Knowledge Discovery in Databases (KDD) - Data Mining (DM)

Vera Goebel
Department of Informatics, University of Oslo
2016
Why Data Mining / KDD?

Applications

• Banking: loan/credit card approval
  – predict good customers based on old customers

• Customer relationship management:
  – identify those who are likely to leave for a competitor.

• Targeted marketing:
  – identify likely responders to promotions

• Fraud detection: telecommunications, financial transactions
  – from an online stream of event identify fraudulent events

• Manufacturing and production:
  – automatically adjust knobs when process parameter changes
Applications (continued)

• Medicine: disease outcome, effectiveness of treatments
  – analyze patient disease history: find relationship between diseases
• Molecular/Pharmaceutical: identify new drugs
• Scientific data analysis:
  – identify new galaxies by searching for sub clusters
• Web site/store design and promotion:
  – find affinity of visitor to pages and modify layout
Why Now?

• Data is being produced
• Data is being warehoused
• The computing power is available
• The computing power is affordable
• The competitive pressures are strong
• Commercial products are available
Why Integrate DM/KDD with DBS?

Copy → Mine → Models

Extract

Data

Consistency?
Integration Objectives

• Avoid isolation of querying from mining
  – Difficult to do “ad-hoc” mining
• Provide simple programming approach to creating and using DM models
• Make it possible to add new models
• Make it possible to add new, scalable algorithms

Analysts (users)  DM Vendors
Why Data Mining?
Can we solve all problems with DM?

• **Four questions to be answered:**

  • Can the problem clearly be defined?
  • Does potentially meaningful data exists?
  • Does the data contain hidden knowledge or useful only for reporting purposes?
  • Will the cost of processing the data be less then the likely increase in profit from the knowledge gained from applying any data mining project?
Definitions: DM = KDD

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

• Data mining is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data. Process of semi-automatically analyzing large databases to find patterns that are:
  – Valid: The patterns hold in general.
  – Novel: We did not know the pattern beforehand.
  – Useful: We can devise actions from the patterns.
  – Understandable: We can interpret and comprehend the patterns.

• Why? 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
  • …
KDD Process

• Problem formulation
• Data collection
  – subset data: sampling might hurt if highly skewed data
  – feature selection: principal component analysis, heuristic search
• Pre-processing: cleaning
  – name/address cleaning, different meanings (annual, yearly), duplicate removal, supplying missing values
• Transformation:
  – map complex objects e.g. time series data to features e.g. frequency
• Choosing mining task and mining method
• Result evaluation and visualization

Knowledge discovery is an iterative process
KDD Process Steps (detailed)

• **1. Goal identification:**
  – Define problem
  – relevant prior knowledge and goals of application

• **2. Creating a target data set:** data selection

• **3. Data preprocessing:** (60%-80% of effort!) \(\rightarrow\) good Data Warehouse needed!
  – removal of noise or outliers
  – strategies for handling missing data fields
  – accounting for time sequence information

• **4. Data reduction and transformation:**
  – Find useful features, dimensionality/variable reduction, invariant representation

• **5. Data Mining:**
  – Choosing functions of data mining:
    • summarization, classification, regression, association, clustering
  – Choosing the mining algorithm(s):
    • which models or parameters
  – Search for patterns of interest

• **6. Presentation and evaluation:**
  – visualization, transformation, removing redundant patterns, etc.

• **7. Taking action:**
  – incorporating into the performance system
  – documenting
  – reporting to interested parties
What can computers learn?

4 levels of learning [Merril & Tennyson 1977]

• **Facts**: simple statement of truth
• **Concepts**: set of objects, symbols, or events grouped together -> share certain properties
• **Procedures**: step-by-step course of action to achieve goal
• **Principles**: highest level of learning, general truths or laws that are basic to other truths

DM tool dictates form of learning concepts:

• Human understandable: trees, rules,
• Black-box: networks, mathematical equations
Data Warehouse (DW)

- Data Mining needs multi-dimensional data input

DW: Repository of multiple heterogeneous data sources, organized under a unified multi-dimensional schema at a single site in order to facilitate management decision making. Data warehouse technology includes:

- Data cleaning
- Data integration
- On-Line Analytical Processing (OLAP): Techniques that support multidimensional analysis and decision making with the following functionalities
  - summarization
  - consolidation
  - aggregation
  - view information from different angles
- but additional data analysis tools are needed for
  - classification
  - clustering
  - characterization of data changing over time
Where SQL Stops Short

- Successful KDD uses the entire tool hierarchy

- Creates new **knowledge** in the form of predictions, classifications, associations, time-based patterns

- Analyzes a multi-dimensional view of existing knowledge

- Extends the DB schema to a limited, multi-dimensional view

- DB schema describes the structure you already know

- Data Mining Tools

- OLAP Tools

- DSS-Extended SQL Queries

- Complex SQL Queries

- Database

- Observe

- Analyze

- Theoritize

- Predict
OnLine Analytic 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 Selection and Extraction
- Data Cleaning
- Data Enrichment
- Data Coding
- Data Mining
- Result Reporting

Commonly part of data warehouse preparation

Additional preparation for KDD

Data Preparation Phases

- 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
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.

**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

**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
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
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
     • Example: 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 Strategies

• Unsupervised Learning (Clustering)

• Supervised Learning:
  – Classification
  – Estimation
  – Prediction

• Market Basket Analysis
Two Styles of Data Mining

• **Descriptive data mining**
  – characterize the general properties of the data in the database
  – finds patterns in data
  – user determines which ones are important

• **Predictive data mining**
  – perform inference on the current data to make predictions
  – we know what to predict

• **Not mutually exclusive**
  – used together
  – descriptive → predictive

• E.g. customer segmentation – descriptive by clustering
• Followed by a risk assignment model – predictive
Supervised vs. Unsupervised Learning

- **Supervised learning** (classification, prediction)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set

- **Unsupervised learning** (summarization, association, clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data
Descriptive Data Mining (1)

- Discovering new patterns inside the data
- Used during the data exploration steps
- Typical questions answered by descriptive data mining
  - what is in the data
  - what does it look like
  - are there any unusual patterns
  - what does the data suggest for customer segmentation
- users may have no idea
  - which kind of patterns may be interesting
Descriptive Data Mining (2)

- Patterns at various granularities:
  - geography
    - country - city - region - street
  - student
    - university - faculty - department - minor

- Functionalities of descriptive data mining:
  - Clustering
    - Example: customer segmentation
  - Summarization
  - Visualization
  - Association
    - Example: market basket analysis
Black-box Model

X: vector of independent variables or inputs
Y = f(X): an unknown function
Y: dependent variables or output
   a single variable or a vector

inputs \[\rightarrow\] Model \[\rightarrow\] Y output

The user does not care what the model is doing
it is a black box
interested in the accuracy of its predictions
Predictive Data Mining

- Using known examples the model is trained
  - the unknown function is \textit{learned from data}
- the more data with known outcomes is available
  - the better the predictive power of the model
- Used to predict outcomes whose inputs are known but the output values are not realized yet
- Never 100% accurate
- The performance of a model on past data is not important
  - to predict the known outcomes
- Its performance on unknown data is much more important
Typical questions answered by predictive models

• Who is likely to respond to our next offer
  – based on history of previous marketing campaigns

• Which customers are likely to leave in the next six months

• What transactions are likely to be fraudulent
  – based on known examples of fraud

• What is the total amount spending of a customer in the next month
Data Mining Functionalities (1)

• **Concept description: characterization and discrimination**
  – Generalize, summarize, and contrast data characteristics, e.g., big spenders vs. budget spenders

• **Association (correlation and causality)**
  – Multi-dimensional vs. single-dimensional association
  – age(X, “20..29”) ^ income(X, “20..29K”) ->
    buys(X, “PC”) [support = 2%, confidence = 60%]
  – contains(T, “computer”) -> contains(x, “software”) [1%, 75%]
Data Mining Functionalities (2)

- **Classification and Prediction**
  - Finding models (functions) that describe and distinguish classes or concepts for future prediction
  - E.g., classify people as healthy or sick, or classify transactions as fraudulent or not
  - Methods: decision-tree, classification rule, neural network
  - Prediction: Predict some unknown or missing numerical values

- **Cluster analysis**
  - Class label is unknown: Group data to form new classes, e.g., cluster customers of a retail company to learn about characteristics of different segments
  - Clustering based on the principle: maximizing the intra-class similarity and minimizing the interclass similarity
Data Mining Functionalities (3)

• **Outlier analysis**
  – Outlier: a data object that does not comply with the general behavior of the data
  – It can be considered as noise or exception but is quite useful in fraud detection, rare events analysis

• **Trend and evolution analysis**
  – Trend and deviation: regression analysis
  – Sequential pattern mining: click stream analysis
  – Similarity-based analysis

• **Other pattern-directed or statistical analyses**
# 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</td>
<td>Deviation Detection</td>
<td>Statistics, Visualization, Genetic Algorithms</td>
</tr>
<tr>
<td>classes.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Classification (Supervised Learning)

Goal:
Learn a function that assigns a record to one of several predefined classes.

Requirements on the model:
- High accuracy
- Understandable by humans, interpretable
- Fast construction for very large training database

Example: Given old data about customers and payments, predict new applicant’s loan eligibility.

Previous customers
- Age
- Salary
- Profession
- Location
- Customer type

Classifier

Decision rules
- Salary > 5 L
- Prof. = Exec

New applicant’s data
Good/bad
Classification (cont.)

- Decision trees are one approach to classification.

- Other approaches include:
  - Linear Discriminant Analysis
  - $k$-nearest neighbor methods
  - Logistic regression
  - Neural networks
  - Support Vector Machines
Classification methods

- **Goal**: Predict class $C_i = f(x_1, x_2, \ldots, X_n)$
- Regression: (linear or any other polynomial)
  - $a \cdot x_1 + b \cdot x_2 + c = C_i$.
- Nearest neighbour
- Decision tree classifier: divide decision space into piecewise constant regions.
- Probabilistic/generative models
- Neural networks: partition by non-linear boundaries
Nearest neighbor

• Define proximity between instances, find neighbors of new instance and assign majority class
• Case based reasoning: when attributes are more complicated than real-valued

• Pros
  + Fast training

• Cons
  – Slow during application.
  – No feature selection.
  – Notion of proximity vague
Decision trees

- Tree where internal nodes are simple decision rules on one or more attributes and leaf nodes are predicted class labels.

![Decision Tree Diagram]

- Salary < 1 M
  - Prof = teacher
    - Good
  - Age < 30
    - Bad
    - Bad
    - Good
Decision tree classifiers

• Widely used learning method
• Easy to interpret: can be represented as if-then-else rules
• Approximates function by piece wise constant regions
• Does not require any prior knowledge of data distribution, works well on noisy data.
• Has been applied to:
  – classify medical patients based on the disease,
  – equipment malfunction by cause,
  – loan applicant by likelihood of payment.
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>0</td>
<td>1</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>0</td>
<td>1</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>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
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

```
Car Magaz

Age > 44.5  --->  99%

Age <= 44.5  --->  38%

62% of the people under 44.5 years of age buys a car magazine

Only 1% of the people over 44.5 years of age buys a car magazine

Age > 48.5  --->  100%

Age <= 48.5  --->  92%

Income > 34.5  --->  100%

Income <= 34.5  --->  47%

Age <= 31.5  --->  0%

No one over 48.5 years of age buys a car magazine

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
```
Pros and Cons of decision trees

- **Pros**
  + Reasonable training time
  + Fast application
  + Easy to interpret
  + Easy to implement
  + Can handle large number of features

- **Cons**
  - Cannot handle complicated relationship between features
  - Simple decision boundaries
  - Problems with lots of missing data
Clustering (Unsupervised Learning)

- Unsupervised learning when old data with class labels not available e.g. when introducing a new product.
- Group/cluster existing customers based on time series of payment history such that similar customers in same cluster.
- Key requirement: Need a good measure of similarity between instances.
- Identify micro-markets and develop policies for each
Applications

• Customer segmentation e.g. for targeted marketing
  – Group/cluster existing customers based on time series of payment history such that similar customers in same cluster.
  – Identify micro-markets and develop policies for each

• Collaborative filtering:
  – group based on common items purchased

• Text clustering

• Compression
Distance Functions

• Numeric data: euclidean, manhattan distances
• Categorical data: 0/1 to indicate presence/absence followed by
  – Hamming distance (# dissimilarity)
  – Jaccard coefficients: #similarity in 1s/(# of 1s)
  – data dependent measures: similarity of A and B depends on co-occurrence with C
• Combined numeric and categorical data:
  – weighted normalized distance
Clustering Methods

• Hierarchical clustering
  – agglomerative vs. divisive
  – single link vs. complete link

• Partitional clustering
  – distance-based: K-means
  – model-based
  – density-based
Association Rules

- Given set $T$ of groups of items
- Example: set of item sets purchased
- Goal: find all rules on itemsets of the form $a \rightarrow b$ such that
  - support of $a$ and $b >$ user threshold $s$
  - conditional probability (confidence) of $b$ given $a >$ user threshold $c$
- Example: Milk $\rightarrow$ bread
- Purchase of product $A \rightarrow$ service $B$

<table>
<thead>
<tr>
<th>$T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milk, cereal</td>
</tr>
<tr>
<td>Tea, milk</td>
</tr>
<tr>
<td>Tea, rice, bread</td>
</tr>
<tr>
<td>cereal</td>
</tr>
</tbody>
</table>
Variants

• High confidence may not imply high correlation
• Use correlations. Find expected support and large departures from that interesting.
  – see statistical literature on contingency tables.
• Still too many rules, need to prune...
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.
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)*
Neural Network

- Set of nodes connected by directed weighted edges

**Basic NN unit**

\[ o = \sigma \left( \sum_{i=1}^{n} w_i x_i \right) \]

\[ \sigma(y) = \frac{1}{1 + e^{-y}} \]

**A more typical NN**
Neural Networks

- Useful for learning complex data like handwriting, speech and image recognition
Pros and Cons of Neural Network

· Pros
  + Can learn more complicated class boundaries
  + Fast application
  + Can handle large number of features

· Cons
  - Slow training time
  - Hard to interpret
  - Hard to implement: trial and error for choosing number of nodes

Conclusion: Use neural nets only if decision-trees/NN fail
Bayesian Learning

• Assume a probability model on generation of data.

• predicted class : \( c = \max_{c_j} p(c_j \mid d) = \max_{c_j} \frac{p(d \mid c_j)p(c_j)}{p(d)} \)

• Apply bayes theorem to find most likely class as:

\[
c = \max_{c_j} \frac{p(c_j)}{p(d)} \prod_{i=1}^{n} p(a_i \mid c_j)
\]

• Naïve bayes: Assume attributes conditionally independent given class value

• Easy to learn probabilities by counting,

• Useful in some domains e.g. text
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

- Stop when the solutions change very little
Choosing DM Technique

• Is learning supervised or unsupervised?
• Clear explanation needed about the relationships in the data?
• Is there 1 set of input attributes and 1 set of output attributes, or can attributes interact with one another in several ways?
• Is the input data categorical, numeric, or both?
• If learning is supervised, is there 1 or multiple output attributes? Categorical or numeric output attributes?

Still several choices …

• Do we know the distribution of the data?
• Do we know which attributes best define the data to be modeled?
• Does the data contain several missing values?
• Is time an issue?
• Which technique is most likely to give best classification accuracy?
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></th>
<th>Many recs</th>
<th>Many attrs</th>
<th>Numeric values</th>
<th>String values</th>
<th>Learn rules</th>
<th>Learn incre</th>
<th>Est stat</th>
<th>L-Perf disk</th>
<th>L-Perf cpu</th>
<th>A-Perf disk</th>
<th>A-Perf cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision Trees</strong></td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td><strong>Neural Networks</strong></td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td><strong>Genetic Algs</strong></td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Quality of Input</th>
<th>Quality of Output</th>
<th>Learning Performance</th>
<th>Application Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Good</strong></td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>Average</td>
<td>Average</td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td><strong>Poor</strong></td>
<td>Poor</td>
<td>Poor</td>
<td>Poor</td>
<td>Poor</td>
</tr>
</tbody>
</table>
An Explosion of Mining Results

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

**When is a pattern *interesting*?**

1) If it can be understood by humans

2) The pattern is strong (i.e., valid for many new data records)

3) It is potentially useful for your business needs

4) It is novel (i.e., previously unknown)

**Metrics of *interestingness***

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

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

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

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

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

• Around 20 to 30 data mining tool vendors (trend ++)

• Major tool players:
  – Clementine
  – IBM Intelligent Miner
  – SGI MineSet
  – SAS Enterprise Miner

• All pretty much the same set of tools

• Many embedded products:
  – fraud detection
  – electronic commerce applications
  – health care
  – customer relationship management: Epiphany
Vertical Integration: Mining on the Web

• Web log analysis for site design:
  – what are popular pages,
  – what links are hard to find.

• Electronic stores sales enhancements:
  – recommendations, advertisement:
  – Collaborative filtering: Net perception, Wisewire
  – Inventory control: what was a shopper looking for and could not find.
OLAP Mining Integration

• OLAP (On Line Analytical Processing)
  – Fast interactive exploration of multi-dimensional aggregates
  – Heavy reliance on manual operations for analysis
  – Tedium and error-prone on large multidimensional data
• Ideal platform for vertical integration of mining but needs to be interactive instead of batch.
SQL/MM: Data Mining

• A collection of classes that provide a standard interface for invoking DM algorithms from SQL systems.

• Four data models are supported:
  – Frequent itemsets, association rules
  – Clusters
  – Regression trees
  – Classification trees
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