Data Mining: Methods and Applications

in Engineering and Management

Detlef Nauck

Faculty of Computer Science
University of Magdeburg, Germany

Data Mining: Introduction

- More and more data is gathered due to progress in computers and databases
- Manual analysis, charts etc. are no longer feasible
- Scientifc progress stimulates needs and offers solutions
- Problem: Find *information nuggets* in vast amounts of data
  Solution: Knowledge Discovery in Databases (KDD)

Data Mining: Introduction (contd.)

- KDD: process of finding knowledge in data by "high level" application of data mining methods
- Data mining is only one step of KDD
- Blind application of data mining methods can be dangerous as invalid patterns might be detected
- Data mining is not a single method – "data mining" refers to various methods

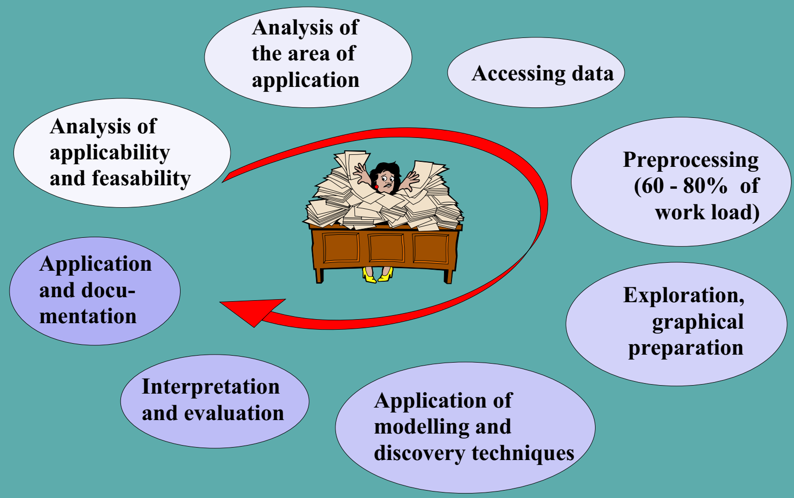
KDD: Knowledge Discovery in Databases

KDD is the non-trivial process of identifying

- valid,
- novel,
- potentially useful,
- ultimately understandable,

patterns in data (Fayyad, Piatetsky-Shapiro & Smyth, 1996).

KDD: Knowledge Discovery in Databases



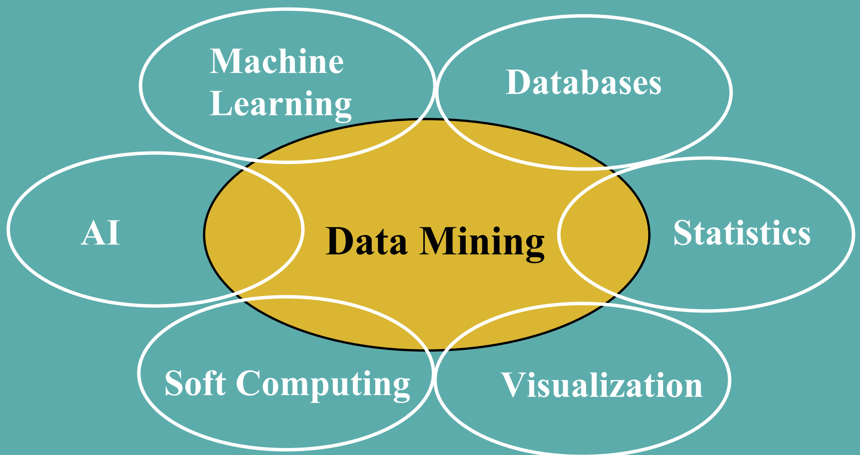
Data Mining

Data Mining is: The application of various methods of analysis and learning to discover knowledge in data

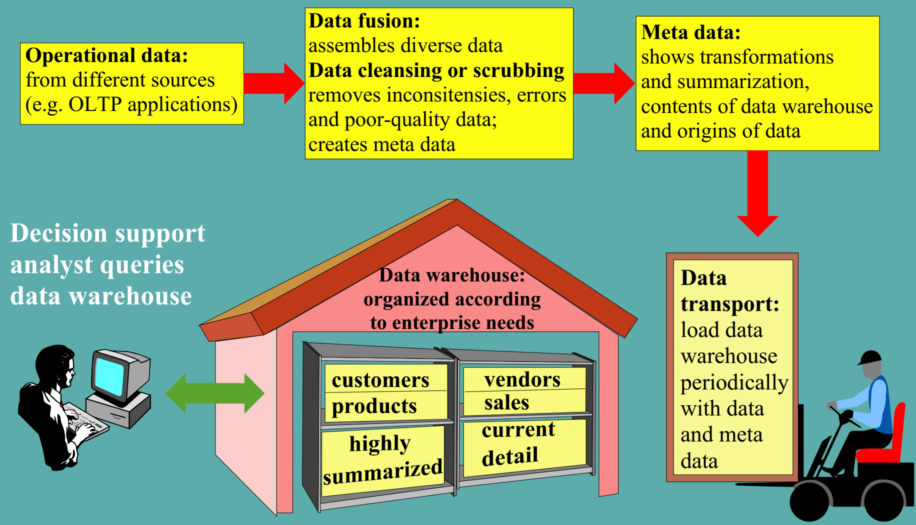
or: Torture your data until they confess.

Data Mining is not: Ad-hoc queries, reports, data warehousing, OLAP software agents, XPS, alerting,...

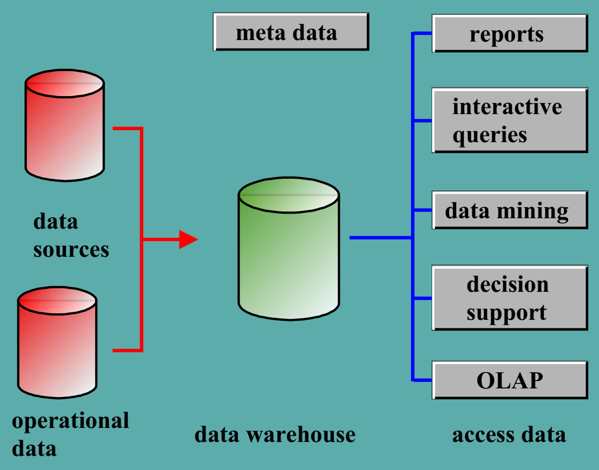
Data Mining is Interdisciplinary



Data Warehouse



Data Mining as Query Tool



Data Mining - Caveats

- A lot of hype out there: Data Mining is a buzzword (yesterday C++ and statistics, today Java and data mining)
- There is a trade-off between usability and accuracy
- Most of the software aims at special applications. There are a lot of tools (over 50 in 1997)
- Severe errors occur if complex methods are not used correctly or are not explained to the end user
- The number and variety of data have more of an effect on the accuracy than the selected mining method

Detlef Nauck Data Mining Tutorial 1998

Data Mining Tasks

- Classification
  Is this a good customer ?
- Concept Description
  What makes a good customer ? (age, income, ...)
- Segmentation (Clustering)
  What kind of customers do I have ?

Detlef Nauck Data Mining Tutorial 1998

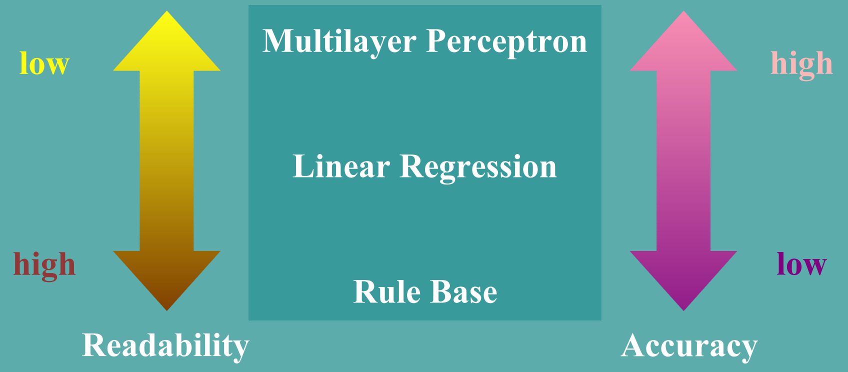
Data Mining Tasks

- Prediction
  What will be the demand for my product ?
- Dependency Analysis
  80% of customers who buy diapers buy beer, too
- Deviation Analysis
  Why do we sell less insurances in Cleveland ?

Detlef Nauck Data Mining Tutorial 1998

Model Selection

How readable / accurate is the selected model?



Imperfect Information

$x = 0.127$	precise (crisp)
$x \in [0, 1]$	imprecise (interval valued)
x is *approximately zero*	vague (fuzzy)
$x = ?$	missing value

Uncertain Information

I am 90% certain that Peter is married

My belief that Peter is married is 0.9

Usually modeled by (subjective) probabilities
e.g. p(Peter is married) = 0.9

It can become worse:

I am 90% certain that Peter is tall
I am very certain that Peter is tall

Scenario A

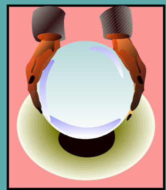
You are a marketing manager for an insurance company

- The company sells liability insurance, personal effects insurance, life/accident insurance and car insurance.
- There are many cancelations / new contracts in car insurance each year.
- There are 1 million customers
- Goals: Prevent cancellations, cross-selling.

Solution A1

Prevent cancelations:

- Use historic data to predict which customers are likely to cancel their contract within the next 3 months.
- Contact these customers (send sales representative, offer better rates or benefits, ...).
- How can we predict future behavior?



Solution A2

Cross-selling:

- Use historical data:

  - find car insurance customers who bought life or accident insurance, create a classifier.

  - classify new customers according to your findings and send them an offer (mailing).

  - same method as used for solution to prevent cancellations

Solution A3

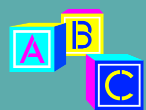
Cross-selling:

- If there is no historical data for this task:

  - find groups of customers, analyse and label them *(young parent, parsimonious, pensioners, ...)*

  - select groups that may be responsive to mailings *(pensioners don't buy life insurance, young parents do)*

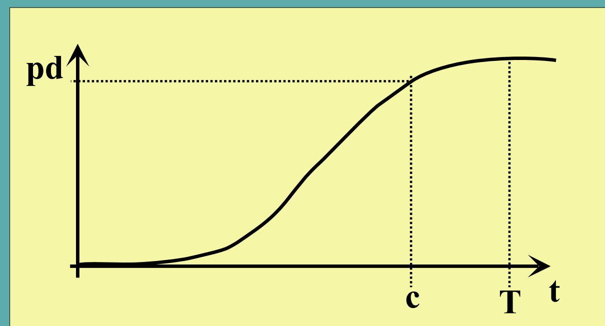
  - How to find groups, how many are there?



Scenario B

You are an engineer who wants to automate a process

There is a time-controlled process, but you want to tell from process data, when it is completed.



pd: process data

T: process is stopped

c: process is completed

Scenario B - Solution

Find the real end of the process

There is a database of (noisy) process data

Use this data to compute criteria to detect *c*.

Signal that *c* was reached by observing process data.

How can we find *c* in the process data?

Classical Statistical Approaches

- Discriminant Analysis
- Regression Analysis
- Cluster Analysis
- Bayesian Learning

Problem: Often only linear models are applied, because non-linear models are not understood or cannot be handled by the user.

Alternative Approaches

Machine Learning (ML, symbolic)

- Inductive Logic Programming
  creating propositional rule bases
- Conceptual Clustering
  clustering of symbolic data
- Instance Based Learning
  case based reasoning, detection of similar cases
- Decision Trees
  construct tree-based classifiers

Alternative Approaches

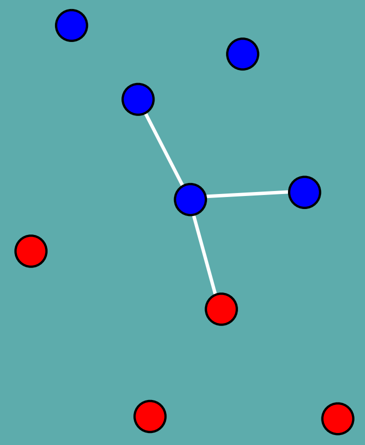
Soft Computing (SC, numerical)

- Neural Networks
  the final black box, can be very powerful
- Fuzzy Systems and Neuro-Fuzzy Systems
  based on linguistic (fuzzy) rules
- Evolutionary Computation
  parallel search algorithms, optimization
- Probabilistic Appraoches, Bayesian Networks
  dependencies modeled by probabilities

Classification

- Storing Known Cases: K-Nearest Neighbor
- Statistics: Discriminant Analysis, Logistic Regression
- Induction of Decision Trees
- Neural Networks: MLP or RBFN
- Fuzzy Classifier, Neuro-Fuzzy Classifier

Classification: K-Nearest Neighbor



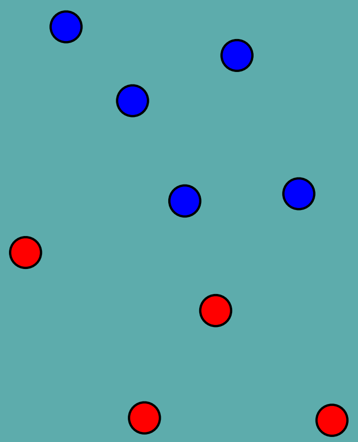
What color should the new ball have - red or blue?

Let the 3 nearest neighbors vote for it.

2 blue balls and 1 red ball: the new ball should be blue.

3-nearest neighbor classifier

Classification: K-Nearest Neighbor



Add the new case to the code book.

The code book can become very large (when to stop?)

Classifier simple to create, but classification takes long.

Possible: Store prototypes of clusters or mean values.

Classification: Statistics

Discriminant Analysis

Search for **linear model** that discriminates classes best (set of linear discriminants: $w_1x_1 + w_2x_2 + ... + w_nx_n$)

Dependent variables: continuous, normally distributed

Independent variable: categorical, more than 2 possible

Classes: identical covariance matrices (identical multivariate normal distributions)

Classification: Statistics

Logistic Regression

Use for **dichotomic** (binary) classifications

Less strict assumptions than discriminant analysis, it **does not rely on a multivariate normal distribution**

Linear model to predict class

Independent variables: categorical or continuous

Dependent variable: categorical, dichotom

Classification: Induction of Decision Trees

Machine Learning (ML) Approach

A decision tree is a **tree-like classifier** that can be **interpreted by rules**

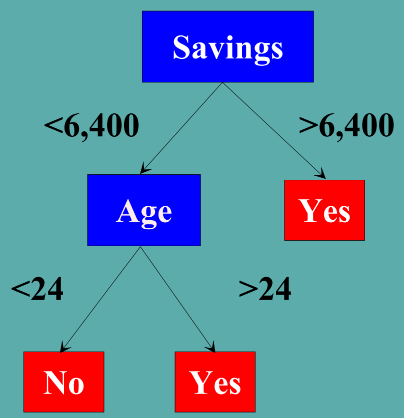
Inner nodes of tree: testing attributes

Leaf nodes: class labels

Idea: use an information theoretic measure to select "best" attributes first.

Classification: Decision Trees

Age	Income	Savings	Response
45	3,000	10,000	Yes
20	2,500	2,800	No
52	5,700	150,000	Yes
27	2,800	800	Yes
.	.	.	.
.	.	.	.



If Savings > 6,400 then send information
If Savings < 6,400 and Age > 24 then send information

Classification: Decision Trees

Goal: Create smallest tree, each leaf node should represent many cases.

Problem: NP-hard, therefore greedy algorithm (heuristics)

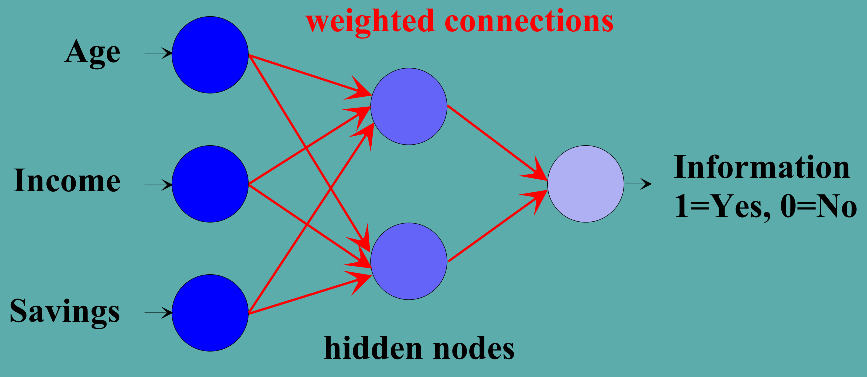
ID3 (Quinlan): information gain

C4.5 (Quinlan): information gain ratio

CART: Classification and Regression Trees

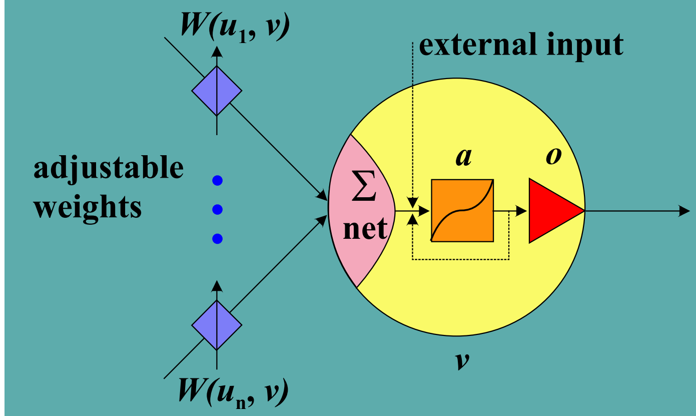
CHAID: Chi Square Automatic Interaction Detection.

Classification: Neural Networks



Non-linear model, universal function approximator, connection weights found by *"learning"*.

Neural Networks: Artificial Neuron



net = network (internal) input

a = activation

o = output

Classification: Neural Networks

Multilayer feedforward network with hidden layers

Nodes receive weighted sum as input

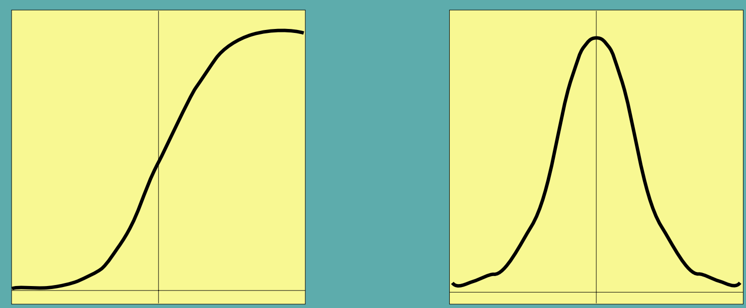
Hidden (and output) nodes use non-linear transfer functions

Multilayer Perceptron (MLP):
 s-shaped (sigmoid) function

Radial Basis Function Network (RBFN):
 bell-shaped (Gaussian) function

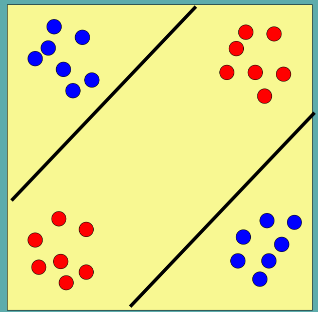
Classification: Neural Networks

Common activation functions in neural networks

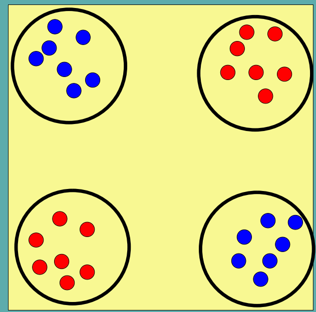


MLP: sigmoid RBFN: Gaussian

Classification: Neural Networks

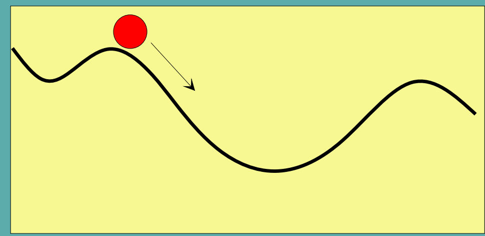


MLP:
global classification
using hyperplanes



RBFN:
local classification
using hyperellipsoids

Classification: Neural Networks



Learning in NN:
estimate weights
iteratively by
gradient descent.

Learning method: **Error Backpropagation (BP)** or variations like **Resilient Propagation** (RPROP: adaptive learning rate for each weight, just sign of gradient used, faster than BP)

Problems: local minima, oscillations, can be time-consuming

Classification: Neural Networks

NN are called *model-free estimators*
(actually they represent a very general model, but there is no interpretation of model parameters)

NN do not rely on any special distribution of the data

NN cannot be interpreted (the ultimate black box)

Parameters of the NN are hard to determine without proper experience (e.g. how many hidden units?)

NN can outperform other methods,
but there is no guarantee

Classification: Fuzzy Systems

Classification with linguistic (fuzzy) rules, e.g. mailing:

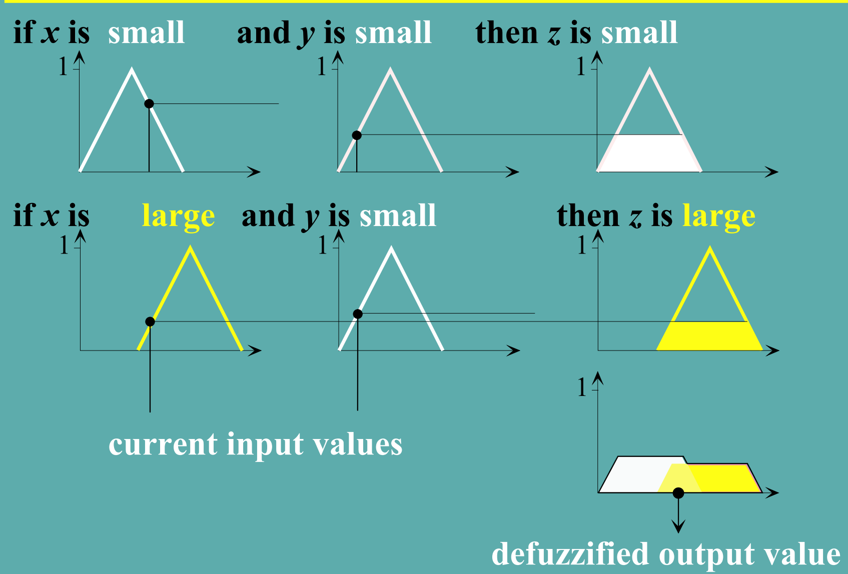
if age is *medium* and income is *high* then send informtion

if age is *young* and income is *low* then don't send information

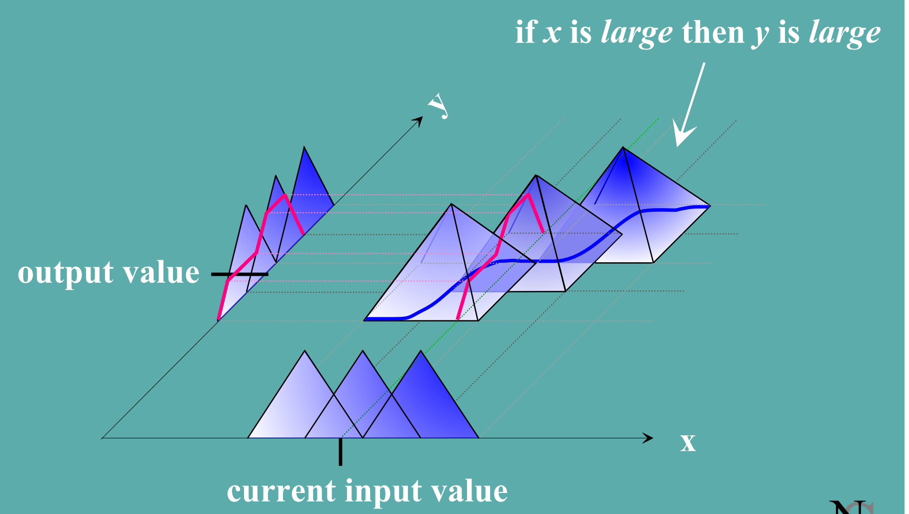
Advantages:

- simple model, easy to understand and to apply
- a case can belong to several classes to different degrees

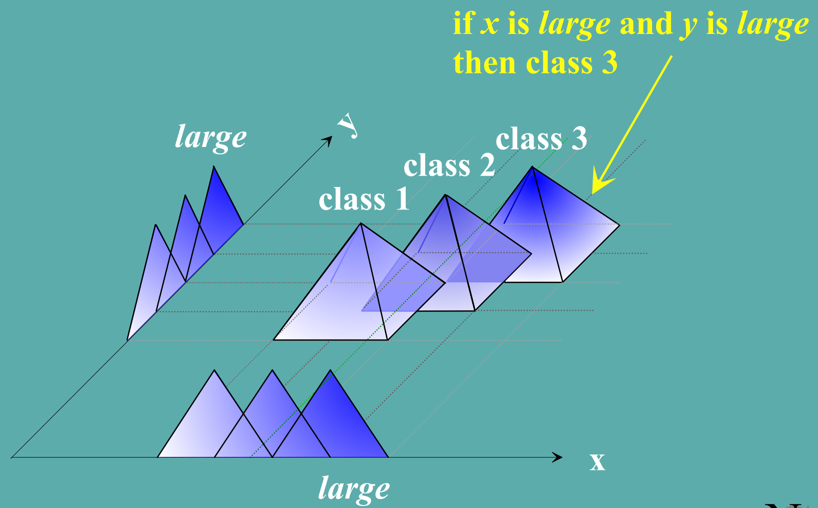
Function Approximation with Fuzzy Rules



Function Approximation with Fuzzy Rules



Classification with Fuzzy Rules



Classification: Fuzzy Systems

How to derive a fuzzy system from data?

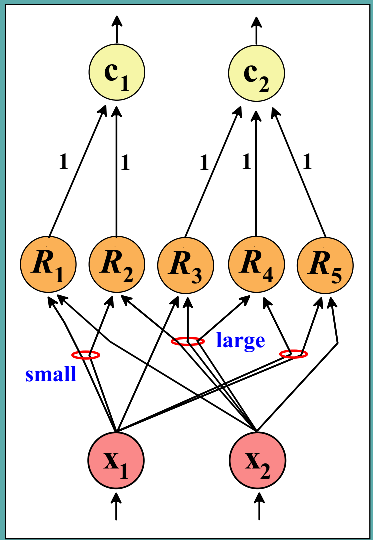
Fuzzy rules and fuzzy sets must be found.

Fuzzy rules: clustering or structuring

Fuzzy sets: learning techniques derived from NN

➡ **Neuro-Fuzzy Systems**

Neuro-Fuzzy Classification: NEFCLASS



Fuzzy system drawn as a special kind of neural network

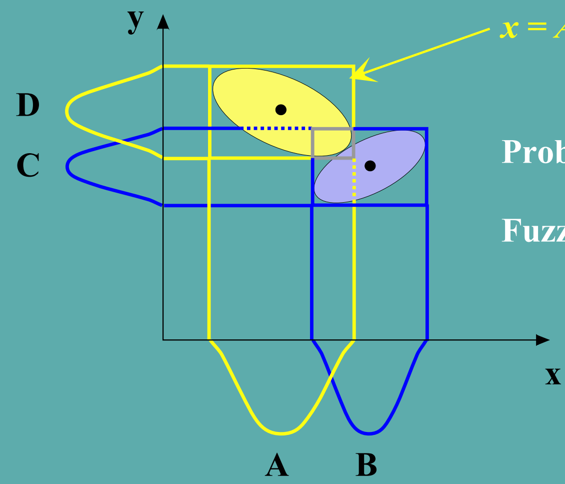
Fuzzy rules and fuzzy sets are obtained by learning, e.g.:

R_2: if x_1 is *small* and x_2 is *large* then class c_1

Neuro-Fuzzy: learn fuzzy systems from data by NN-like heuristics, usually no real NN involved

Classification: Creating Fuzzy Rules

Creating fuzzy rules by projection of fuzzy clusters

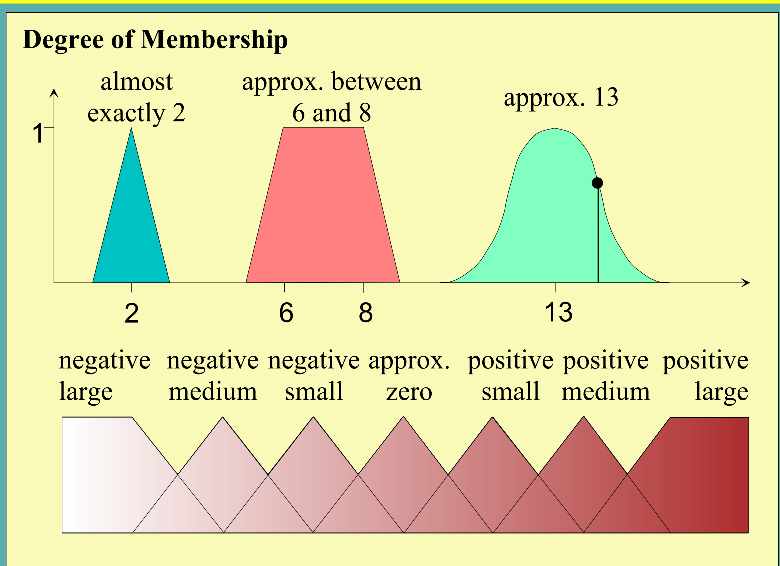


$x = A$ and $y = D$

Problem:

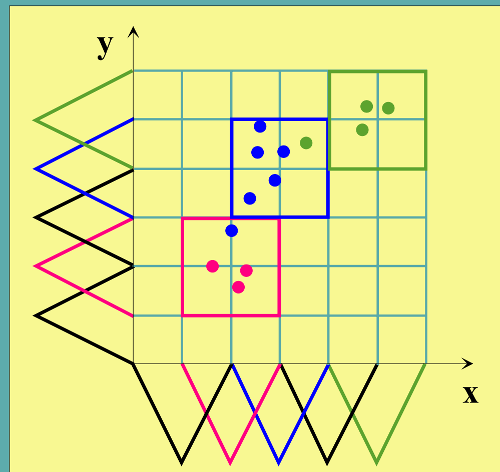
Fuzzy Rule $\neq$ Cluster

Fuzzy Systems: Typical Fuzzy Sets



Classification: Creating Fuzzy Rules

Creating fuzzy rules by structuring the data space



Classification: Medical Example

Data: Results obtained from testing for breast cancer.

Cases: 699 (16 cases have missing values)

Features: 9 discrete attributes per pattern, ranges: 1 - 10

Classes: 2 (benign: 458, malignant: 241)

Origin: University of Wisconsin Hospitals, Madison (W.H. Wolberg), available by FTP from (ics.uci.edu, Wisconsin Breast Cancer (WBC) data)

WBC Example: NEFCLASS Results

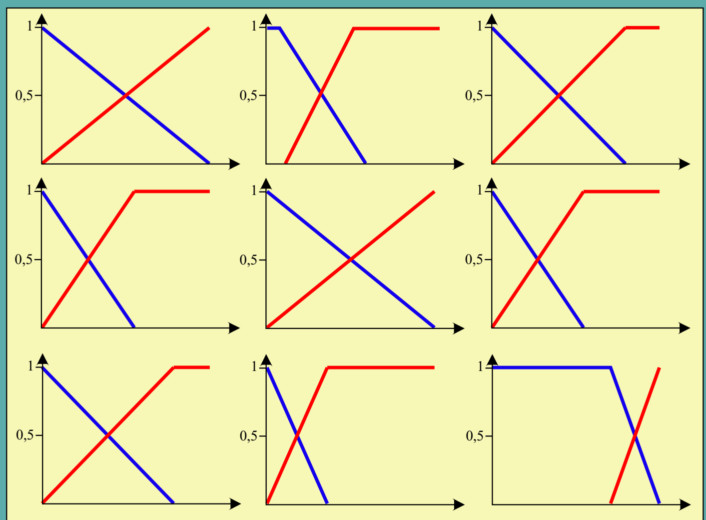
Using two fuzzy sets (*small* and *large*) for each variable results in $2^9 = 512$ possible rules.
NEFCLASS detects 83 rules and uses the best 4 rules.
The result is then further improved by pruning

After pruning: 2 fuzzy rules with 6 or 5 variables using 2 fuzzy sets per variable

if	uniformity of cell size is *large* and uniformity of cell shape is *large* and marginal adhesion *large* and bare nuclei is *large* and bland chromatin is *large* and normal nucleoli is *large*
then	class is *malignant*

if	uniformity of cell shape is *small* and marginal adhesion is *small* and bare nuclei is *small* and bland chromatin is *small* and normal nucleoli is *small*
then	class is *benign*

Fuzzy Sets for WBC Data



WBC Example: Other Results

Model	Tool	Remarks	Error	Validation
Discriminant Analysis	SPSS	linear model, 9 variables	3.95%	1 leave out
MLP	SNNS	9-4-2 MLP, RProp	5.18%	50% test set
Decision Tree	C4.5	31 (24.4) nodes, pruned	4.9%	10-fold
Rules from Decision Tree	C4.5rules	8 (7.5) rules, 1-3 variables	4.6%	10-fold
NEFCLASS	NEFCLASS-X	2 (2.1) rules, 5-6 variables	4.94%	10-fold

Neuro-Fuzzy Systems

- Tools to create fuzzy systems from data.
- Not fuzzy logic in the narrow sense.
- Neuro fuzzy systems perform function approximation.
- The learning algorithms must be constrained to not destroy the semantics of the underlying fuzzy sytem.
- Neuro fuzzy systems are used, if a fuzzy systems is sought as a solution and/or if prior knowledge is available.

Dependency Analysis

- Association Rules: If A then B in *x*% of all cases
- Bayesian Networks: Dependencies are modeled by conditional probabilities
- Possibilistic Networks: Dependencies can be modeled by fuzzy rule bases (different inference mechanism then in fuzzy systems!)

Association Rules

Bar-code technology makes it possible to store huge amounts of sales data.

Find rules in *basket data* for
- cross-marketing
- mailings
- catalog design
- store layout
- customer segmentation

Association Rules

Data: Set of transactions with several items each

1. Find large item sets *L* that occur in more than *s*% of all transactions
2. For every large item set *L* find all its subsets *A*
3. Create rule $A \rightarrow (L\text{-}A)$ if more than *c*% of transactions containing *A* contain also *L*
4. Analyze rules and keep only the "interesting" ones

Association Rules

The rules are probalistic in nature, not logical:
$X \rightarrow A$ does not necessarily mean $X + Y \rightarrow A$ holds.
$X \rightarrow Y$ and $Y \rightarrow Z$ does not necessarily mean $X \rightarrow Z$ holds.

It is not feasible to enumerate and test all possible rules.

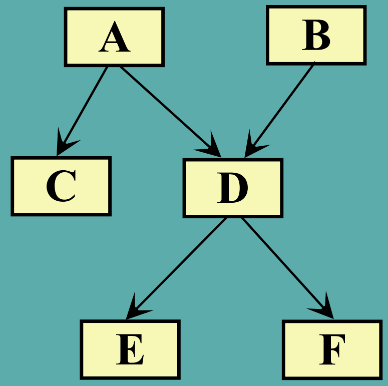
Therefore the user provides minimum values for
s (support) and *c* (confidence)

The main costs are in finding large item sets

Fast algorithms are availiable (Agrawal, Srikant 1994)

Bayesian Networks

Dependency Graph
(directed acyclic graph)



For A and B specify prior probabilities, e.g. $p(A=a_1) = 0.3$ etc.

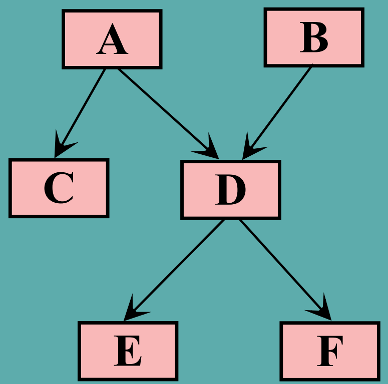
For C-F specify conditional probabilities, e.g. $p(D=d_1 \mid A=a_1, B=b_1) = 0.2$ etc.

Algorithm operates only on subspaces, e.g. {A, B, D}.

Reasoning in all directions

Possibilistic Networks

Dependency Graph
(directed acyclic graph)



For A and B specify possibility distributions (by fuzzy sets)

For C-F specify conditional possibility distributions (in form of fuzzy rules)

Algorithm operates only on subspaces, e.g. {A, B, D}.

Reasoning in all directions

Bayesian vs. Possibilistic Networks

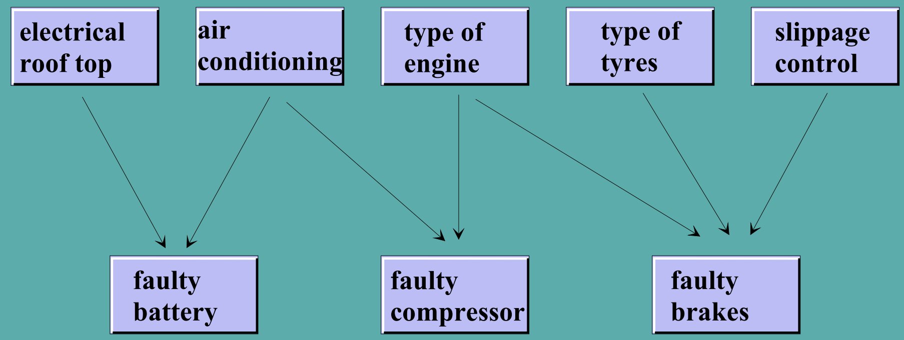
Bayesian Network	Possibilistic Network
precise data	precise, imprecise and vague data
uncertainty modeled by probability distribution	uncertainty modeled by possibility distribution
Result: which combination of attributes has the highest belief	Result: which combination of attributes has the highest possibility

Both type of networks can be learned from data

Example: Mercedes Benz Database

- Two large data sets:
  - passenger cars (18,500 cases)
  - trucks (13,000 cases)
- More than 100 attributes
- Learning a Bayesian network from data
- Less than 30 min runtime on a SUN Sparcstation 20
- Interesting dependencies found between special equipment and faults

Bayesian Network (Application at Daimler)



Fictituous example: There are significantly more faulty batteries, if both air conditioning and electrical roof top are built into the car.

Segmentation

Goal: Detect groups of cases that are similar and belong together

Problem: We don't know how many groups there are and how they should look like

Approach: Cluster Analysis

Segmentation: Cluster Analysis

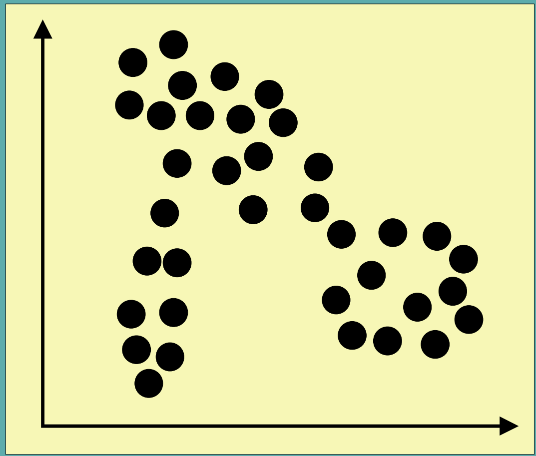
Numerical Attributes

- similarity is defined by a distance measure
- clusters are high-dimensional spheres or ellipsoids (hyper-ellipsoids)

Symbolic Attributes

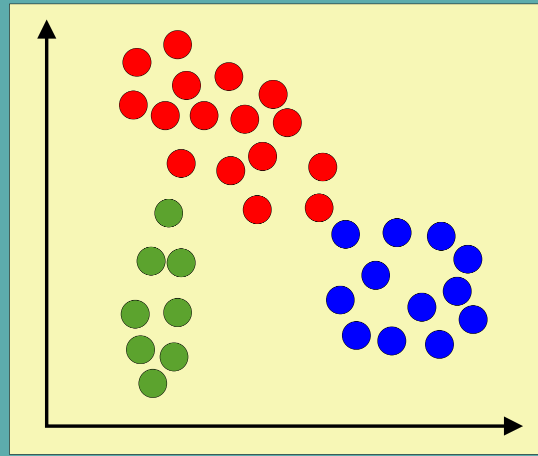
- there is no distance between symbols
- conceptual clustering - try to find separating descriptions

Segmentation: Cluster Analysis



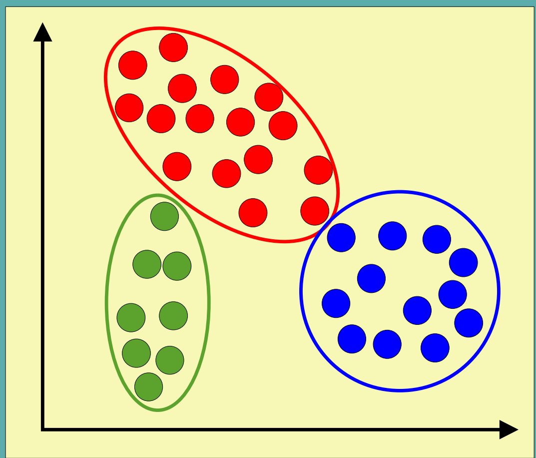
How many clusters are there?

Segmentation: Cluster Analysis



What shape are the clusters?

Segmentation: Cluster Analysis



How to describe the clusters?

Statistical Cluster Analysis

Hierarchical Cluster Analysis

- begin with one cluster for each case, and iteratively combine the two most similar clusters

- proceed until all cases belong to one cluster

- the merging stages can be displayed in a diagram

Select a stage (number of clusters) based on goodness

Each case belongs to exactly one cluster
Not applicable, if there is a large number of cases

Statistical Cluster Analysis

Non-Hierarchical Cluster Analysis (k-means, c-means)

- clusters are defined by a prototype and a distance measure
- specify number of clusters *k*, use *k* random prototypes (cases)
- iteratively update prototypes until they do not change anymore, or maximum number of iterations is over

Goal: find prototypes with large distance between them

Each case belongs to exactly one cluster
The number of clusters must be known (guessed)

Statistical Cluster Analysis

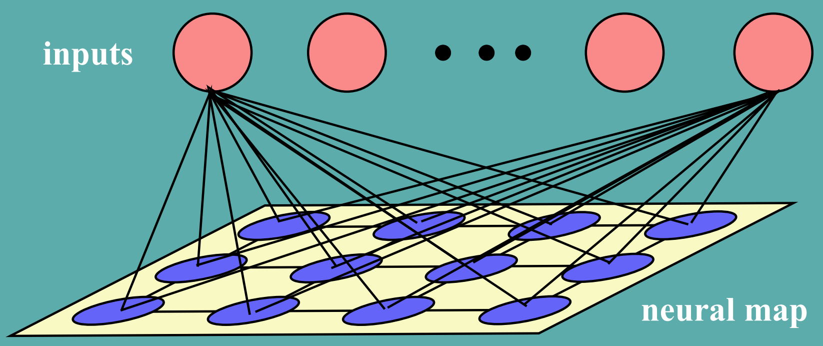
Distance measure: $\sqrt{(\vec{x}_0 - \vec{x})^T \Sigma^{-1} (\vec{x}_0 - \vec{x})}$

x_0 is the prototype and Σ^{-1} is an inverse covariance matrix (i.e. symmetrical and only positive Eigenvalues)

If Σ^{-1} is
- the identiy matrix: sphere (Euklidean distance)
- a diagonal matrix: axes-paralles hyper-ellipsoid
- otherwise: arbitrarily rotated hyper-ellipsoid

Segmentation: Self-Organizing Maps (NN)

Training by adaptive vector quantization
Similar patterns activate adjacent neurons in the map



Kohonen's self-organizing feature map

Clustering with Kohonen Feature Maps

Weight vectors of the map neurons are prototypes

Competitive learning, weights are slowly "frozen"

Topology preserving mapping from high-dimensional data space onto 2-dimensional map

Parts of the data space with many data are represented by more neurons in the map than parts with few data

Can be used for visualization of high-dimensional data

Segmentation: Fuzzy Cluster Analysis

Each case can belong to serveral clusters with different degree of membership

Overlapping of clustes is tolerated

Each cluster is a high-dimensional fuzzy set (fuzzy relation)

Membership degrees of each case must sum up to 1
➡ probabilistic interpretation

If this restriction is not applied
➡ possibilistic clustering (more robust against outliers)

Segmentation: Fuzzy Cluster Analysis

Minimize $J(X, U, v) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m d^2(v_i, x_k)$

with $\sum_{i=1}^{c} u_{ik} = 1$ and $\sum_{k=1}^{n} u_{ik} > 0$

X : data set, m : fuzzifier (usually $1 < m < 2$)

u_{ik} : degree of membership of x_k to cluster i

v_i : prototype of cluster i, d : distance measure

Segmentation: Fuzzy Cluster Analysis

Most often used algorithm: fuzzy c-means (FCM)
- searches for hyper-spheres of similar size
- fuzzification of c-means clustering

Advanced approaches:
- Gustafson & Kessel: hyper-ellipsoids of same size
- Gath & Geva: hyper-ellpsoids of arbitrary size

Rule creation by projection of clusters:
- search for axis-parallel ellipsoids only
- search for rectangular clusters (hyper-boxes)

Problem: Number of clusters must be given!

Segmentation: Fuzzy Cluster Analysis

The quality of the clustering result can be estimated by goodness measures.

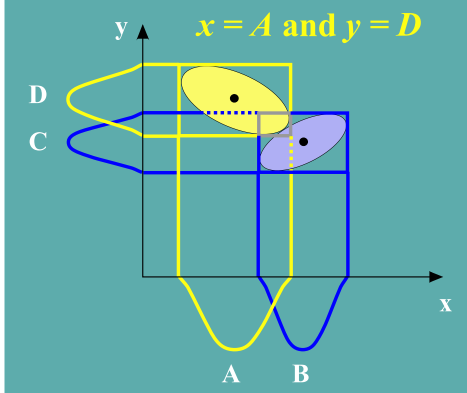
Idea: patterns should have high membership degrees (msd) with "their" cluster and low msd with other clusters.

Determine number of cluster automatically:

- Compute cluster analyses for 2 , 3, 4, ... clusters.

- Continue as long the goodness measure improves.

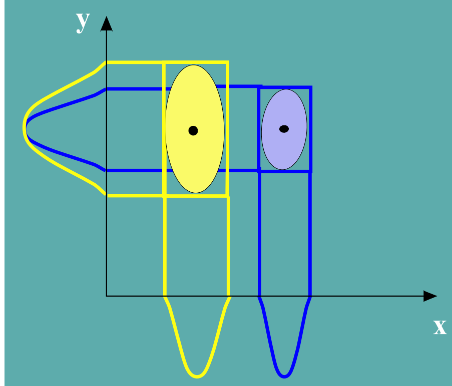
Fuzzy Clustering: Creating Fuzzy Rules



Problem: There is a loss of informaton, clusters and rules are not identical.

The resulting fuzzy sets must be labeled with suitable linguistic terms.

Fuzzy Clustering: Creating Fuzzy Rules



Fuzzy sets created by projection may be hard to interpret.

Unusual distributions of fuzzy sets can be obtained.

Axis-parallel ellipsoids reduce the loss of information due to projection

Preprocessing

Reduction of Dimensionality (Number of Variables)

Select influential variables

- approaches: statistical tests, e.g. correlations

Combine variables to create new influential variables

- approaches: main component analysis, factor analysis

Preprocessing

Reduction of Size (Number of Cases)

Remove outliers

approach: do "sanity checks" on the attribute values

Select a subset from all cases

approach: select randomly, but watch distribution

Preprocessing

Data Cleansing (Reduce Size, Improve Data)

Missing Values
- delete cases with missing values
- estimate missing values by statistical methods
- do nothing, if your data mining method can handle them

Remove Noise
- filter the data to remove high frequency noise (mainly for function approximation and time series prediction)

Preprocessing

Know Your Data Like Yourself

Compute basic descriptive statistics (mean, variance, ...).

Try simple linear models to see how they perform.

Visualize the data
- plot bar charts, 2D and 3D projections, ...

Ask "experts", i.e. persons who work with the data and collected it.

Validation

Always validate the model that is created during data mining!

N-fold Cross Validation:

Divide the data in *n* equal parts (same size and distribution).

Use *n*-1 parts to create a model and test on the remaining part.

Repeat *n* times, and compute the mean error.

Create a final model from the whole data set.

The mean error is an estimate for the error on unseen data.

Postprocessing

Interpret the result
Is is usable, efficient, easy to understand and to maintain?

Report all steps of the data mining process
It is essential that the result can be reproduced

Visualize the result
It is important that other persons can understand the result

Update the result, if your data changes
Specify when the result may be out of date

Evaluation Criteria

- Scalability
- Integretation with data warehouse
- Completeness
  - Is it an algorithm or a solution (application)?
- Usability
  - Does it solve a marketing problem?
  - Who is going to use it?
  - How is it going to be used?
  - How much does it cost?

Explaining the Results

- Depending on the selected model the results can be quite complex
- The results may influence strategical decisions
- Words are often better than numbers
- Interaction with users:
  - users must "get a feeling" for the result
  - let users identify their customers
  - reveal the data on several levels of detail, from a broad overview to the fine structure

Legal and Ethical Questions

- Privacy Concerns
  - Becoming more important
  - Will impact the way data can be used and analyzed
  - Ownership issues
- It may not be legal to use or combine data that is legally stored in different databases
- Think as a customer: Do you feel alright about the way data about you is gathered and analyzed?

Back to Scenario A - Solutions 1 and 2

Prevention of cancelation and cross-selling (classification)

- Assumption: we cannot handle all 1 mio cases

- Select a subset from the data base for training

- Preprocess, deal with missing values (estimation, deletion)

- Begin with statistical analysis to learn more about the data

- Select classifier(s) (black box or easy to understand)

- Validate the solution(s), select one

Back to Scenario A - Solution 3

Cross-selling without historic data (clustering)

- Begin like in solutions 1 and 2

- Select a cluster analysis appraoch (e.g. fuzzy clustering)

- Create rules to describe the cluster

- Try to identify and label groups described by rules

- Direct validation is not possible (no targets are given), but is the cluster goodness similar on unseen data?

Back to Solution of Scenario B

Detect the end of the process

Expert selects "typical" processes and marks process end *c*.

Filter the data to reduce noise.

Detect *c* using the process history as input.

The company tried NN at first, but failed due to lack of expertise in handling NN.

It turned out, that a simple linear filter and observing the deviation from a regression line was sufficient.

Philosophies of Tools

Ground Level:
- add more sophisticated approaches to existing tools
- very flexible, but require a lot of expertise

One Step Up:
- data mining toolboxes
- problematic: often aim at users with insufficient expertise to consider tradeoffs

High Level Tools:
- end user applications, integrated into data warehouse
- interactive graphical tools: aimed at non-experts
- ease of use more important then accuracy

Some Tools

Statistics
- SAS (also data warehousing, statistics, NN, decision trees)
- SPSS (standalone statistics, add-ons for NN, CHAID)

Neural Networks
- SNNS (Stuttgart Neural Network Simulator, free)
- ECANSE, SENN (Siemens)

Data Mining
- Clementine (decision trees, NN)
- Data Engine (MIT GmbH, fuzzy, NN, plug-in extensions)
- IBM Data Mining Tool (statistics, NN, decision trees)
- Kepler (multi-relational data, logical rules, dec. trees)

Resources

Book:
Fayyad U.M. et al.:
Advances in Knowledge Discovery and Data Mining
MIT Press, Cambridge, MA 1996

WWW:
Knowledge Discovery Nuggets (with links to software)
http://www.kdnuggets.com

Journal on Data Mining and Knowledge Discovery
http://www.research.microsoft.com/research/datamine/

Conclusions

- There is not a single best method for data mining
- There are many methods, some are interchangeable.
- Thoroughly preprocess your data (get to know them).
- Know your objectives: interpretability or accuracy?
- At first, try methods and tools your are familiar with.
- Thoroughly validate and evaluate your results.