mutation function)
  - Discard weak solutions (based on a fitness function)
Classification Using Genetic Algorithms

- Suppose we have money for 5 marketing campaigns, so we wish to cluster our magazine customers into 5 target marketing groups
- Customers with similar attribute values form a cluster (assumes similar attributes => similar behavior)
- Preparation:
  - Define an encoding to represent solutions (i.e., use a character sequence to represent a cluster of customers)
  - Create 5 possible initial solutions (and encode them as strings)
  - Define the 3 genetic functions to operate on a cluster encoding
    - Cross-over(), Mutate(), Fitness_Test()

Genetic Algorithms - Initialization

- Define 5 initial solutions
  - Use a subset of the database to create a 2-dim scatter plot
    - Map customer attributes to 2 dimensions
  - Divide the plot into 5 regions
  - Calculate an initial solution point (guide point) in each region
    - Equidistant from region lines
- Define an encoding for the solutions
  - Strings for the customer attribute values
  - Encode each guide point
Genetic Algorithms – Evolution

- Cross-over function
  Create 2 children
  Take 6 attribute values from one parent and 4 from the other

- Mutate function
  Randomly switch several attribute values from values in the sample subspace

- Fitness function:
  Average distance between the solution point and all the points in the sample subspace

<table>
<thead>
<tr>
<th>1st Gen</th>
<th>2nd Gen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution 1</td>
<td>Fitness 13</td>
</tr>
<tr>
<td>30 10</td>
<td>35 18</td>
</tr>
<tr>
<td>40 20</td>
<td>42 20</td>
</tr>
<tr>
<td>55 28</td>
<td>55 13</td>
</tr>
<tr>
<td>60 50</td>
<td>40 48</td>
</tr>
<tr>
<td>70 16</td>
<td>75 16</td>
</tr>
<tr>
<td>Fitness 14.5</td>
<td>Fitness 17</td>
</tr>
<tr>
<td>10 90</td>
<td>10 90</td>
</tr>
<tr>
<td>33 10</td>
<td>33 10</td>
</tr>
<tr>
<td>48 28</td>
<td>37 28</td>
</tr>
<tr>
<td>60 50</td>
<td>30 50</td>
</tr>
<tr>
<td>26 50</td>
<td>26 50</td>
</tr>
<tr>
<td>75 16</td>
<td>75 16</td>
</tr>
<tr>
<td>Fitness 20</td>
<td>Fitness 30</td>
</tr>
<tr>
<td>57 80</td>
<td>57 80</td>
</tr>
<tr>
<td>57 80</td>
<td>57 80</td>
</tr>
<tr>
<td>Fitness 33</td>
<td></td>
</tr>
</tbody>
</table>

- Stop when the solutions change very little

Selecting a Data Mining Mechanism

- Multiple mechanisms can be used to answer a question
  – select the mechanism based on your requirements

<table>
<thead>
<tr>
<th>Decision Trees</th>
<th>Neural Networks</th>
<th>Genetic Algs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many recs</td>
<td>Many attrs</td>
<td>String values</td>
</tr>
<tr>
<td>Learn rules</td>
<td>Learn values</td>
<td>Learn</td>
</tr>
<tr>
<td>Est stat</td>
<td>Learn incre</td>
<td>L-Perf</td>
</tr>
<tr>
<td>Signif</td>
<td>Signif</td>
<td>L-Perf</td>
</tr>
<tr>
<td>Disk</td>
<td>Disk</td>
<td>A-Perf</td>
</tr>
<tr>
<td>Cpu</td>
<td>Cpu</td>
<td>A-Perf</td>
</tr>
</tbody>
</table>

Quality of Input

<table>
<thead>
<tr>
<th>Good</th>
<th>Avg</th>
<th>Poor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality of Output</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning Performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Performance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
So ...

Artificial Intelligence is fun, ...
but what does this have to do with database?

- Data mining is just another database application
- Data mining applications have requirements
- The database system can help mining applications meet their requirements

So, what are the challenges for data mining systems and how can the database system help?

---

Database Support for DM Challenges

<table>
<thead>
<tr>
<th>Data Mining Challenge</th>
<th>Database Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Support many data mining mechanisms in a KDD system</td>
<td>• Support multiple DB interfaces (for different DM mechanisms)</td>
</tr>
<tr>
<td>• Support interactive mining and incremental mining</td>
<td>• Intelligent caching and support for changing views over query results</td>
</tr>
<tr>
<td>• Guide the mining process by integrity constraints</td>
<td>• Make integrity constraints query-able (meta-data)</td>
</tr>
<tr>
<td>• Determine usefulness of a data mining result</td>
<td>• Gather and output runtime statistics on DB “support”, “confidence”, and other metrics</td>
</tr>
<tr>
<td>• Help humans to better understand mining results (e.g., visualization)</td>
<td>• Prepare output data for selectable presentation formats</td>
</tr>
</tbody>
</table>
Database Support for DM Challenges

Data Mining Challenge | Database Support
--- | ---
• Accurate, efficient, and flexible methods for data cleaning and data encoding | • "Programmable" data warehouse tools for data cleaning and encoding; Fast, runtime re-encoding
• Improved performance for data mining mechanisms | • Parallel data retrieval and support for incremental query processing
• Ability to mine a wide variety of data types | • New indexing and access methods for non-traditional data types
• Support web mining | • Extend XML and WWW database technology to support large, long running queries
• Define a data mining query language | • Extend SQL, OQL and other interfaces to support data mining

Commercial Data Mining Products and Tools

• Some DB companies with Data Mining products:
  – Oracle – Oracle 9i, with BI-Beans (an OLAP toolset)
  – IBM – "Data Miner for Data" and "Data Miner for Text"
  – NCR – TeraMiner™ for the TeraData™ warehouse

• Companies with Data Mining tools
  – COGNOS – “Scenario”, a set of DM and OLAP tools
  – Elseware – ”Classpad” (classification) and ”Previa” (prediction)
  – Logic Programming Associates, Ltd. – ”Datamite” (clustering)
  – Prudential System Software GmbH – credit card fraud
  – RedShed Software – ”Dowser” (association discovery)

http://www.kdnuggets.com/companies/products.html
Conclusions

- To support data mining, we should
  - Enhance database technologies developed for
    - Data warehouses
    - Very large database systems
    - Object-oriented (navigational) database systems
    - View Management mechanisms
    - Heterogeneous databases
  - Create new access methods based on mining access
  - Develop query languages or other interfaces to better support data mining mechanisms
  - Select and integrate the needed DB technologies

Studying for the exam?
Send questions to the appropriate lecturer.

ellenmk@ifi.uio.no
ragnarn@ifi.uio.no
nacia@ifi.uio.no

There will be a review session 19.11.2003, hopefully with questions and answers.

For the success of the last session, please send questions in advance, preferably by mail and preferably to the right lecturer.