Data Mining and Knowledge Discovery

Part of
"New Media and e-Science" M.Sc. Programme
and "Statistics" M.Sc. Programme

2007 / 2008

Nada Lavrač

Jožef Stefan Institute

Ljubljana, Slovenia

Thanks to Blaž Zupan, Sašo Džeroski and Peter Flach for contributing some slides to this course material

Course participants

I. IPS students

- Andraž Brečko
- Blaž Fortuna
- Darko Čerepnalkoski
- Igor Popov
- Jan Rupnik
- Martin Mihajlov
- Matjaž Juršič
- Miha Grčar
- Tomaž Klančnik
- Uroš Platiše
- Uroš Legat
- Vid Podpečan

II. Statistics students

- Jure Ačimovič
- Klemen Kolenc
- Mojca Omerzu
- Mojca Rožman
- Barbara Strnad
- Boris Šuselj
- Šmuc Alenka
- Melita Ulbl
- Tomaž Weiss
- Tina Zupanič

III. Other participants

- Inna Kovalna
- Nejc Trdin
- Anže Vavpetič
- Blažo Đurovič

Course Schedule - 2007/08
Data Mining and Knowledge Discovery (DM)
Knowledge Management (KM)

- DM - Wednesday, 17 Oct. 07, 15-19 - Lavrač, lectures, MPS
- KM - Wednesday, 24 Oct. 07, 15-19 - Lavrač, lectures, MPS
- DM - Thursday, 8 Nov. 07, 15-19 - Kralj et al., practice, E8
- DM - Thursday, 15 Nov. 07, 15-19 - Kralj et al., practice, E8
- KM - Thursday, 22 Nov. 07, 15-19 - Fortuna, practice, E8
- DM - Thursday, 29 Nov. 07, 15-19 - written exam
  & seminar topic presentations, E8
- DM - Wednesday, 13 Feb. 08, 15-19 - seminar results
  presentations, MPS
- KM - Wednesday, 27 Feb. 07, 15-19 - written exam
  & seminar results presentations, MPS

DM - Credits and coursework

“New Media and eScience” MSc Programme

- 12 credits (30 hours)
- Lectures
- Practice
  - Theory exercises and hands-on (WEKA)
- Seminar – choice:
  - Majority: Programming assignment - write your own data mining module, and evaluate it on a (few) domain(s), or
  - Minority: Data analysis results on your own data (e.g., using WEKA for questionnaire data analysis)
- Contacts:
  - Nada Lavrač nada.lavrac@ijs.si
  - Petra Kralj petra.kralj@ijs.si

“Statistics” MSc Programme

- 12 credits (36 hours)
- Lectures
- Practice
  - Theory exercises and hands-on (WEKA)
- Seminar – choice:
  - Majority: Data analysis results on your own data (e.g., using WEKA for questionnaire data analysis), or
  - Minority: Programming assignment - write your own data mining module, and evaluate it on a (few) domain(s
- Contacts:
  - same as for MPS students

DM - Credits and coursework

Exam

Thursday, 29.11.07

- Written exam - Theory – 1 hour
- Oral presentations of your seminar topic (DM task or dataset presentation, max. 4 slides) – 3 hours

Wednesday, 12.2.07 seminar results presentation – 4 hours

- Presentation of your seminar results (max. 8 slides)
- Deliver written report + electronic copy (4 pages, double column, possibly with appendices, in Information Society paper format, see instructions on Petra's Web pages),
  - Report on data analysis of own data needs to follow the CRISP-DM methodology
  - Report on DM SW development needs to include SW uploaded on a Web page – format to be announced

Course Outline

I. Introduction

- Data Mining and KDD process
- DM standards, related research areas and tools

  (Ch. 1 and 11 of Mladenić at al. book, Introduction to Kononenko et al. book)

II. Predictive DM Techniques

- Bayesian classifier

  (Ch. in Kononenko's book)
- Decision Tree learning (Ch. 3 of Mitchell's book)
- Classification rule learning (Ch. 7 of IDA book, Ch. 10 of Mitchell's book)
- Classifier Evaluation

  (Ch. 7 in Bramer's book)

III. Descriptive DM

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

IV. Conclusions and literature

Part I. Introduction

- Data Mining and the KDD process
- DM standards, related research areas and tools

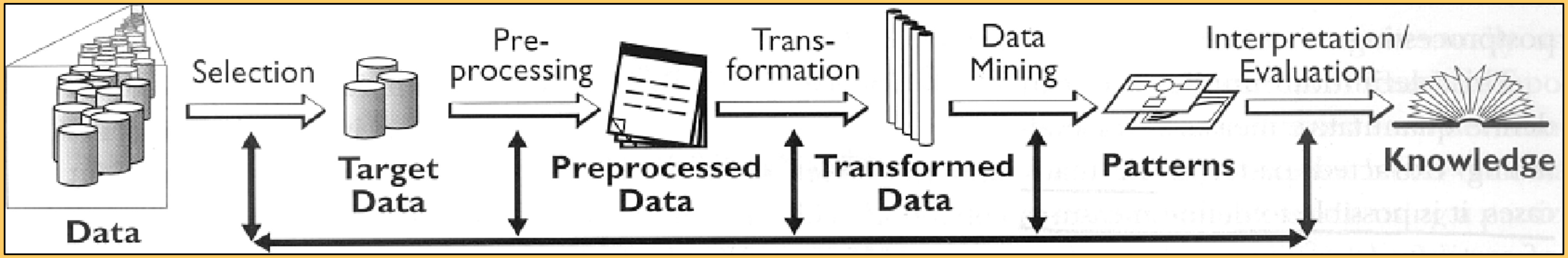
Data Mining and KDD

- 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.

Usama M. Fayyad, Gregory Piatesky-Shapiro, Pedhraic Smyth: The KDD Process for Extracting Useful Knowledge form Volumes of Data. Comm ACM, Nov 96/Vol 39 No 11

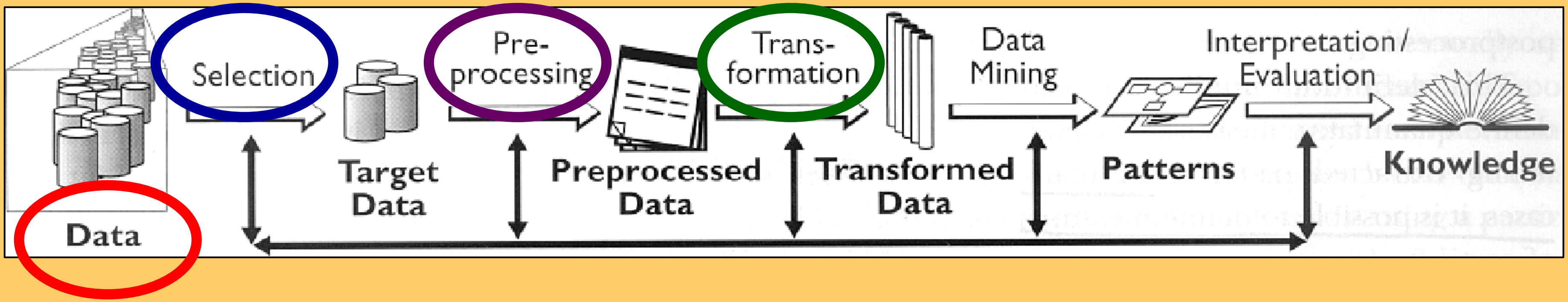
KDD Process

KDD process of discovering useful knowledge from data



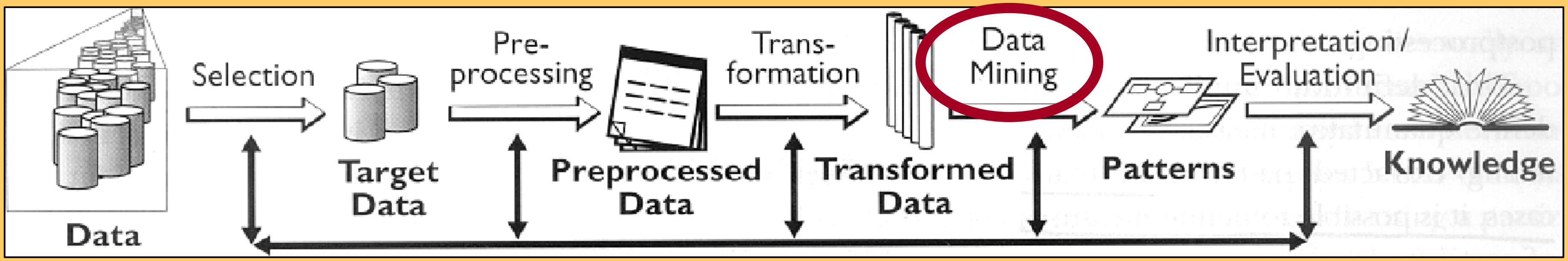
- KDD process involves several phases:
  - data preparation
  - data mining (machine learning, statistics)
  - evaluation and use of discovered patterns
- Data mining is the key step, but represents only 15%-25% of the entire KDD process

MEDIANA – analysis of media research data



- Questionnaires about journal/magazine reading, watching of TV programs and listening of radio programs, since 1992, about 1200 questions. Yearly publication: frequency of reading/listening/watching, distribution w.r.t. Sex, Age, Education, Buying power,..
- Data for 1998, about 8000 questionnaires, covering lifestyle, spare time activities, personal viewpoints, reading/listening/watching of media (yes/no/how much), interest for specific topics in media, social status
- good quality, “clean” data
- table of n-tuples (rows: individuals, columns: attributes, in classification tasks selected class)

MEDIANA – media research pilot study



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

Association rules

Rules X => Y, X, Y conjunction of bin. attributes

Task: Find all association rules that satisfy minimum support and minimum confidence constraints

- Support: Sup(X,Y) = #XY/#D = p(XY)
- Confidence: Conf(X,Y) = #XY/#X = p(XY)/p(X) = p(Y|X)

Example association rule about readers of yellow press daily newspaper SloN (Slovenian News):

read_Love_Stories_Magazine => read_SloN

sup = 3.5% (3.5% of the whole dataset population reads both LSM and SloN)

conf = 61% (61% of those reading LSM also read SloN)

Association rules

Finding profiles of readers of the Delo daily newspaper

1. read_Marketing_magazine 116 =>
   read_Delo 95 (0.82)
2. read_Financial_News (Finance) 223 => read_Delo 180 (0.81)
3. read_Views (Razgledi) 201 => read_Delo 157 (0.78)
4. read_Money (Denar) 197 => read_Delo 150 (0.76)
5. read_Vip 181 => read_Delo 134 (0.74)

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

Association rules (in Slovene)

1. bere_Sara 332 => bere_Slovenske novice 211 (0.64)
2. bere_Ljubezenske zgodbe 283 =>
   bere_Slovenske novice 174 (0.61)
3. bere_Dolenjski list 520 =>
   bere_Slovenske novice 310 (0.6)
4. bere_Omama 154 => bere_Slovenske novice 90 (0.58)
5. bere_Delavska enotnost 177 =>
   bere_Slovenske novice 102 (0.58)

Večina bralcev Sare, Ljubezenskih zgodb, Dolenjskega lista, Omame in Delavske enotnosti bere tudi Slovenske novice.

Association rules (in Slovene)

1. bere_Sportske novosti 303 **=>**
   bere_Slovenski delnicar 164 (0.54)
2. bere_Sportske novosti 303 **=>**
   bere_Salomonov oglasnik 155 (0.51)
3. bere_Sportske novosti 303 **=>**
   bere_Lady 152 (0.5)

Več kot pol bralcev Sportskih novosti bere tudi Slovenskega delničarja, Salomonov oglasnik in Lady.

Classification rules

Set of Rules: if Cond then Class

Interpretation: **if-then** ruleset, or

if-then-else decision list

Class: Reading of daily newspaper EN (Evening News)

if a person does not read MM (Maribor Magazine) and rarely reads the weekly magazine “7Days”

then the person does not read EN (Evening News)

else if a person rarely reads MM and does not read the weekly magazine SN (Sunday News)

then the person reads EN

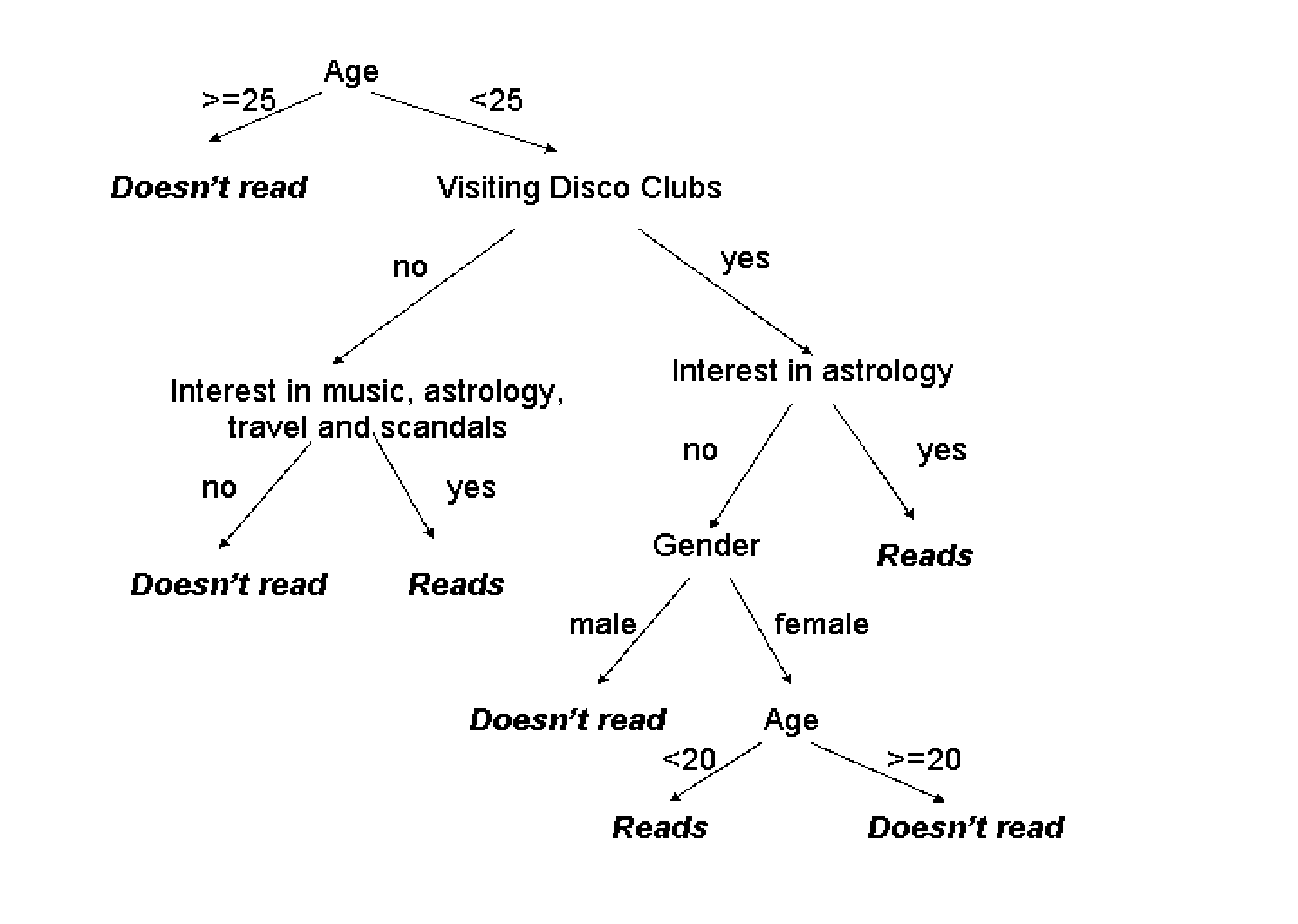
else if a person rarely reads MM

then the person does not read EN

else the person reads EN.

Decision trees

Finding reader profiles: decision tree for classifying people into readers and non-readers of a teenage magazine.



Part I. Introduction

Data Mining and the KDD process

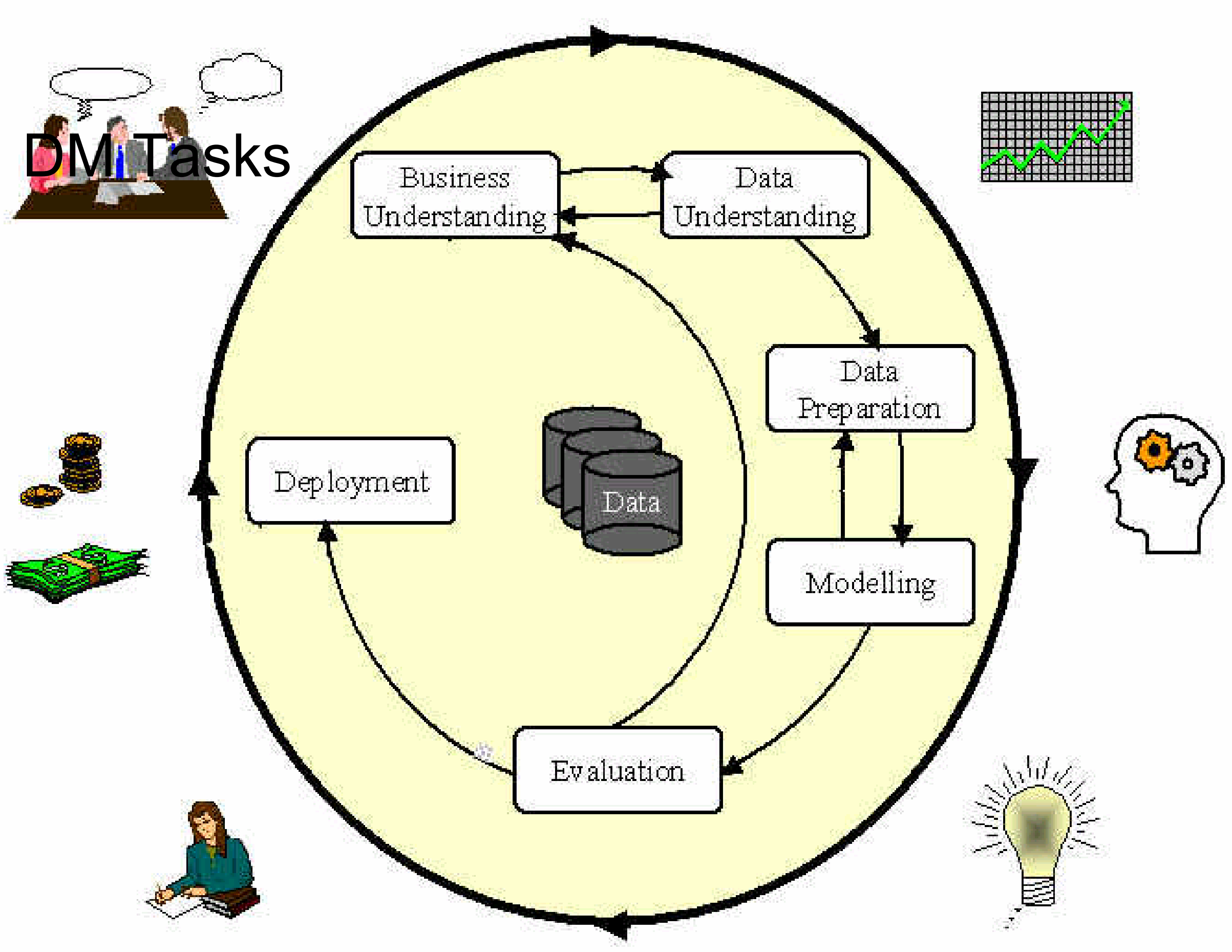
DM standards, related research areas and tools

CRISP-DM

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

CRISP Data Mining Process

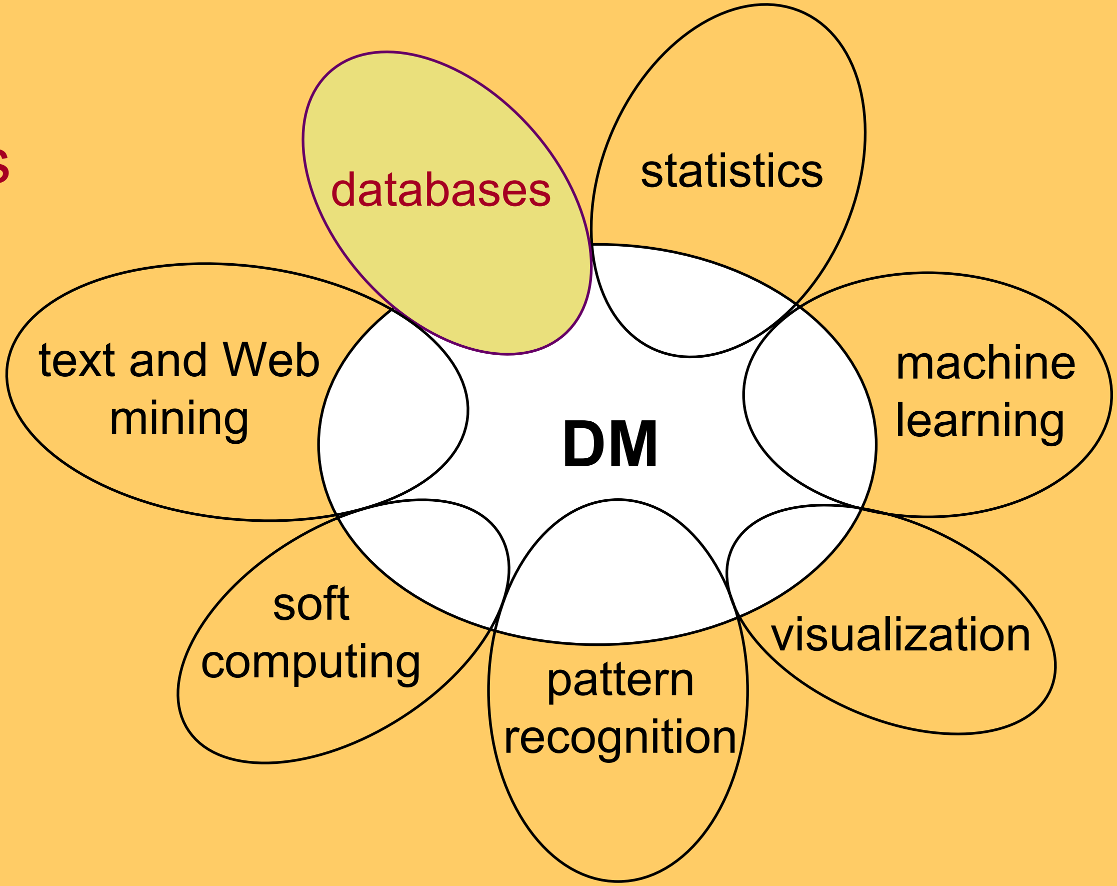
- DM Tasks



Related areas

Database technology
and data warehouses

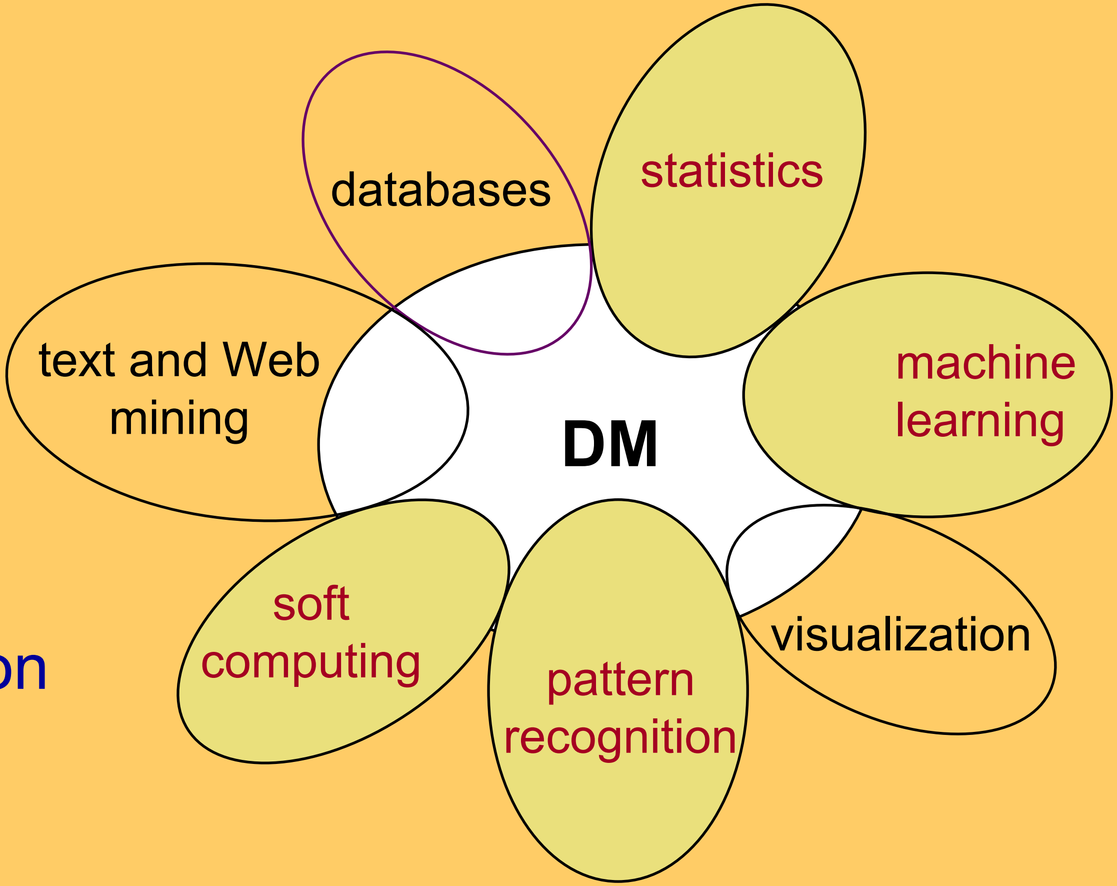
- efficient storage, access and manipulation of data



Related areas

Statistics,
machine learning,
pattern recognition
and soft computing*

- classification techniques and techniques for knowledge extraction from data

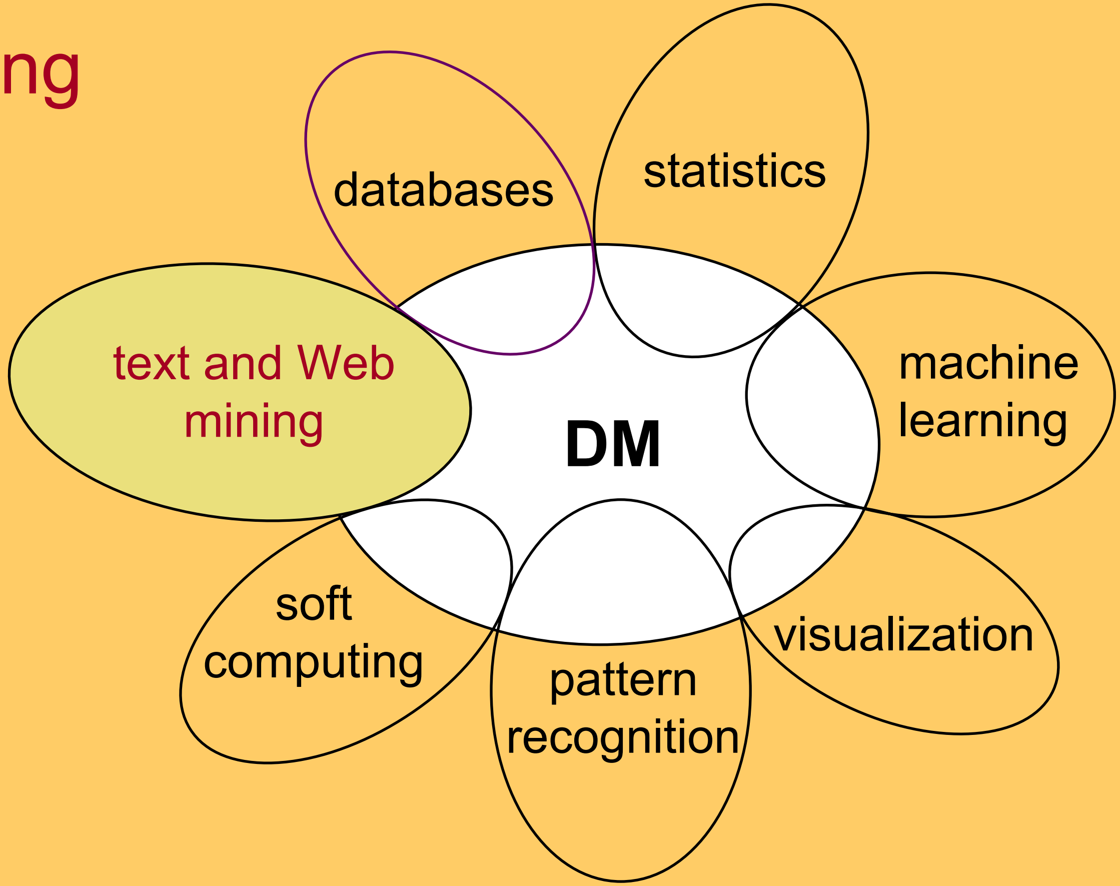


* neural networks, fuzzy logic, genetic algorithms, probabilistic reasoning

Related areas

Text and Web mining

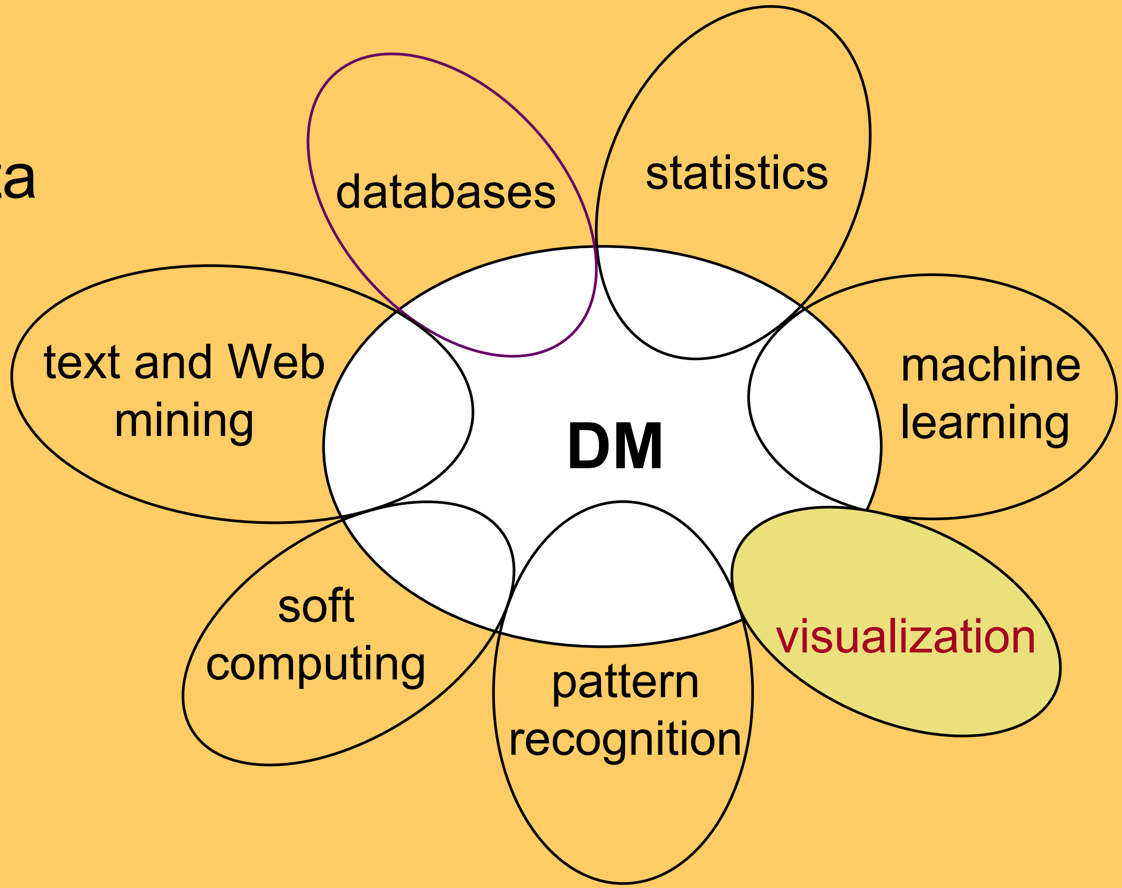
- Web page analysis
- text categorization
- acquisition, filtering and structuring of textual information
- natural language processing



Related areas

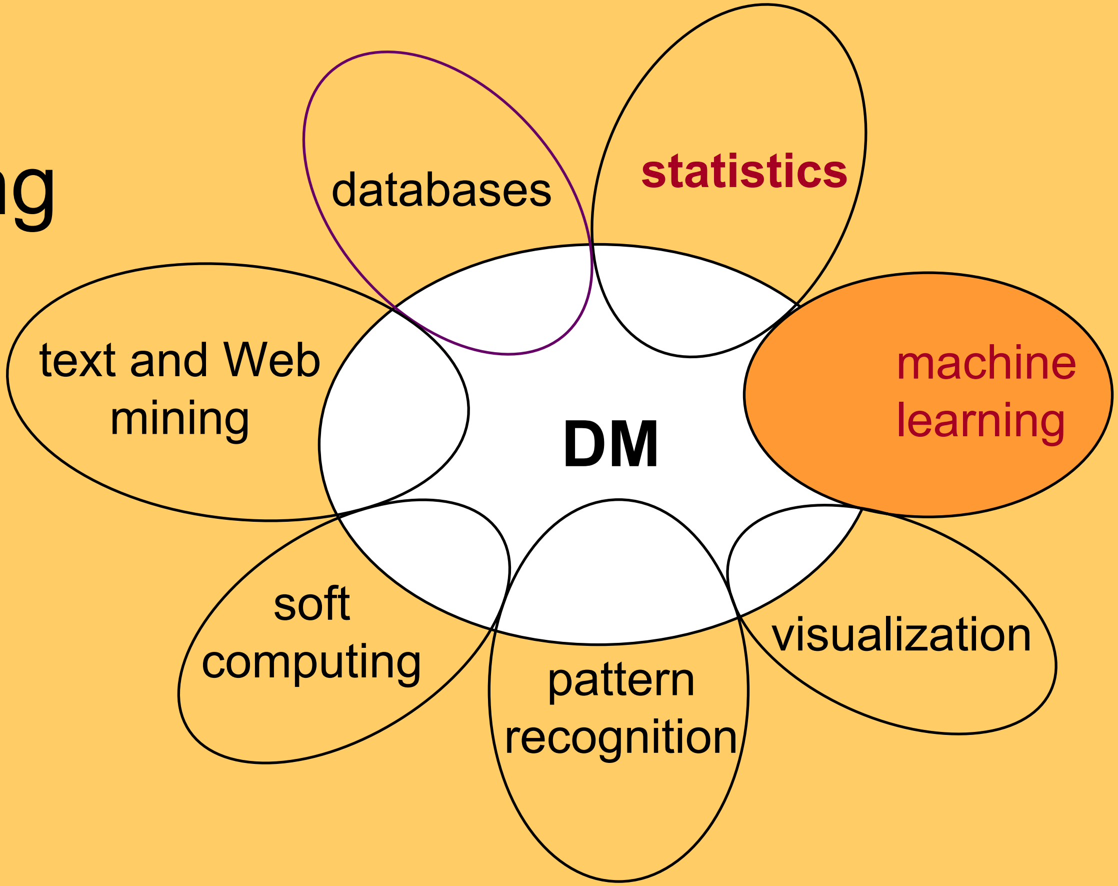
Visualization

- visualization of data and discovered knowledge



Point of view in this tutorial

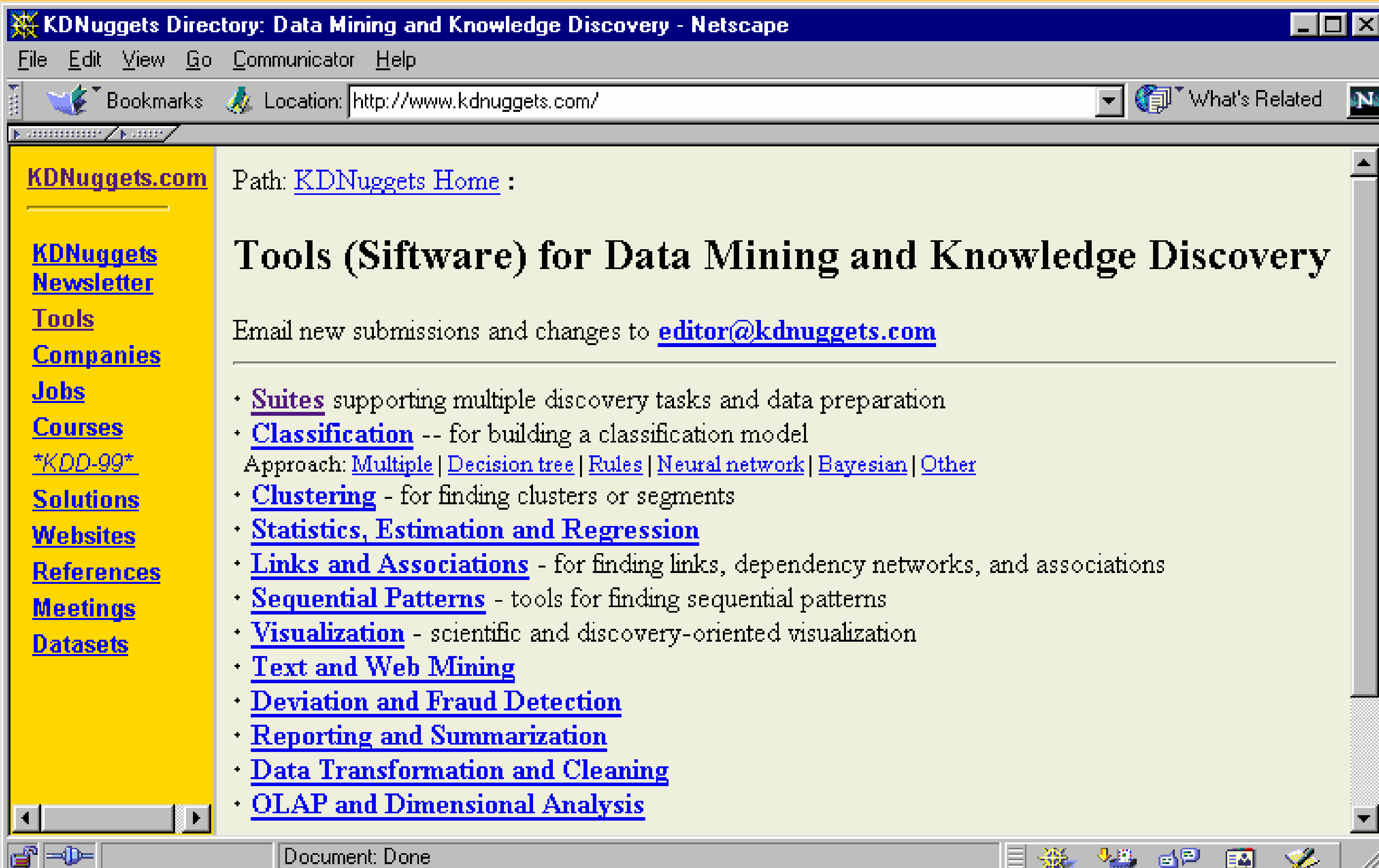
Knowledge discovery using machine learning methods



Data Mining, ML and Statistics

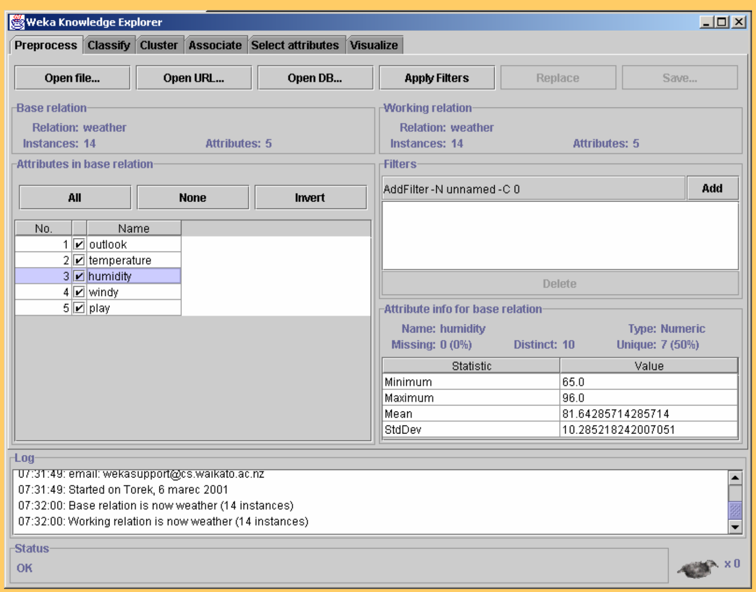
- All areas have a long tradition of developing <u>inductive techniques</u> for data analysis.
  - reasoning from properties of a data sample to properties of a population
- DM vs. ML - Viewpoint in this course:
  - Data Mining is the application of Machine Learning techniques to hard real-life problems
- DM vs. Statistics:
  - Statistics
    - Hypothesis testing when certain theoretical expectations about the data distribution, independence, random sampling, sample size, etc. are satisfied
    - Main approach: best fitting all the available data
  - Data mining
    - Automated construction of understandable patterns, and structured models
    - Main approach: heuristic search for decision trees, rules covering (parts of) the data space

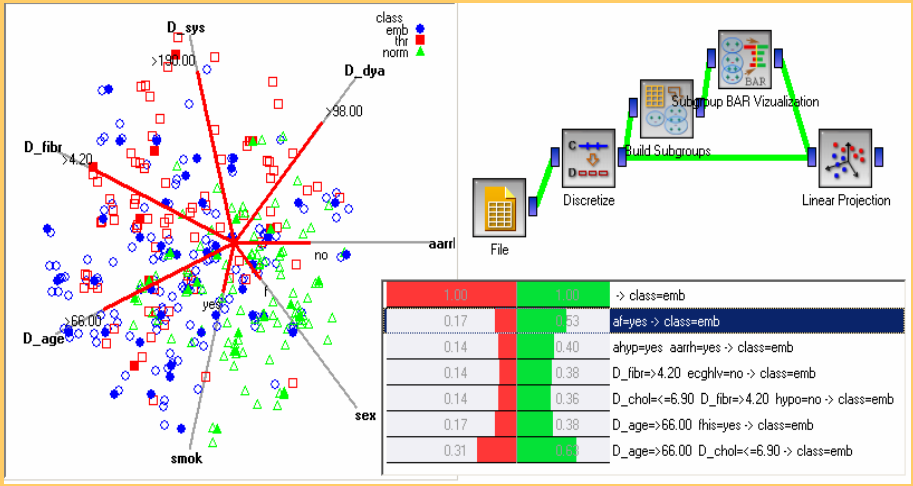
DM tools


KDNuggets Directory: Data Mining and Knowledge Discovery - Netscape
File Edit View Go Communicator Help
Bookmarks Location: http://www.kdnuggets.com/
What's Related
KDNuggets.com
KDNuggets Newsletter
Tools
Companies
Jobs
Courses
KDD-99
Solutions
Websites
References
Meetings
Datasets
Path: KDNuggets Home :
Tools (Siftware) for Data Mining and Knowledge Discovery
Email new submissions and changes to editor@kdnuggets.com
• Suites supporting multiple discovery tasks and data preparation
• Classification -- for building a classification model
Approach: Multiple | Decision tree | Rules | Neural network | Bayesian | Other
• Clustering - for finding clusters or segments
• Statistics, Estimation and Regression
• Links and Associations - for finding links, dependency networks, and associations
• Sequential Patterns - tools for finding sequential patterns
• Visualization - scientific and discovery-oriented visualization
• Text and Web Mining
• Deviation and Fraud Detection
• Reporting and Summarization
• Data Transformation and Cleaning
• OLAP and Dimensional Analysis
Document: Done

Public DM tools

- WEKA - Waikato Environment for Knowledge Analysis
- Orange
- KNIME - Konstanz Information Miner
- …





Part II. Predictive DM techniques

- Naive Bayesian classifier
- Decision tree learning
- Classification rule learning
- Classifier evaluation

Bayesian methods

- Bayesian methods – simple but powerful classification methods
  - Based on Bayesian formula

$$p(H \mid D) = \frac{p(D \mid H)}{p(D)} p(H)$$

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

* Out of scope of this course

Naïve Bayesian classifier

- Probability of class, for given attribute values

$$p(c_j \mid v_1 \ldots v_n) = p(c_j) \cdot \frac{p(v_1 \ldots v_n \mid c_j)}{p(v_1 \ldots v_n)}$$

- 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 $p(C_j \mid v_i)$)

$$p(c_j \mid v_1 \ldots v_n) \approx p(c_j) \cdot \prod_i \frac{p(c_j \mid v_i)}{p(c_j)}$$

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

$$C_{MAX} = \arg\max_{Cj} p(c_j \mid v_1 \ldots v_n)$$

Naïve Bayesian classifier

$$p(c_j \mid v_1 \ldots v_n) = \frac{p(c_j \cdot v_1 \ldots v_n)}{p(v_1 \ldots v_n)} = \frac{p(v_1 \ldots v_n \mid c_j) \cdot p(c_j)}{p(v_1 \ldots v_n)} =$$

$$= \frac{\prod_i p(v_i \mid c_j) \cdot p(c_i)}{p(v_1 \ldots v_n)} = \frac{p(c_j)}{p(v_1 \ldots v_n)} \prod_i \frac{p(c_j \mid v_i) \cdot p(v_i)}{p(c_j)} =$$

$$= p(c_j) \cdot \frac{\prod p(v_i)}{p(v_1 \ldots v_n)} \prod_i \frac{p(c_j \mid v_i)}{p(c_j)} \approx p(c_j) \cdot \prod_i \frac{p(c_j \mid v_i)}{p(c_j)}$$

Semi-naïve Bayesian classifier

- Naive Bayesian estimation of probabilities (reliable)

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

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

$$\frac{p(c_j \mid v_i, v_k)}{p(c_j)}$$

Probability estimation

- Relative frequency:

$$p(c_j) = \frac{n(c_j)}{N}, p(c_j \mid v_i) = \frac{n(c_j, v_i)}{n(v_i)}$$

j = 1. . k, for k classes

- Prior probability: Laplace law

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

- m-estimate:

$$p(c_j) = \frac{n(c_j) + m \cdot p(c_j)}{N + m}$$

Probability estimation: intuition

- Experiment with N trials, n successful
- Estimate probability of success of next trial
- **Relative frequency: n/N**
  - reliable estimate when number of trials is large
  - Unreliable when number of trials is small, e.g., 1/1=1
- **Laplace: (n+1)/(N+2), (n+1)/(N+k),** k classes
  - Assumes uniform distribution of classes
- **m-estimate: $(n+m.p_a)$ /(N+m)**
  - Prior probability of success p_a, parameter m (weight of prior probability, i.e., number of ‘virtual’ examples)

Explanation of Bayesian classifier

- Based on information theory
  - Expected number of bits needed to encode a message = optimal code length -log p for a message, whose probability is p (*)
- Explanation based of the sum of information gains of individual attribute values v_i (Kononenko and Bratko 1991, Kononenko 1993)

$$-\log(p(c_j \mid v_1 \ldots v_n)) =$$

$$= -\log(p(c_j)) - \sum_{i=1}^{n} (-\log p(c_j) + \log(p(c_j \mid v_i))$$

* log p denotes binary logarithm

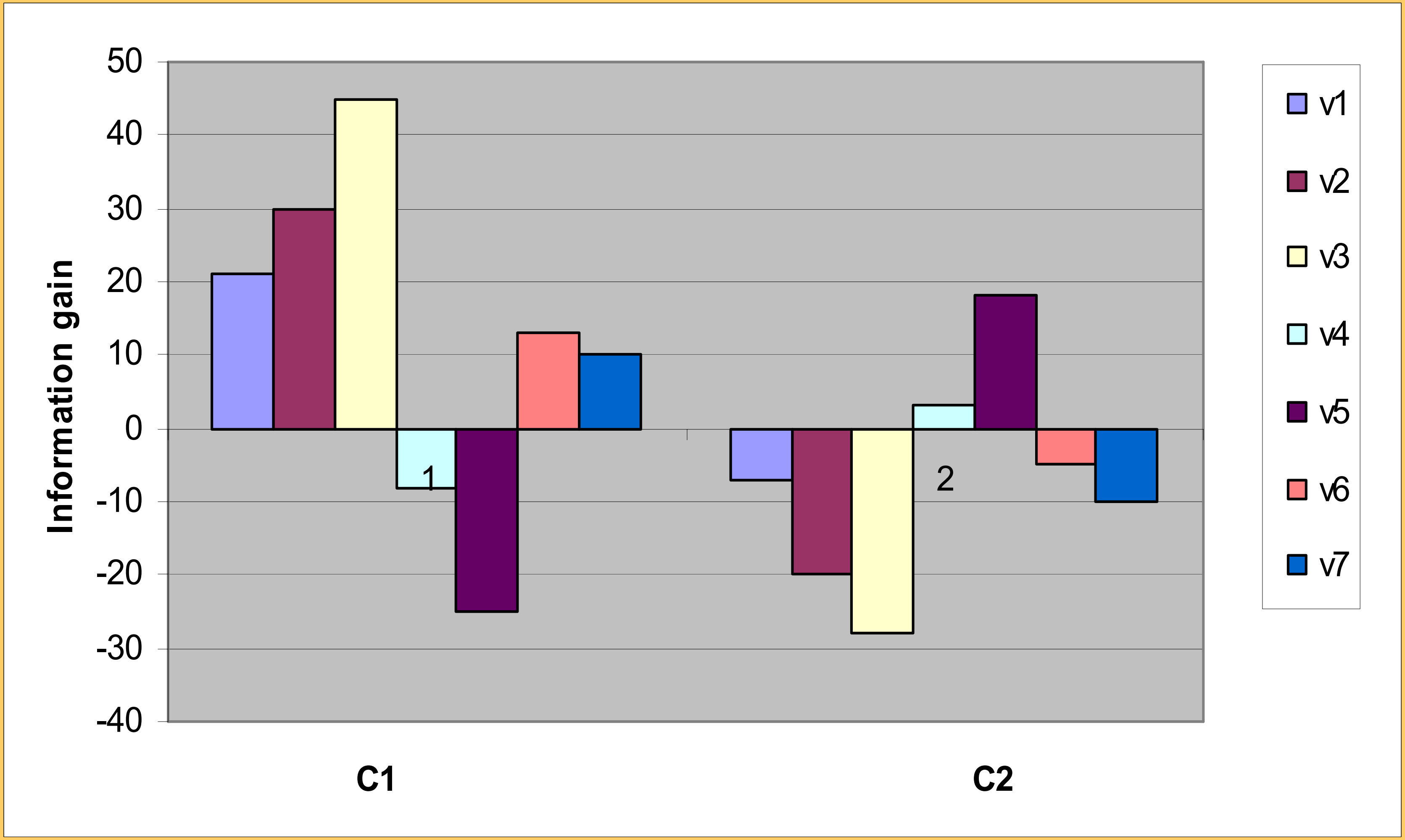
Example of explanation of semi-naïve Bayesian classifier

Hip surgery prognosis

Class = no (“no complications”, most probable class, 2 class problem)

Attribute value	For decision (bit)	Against (bit)
Age = 70-80	0.07	
Sex = Female		-0.19
Mobility before injury = Fully mobile	0.04	
State of health before injury = Other	0.52	
Mechanism of injury = Simple fall		-0.08
Additional injuries = None	0	
Time between injury and operation > 10 days	0.42	
Fracture classification acc. To Garden = Garden III		-0.3
Fracture classification acc. To Pauwels = Pauwels III		-0.14
Transfusion = Yes	0.07	
Antibiotic profilaxies = Yes		-0.32
Hospital rehabilitation = Yes	0.05	
General complications = None		0
Combination:	0.21	
Time between injury and examination < 6 hours		
AND Hospitalization time between 4 and 5 weeks		
Combination:	0.63	
Therapy = Artroplastic AND anticoagulant therapy = Yes		

Visualization of information gains for/against C_i



Naïve Bayesian classifier

- Naïve Bayesian classifier can be used
  - when we have sufficient number of training examples for reliable probability estimation
- It achieves good classification accuracy
  - can be used as ‘gold standard’ for comparison with other classifiers
- Resistant to noise (errors)
  - Reliable probability estimation
  - Uses all available information
- Successful in many application domains
  - Web page and document classification
  - Medical diagnosis and prognosis, …

Improved classification accuracy due to using m-estimate

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

	Relative freq.	m-estimate
Primary tumor	48.20%	52.50%
Breast cancer	77.40%	79.70%
hepatitis	58.40%	90.00%
lymphography	79.70%	87.70%

Part II. Predictive DM techniques

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

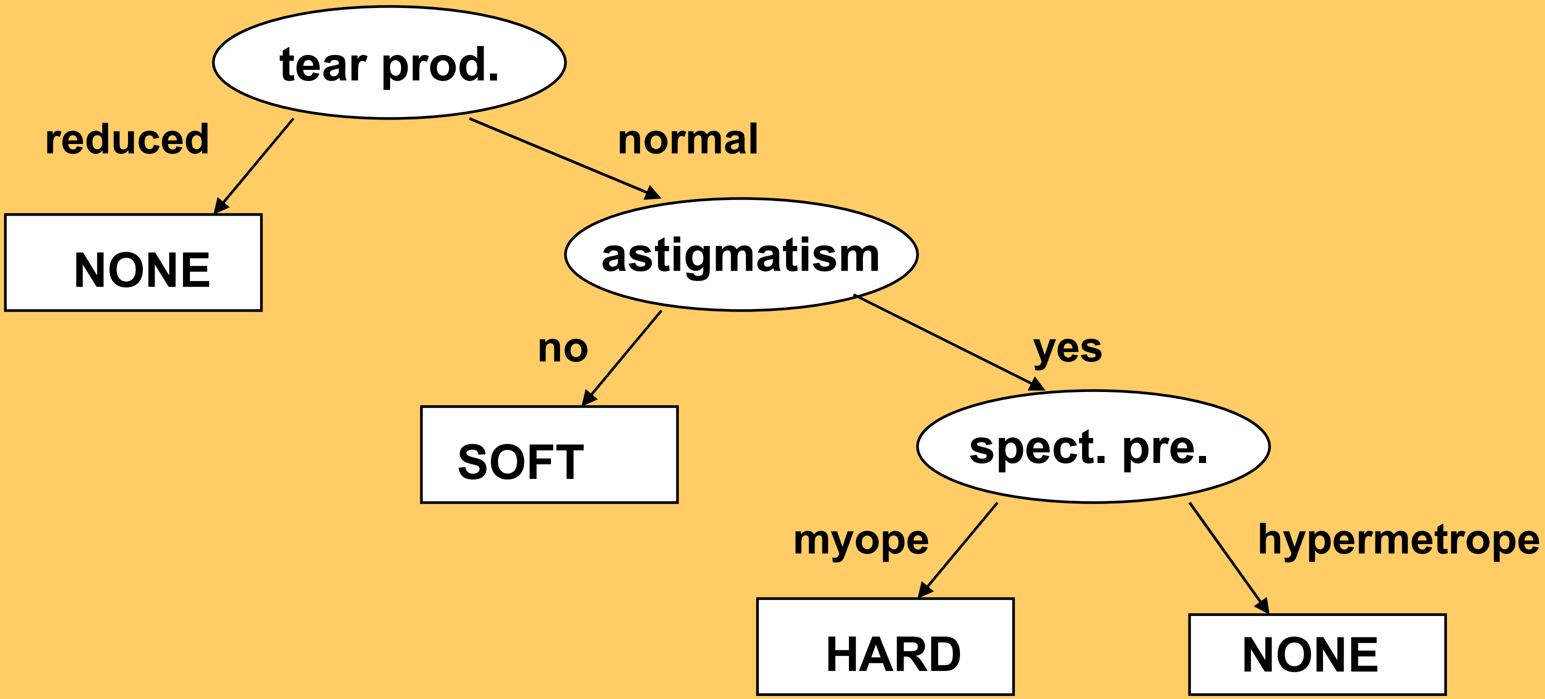
Predictive DM - Classification

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

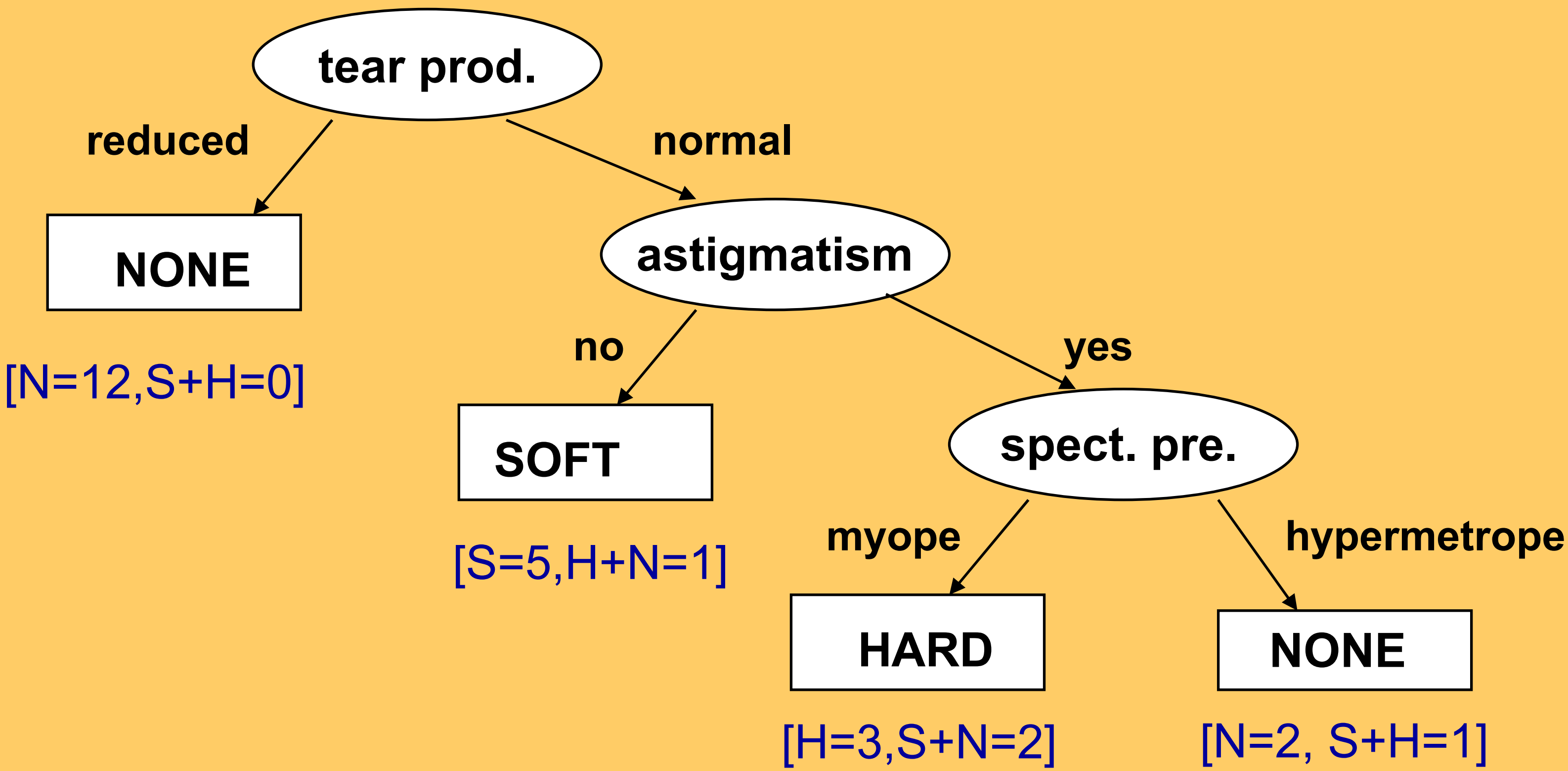
Illustrative example: Contact lenses data

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	young	myope	no	reduced	NONE
O2	young	myope	no	normal	SOFT
O3	young	myope	yes	reduced	NONE
O4	young	myope	yes	normal	HARD
O5	young	hypermetrope	no	reduced	NONE
O6-O13	...	...	...	...	...
O14	pre-presbyo	hypermetrope	no	normal	SOFT
O15	pre-presbyo	hypermetrope	yes	reduced	NONE
O16	pre-presbyo	hypermetrope	yes	normal	NONE
O17	presbyopic	myope	no	reduced	NONE
O18	presbyopic	myope	no	normal	NONE
O19-O23	...	...	...	...	...
O24	presbyopic	hypermetrope	yes	normal	NONE

Decision tree for contact lenses recommendation



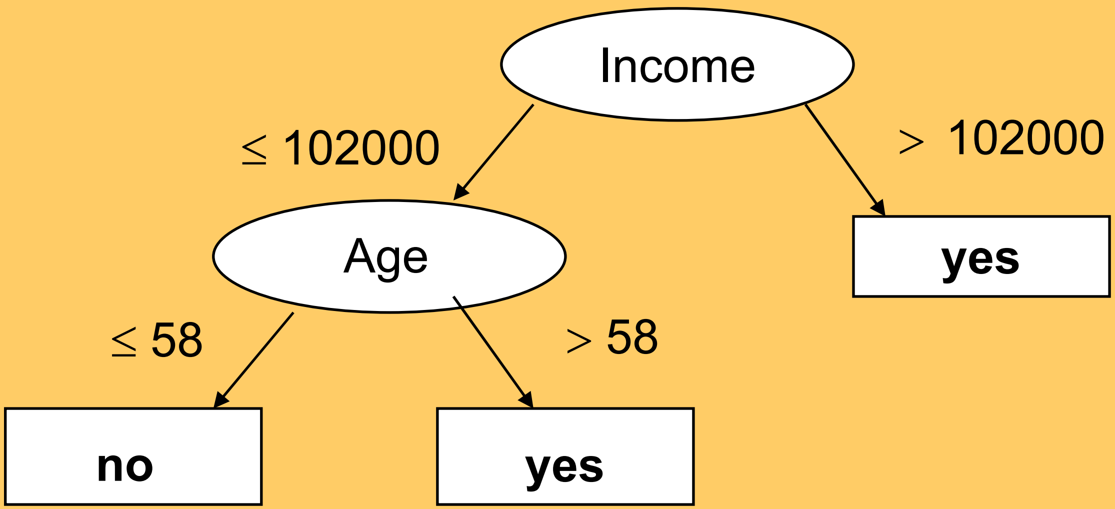
Decision tree for contact lenses recommendation

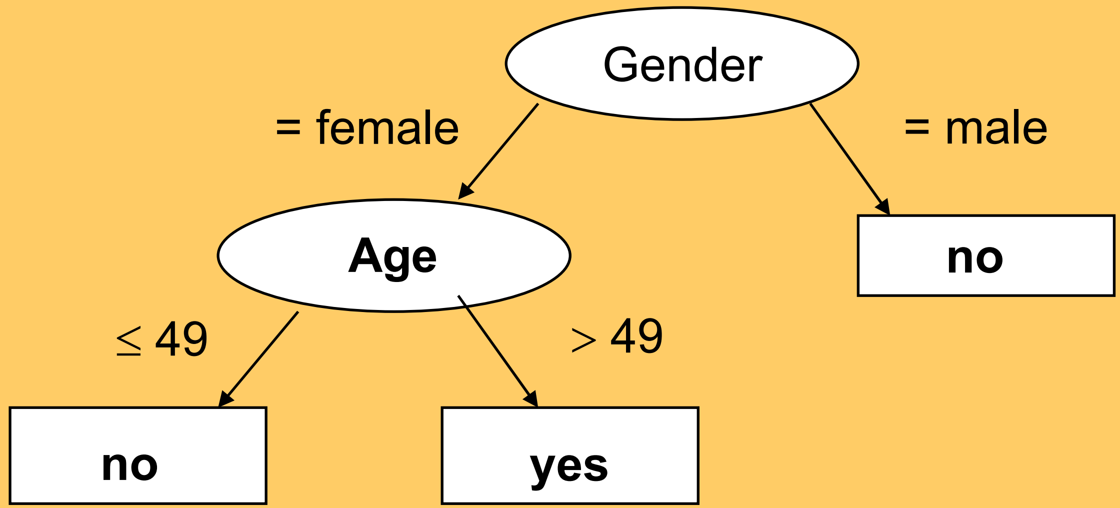


Illustrative example: Customer data

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

Induced decision trees





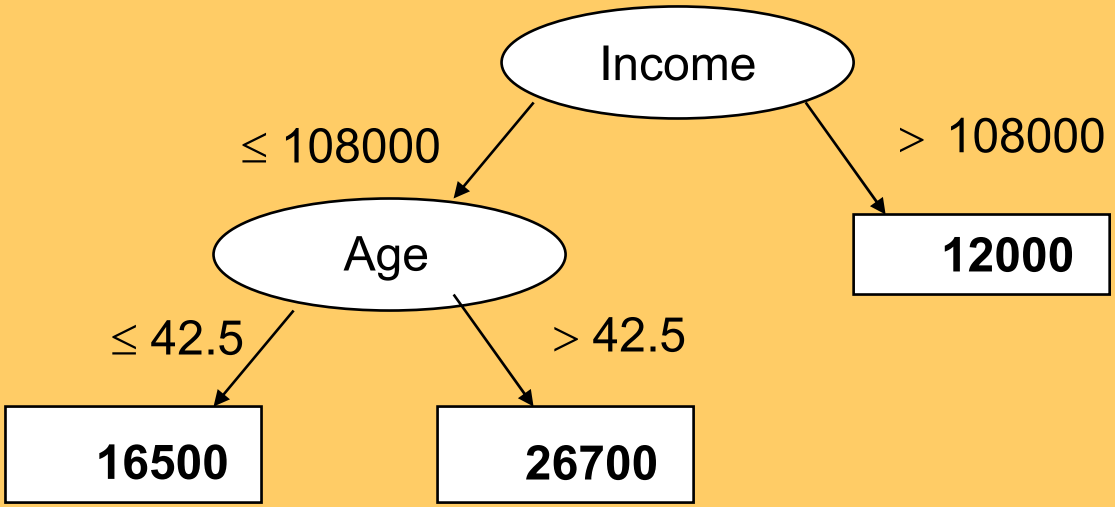
Predictive DM - Estimation

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

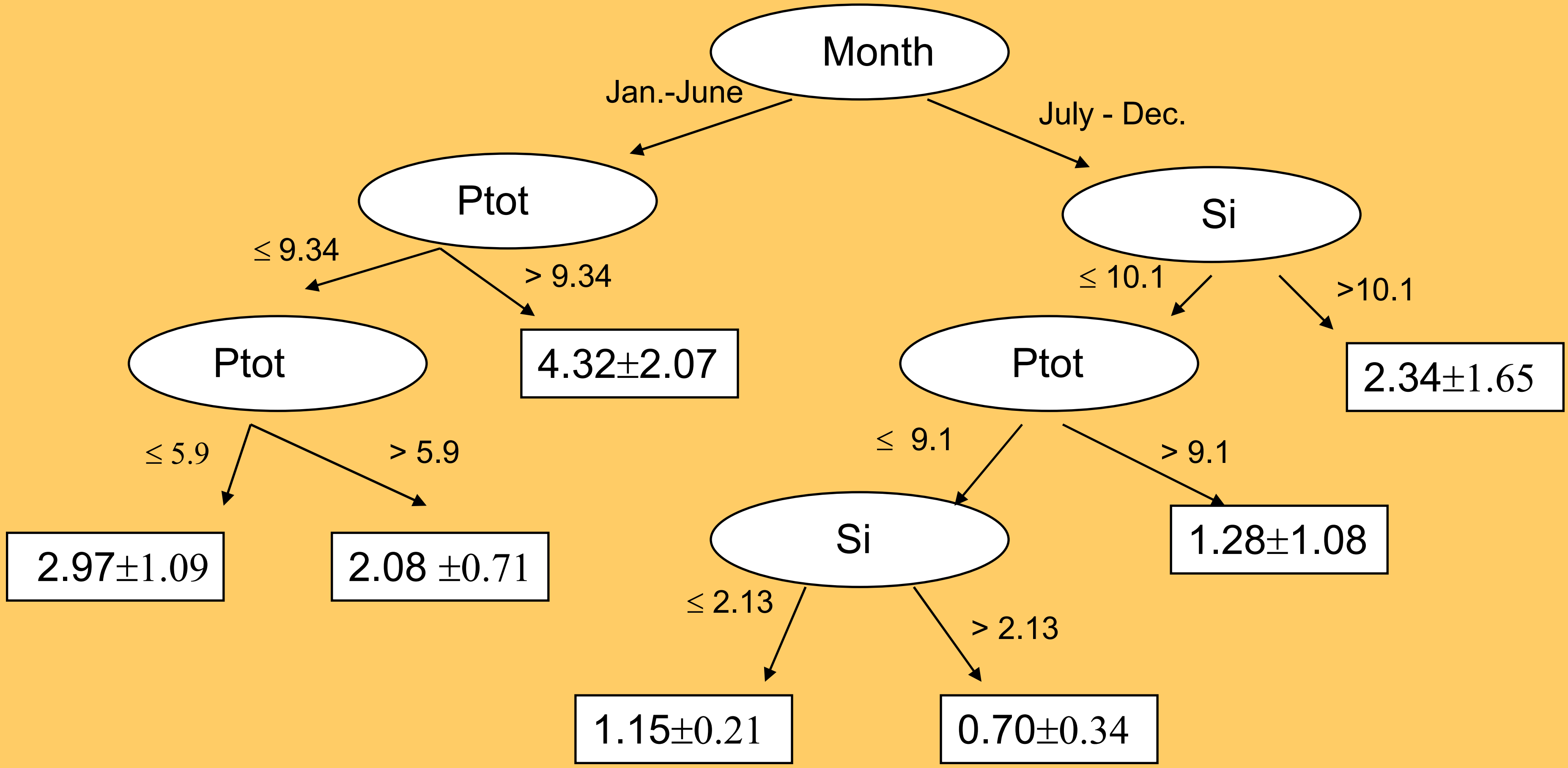
Illustrative example: Customer data

Customer	Gender	Age	Income	Spent	
c1	male	30	214000	18800	
c2	female	19	139000	15100	
c3	male	55	50000	12400	
c4	female	48	26000	8600	
c5	male	63	191000	28100	
O6-O13	...	...	...	...	
c14	female	61	95000	18100	
c15	male	56	44000	12000	
c16	male	36	102000	13800	
c17	female	57	215000	29300	
c18	male	33	67000	9700	
c19	female	26	95000	11000	
c20	female	55	214000	28800	

Customer data: regression tree



Predicting algal biomass: regression tree



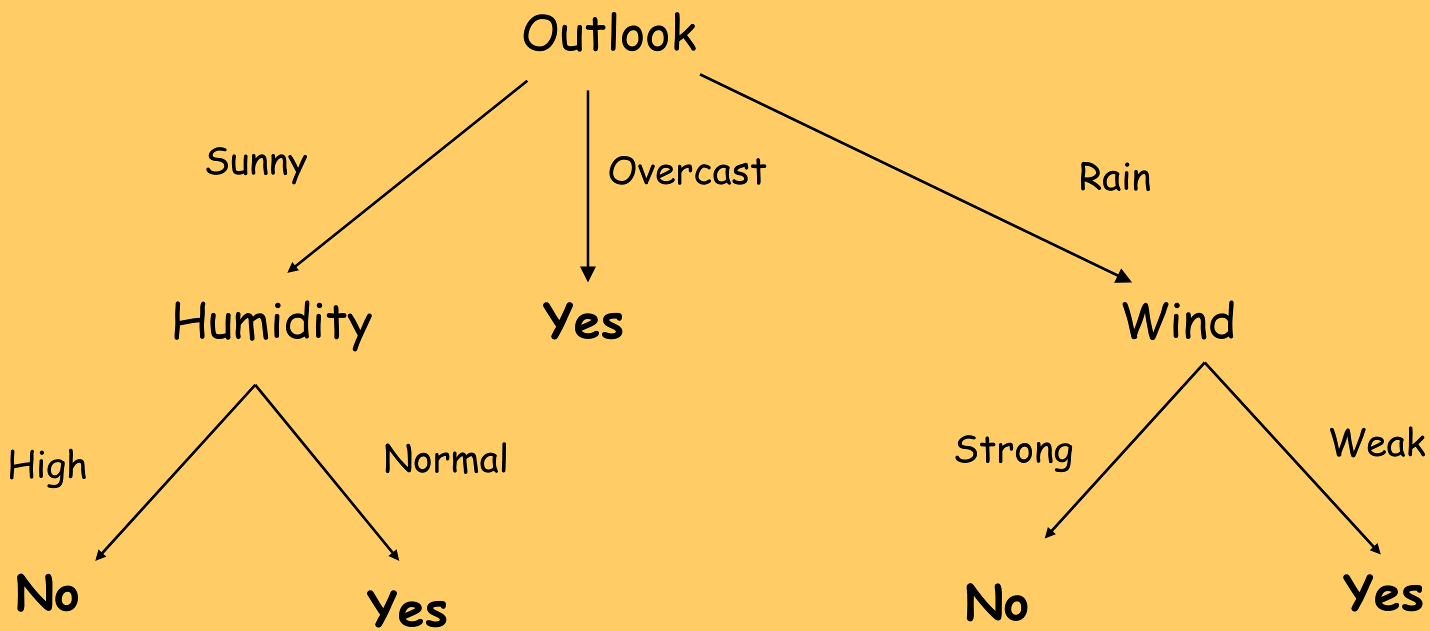
Decision tree learning

- Top-Down Induction of Decision Trees (TDIDT, Chapter 3 of Mitchell's book)
- decision tree representation
- the ID3 learning algorithm (Quinlan 1986)
- heuristics: information gain (entropy minimization)
- overfitting, decision tree pruning
- brief on evaluating the quality of learned trees (more in Chapter 5)

PlayTennis: Training examples

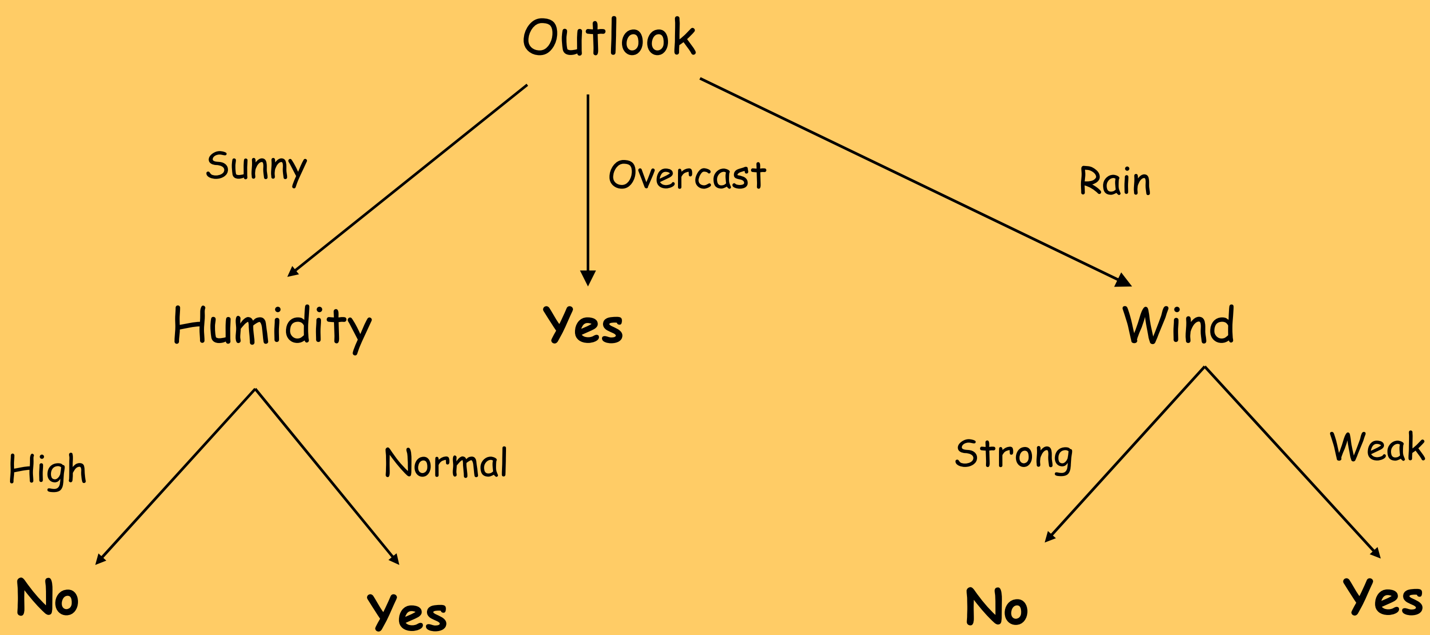
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision tree representation for PlayTennis



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

Decision tree representation for PlayTennis



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

(Outlook=Sunny $\wedge$ Humidity=Normal)

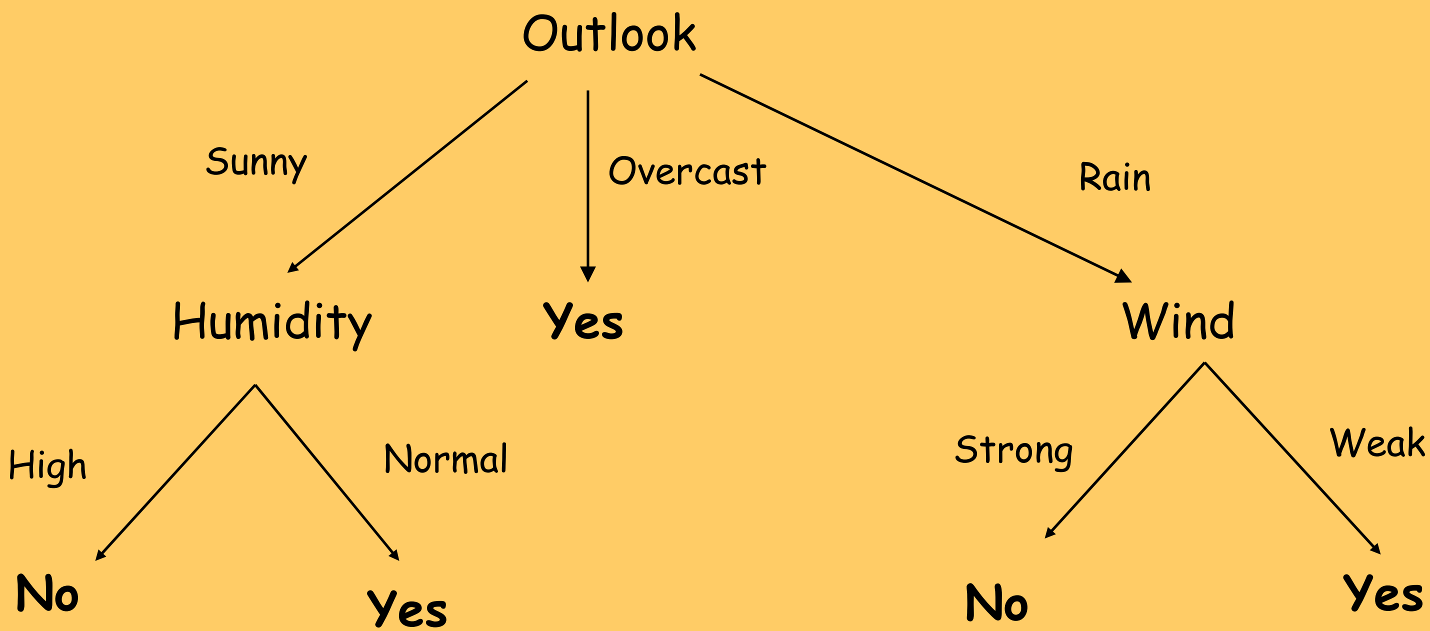
V (Outlook=Overcast)

V (Outlook=Rain $\wedge$ Wind=Weak)

PlayTennis: Other representations

- Logical expression for PlayTennis=Yes:
  - (Outlook=Sunny $\wedge$ Humidity=Normal) $\vee$ (Outlook=Overcast) $\vee$ (Outlook=Rain $\wedge$ Wind=Weak)
- If-then rules
  - **IF** Outlook=Sunny $\wedge$ Humidity=Normal **THEN** PlayTennis=Yes
  - **IF** Outlook=Overcast **THEN** PlayTennis=Yes
  - **IF** Outlook=Rain $\wedge$ Wind=Weak **THEN** PlayTennis=Yes
  - **IF** Outlook=Sunny $\wedge$ Humidity=High **THEN** PlayTennis=No
  - **IF** Outlook=Rain $\wedge$ Wind=Strong **THEN** PlayTennis=No

PlayTennis: Using a decision tree for classification



Is Saturday morning OK for playing tennis?

Outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong

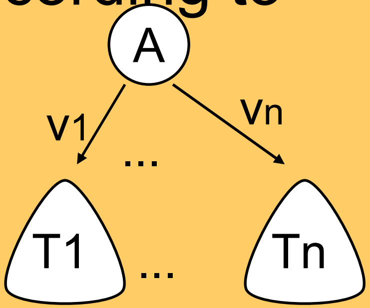
PlayTennis = No, because Outlook=Sunny $\wedge$ Humidity=High

Appropriate problems for decision tree learning

- Classification problems: classify an instance into one of a discrete set of possible categories (medical diagnosis, classifying loan applicants, …)
- Characteristics:
  - instances described by attribute-value pairs
    (discrete or real-valued attributes)
  - target function has discrete output values
    (boolean or multi-valued, if real-valued then regression trees)
  - disjunctive hypothesis may be required
  - training data may be noisy
    (classification errors and/or errors in attribute values)
  - training data may contain missing attribute values

Learning of decision trees

- ID3 (Quinlan 1979), CART (Breiman et al. 1984), C4.5, WEKA, ...
  - create the root node of the tree
  - if all examples from S belong to the same class Cj
    - then label the root with Cj
  - else
    - select the 'most informative' attribute **A** with values **v1, v2, … vn**
    - divide training set **S** into **S1,… , Sn** according to values **v1,…,vn**
    - recursively build sub-trees **T1,…,Tn** for **S1,…,Sn**



Search heuristics in ID3

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

Entropy

- **S** - training set, $\mathbf{C_1,...,C_N}$ - classes
- **Entropy E(S)** – measure of the impurity of training set S

$$E(S) = -\sum_{c=1}^{N} p_c . \log_2 p_c$$

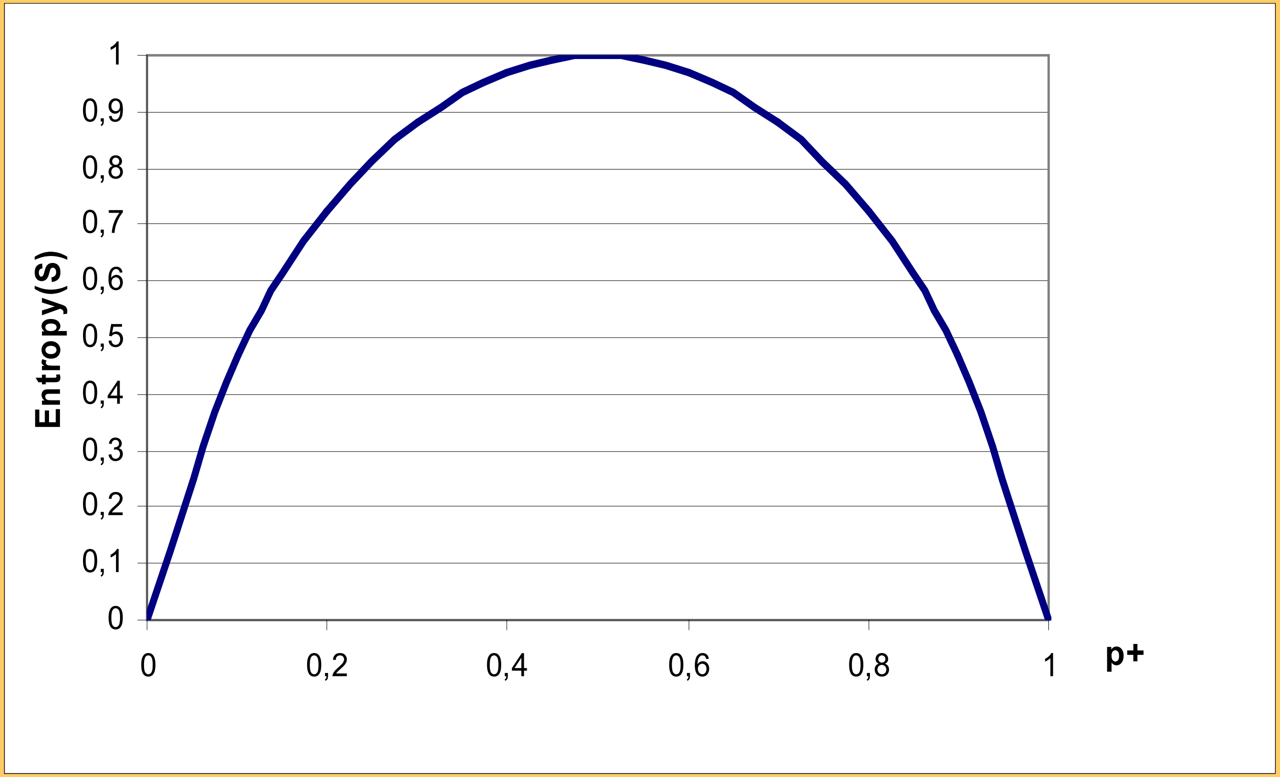
$\mathbf{p_c}$ - prior probability of class $\mathbf{C_c}$
(relative frequency of $\mathbf{C_c}$ in **S**)

- Entropy in binary classification problems

$$\mathbf{E(S) = - p_+ \log_2 p_+ - p_- \log_2 p_-}$$

Entropy

- $E(S) = - p_+ \log_2 p_+ - p_- \log_2 p_-$
- The entropy function relative to a Boolean classification, as the proportion $\mathbf{p_+}$ 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:

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

PlayTennis: Entropy

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

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

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

  $= 0.940$

PlayTennis: Entropy

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

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

Humidity?
- $\xrightarrow{\text{High}}$ [3+, 4-] E=0.985
- $\xrightarrow{\text{Normal}}$ [6+, 1-] E=0.592

Wind?
- $\xrightarrow{\text{Weak}}$ [6+, 2-] E=0.811
- $\xrightarrow{\text{Strong}}$ [3+, 3-] E=1.00

Information gain search heuristic

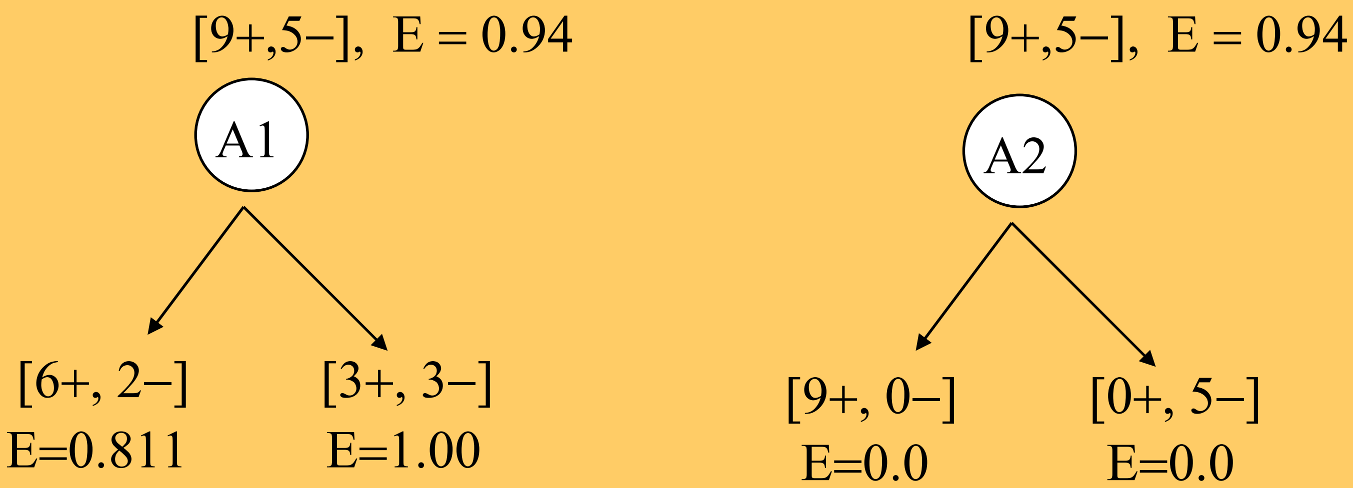
- **Information gain** measure is aimed to minimize the number of tests needed for the classification of a new object
- **Gain(S,A)** – expected reduction in entropy of S due to sorting on A

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

- **Most informative** attribute: **max Gain(S,A)**

Information gain search heuristic

- Which attribute is more informative, A1 or A2 ?



- Gain(S,A1) = 0.94 – (8/14 x 0.811 + 6/14 x 1.00) = 0.048
- Gain(S,A2) = 0.94 – 0 = 0.94 A2 has max Gain

PlayTennis: Information gain

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

- Values(Wind) = {Weak, Strong}

$$\text{Wind?} \begin{cases} \xrightarrow{\text{Weak}} [6+, 2-] \quad E=0.811 \\ \xrightarrow{\text{Strong}} [3+, 3-] \quad E=1.00 \end{cases}$$

  - S = [9+,5-], E(S) = 0.940
  - S_{weak} = [6+,2-], E(S_{weak}) = 0.811
  - S_{strong} = [3+,3-], E(S_{strong}) = 1.0
  - **Gain(S,Wind) =** E(S) - (8/14)E(S_{weak}) - (6/14)E(S_{strong}) = 0.940 - (8/14)x0.811 - (6/14)x1.0=**0.048**

PlayTennis: Information gain

- **Which attribute is the best?**
  - Gain(S,Outlook)=0.246 *MAX !*
  - Gain(S,Humidity)=0.151
  - Gain(S,Wind)=0.048
  - Gain(S,Temperature)=0.029

PlayTennis: Information gain

Outlook?

Rain → {D4,D5,D6,D10,D14} [3+, 2-] E > 0 **???**

Overcast → {D3,D7,D12,D13} [4+, 0-] E = 0 **OK - assign class Yes**

Sunny → {D1,D2,D8,D9,D11} [2+, 3-] E > 0 **???**

- Which attribute should be tested here?
  - Gain(S_{sunny}, Humidity) = 0.97-(3/5)0-(2/5)0 = 0.970 ***MAX !***
  - Gain(S_{sunny},Temperature) = 0.97-(2/5)0-(2/5)1-(1/5)0 = 0.570
  - Gain(S_{sunny},Wind) = 0.97-(2/5)1-(3/5)0.918 = 0.019

Probability estimates

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

$$p(Class \mid Cond) =$$
$$= \frac{n(Class.Cond)}{n(Cond)}$$

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

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

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

$$= \frac{n(Class.Cond)+1}{n(Cond)+k} \quad k=2$$

[6+,1-] (7) = 6+1 / 7+2 = 7/9

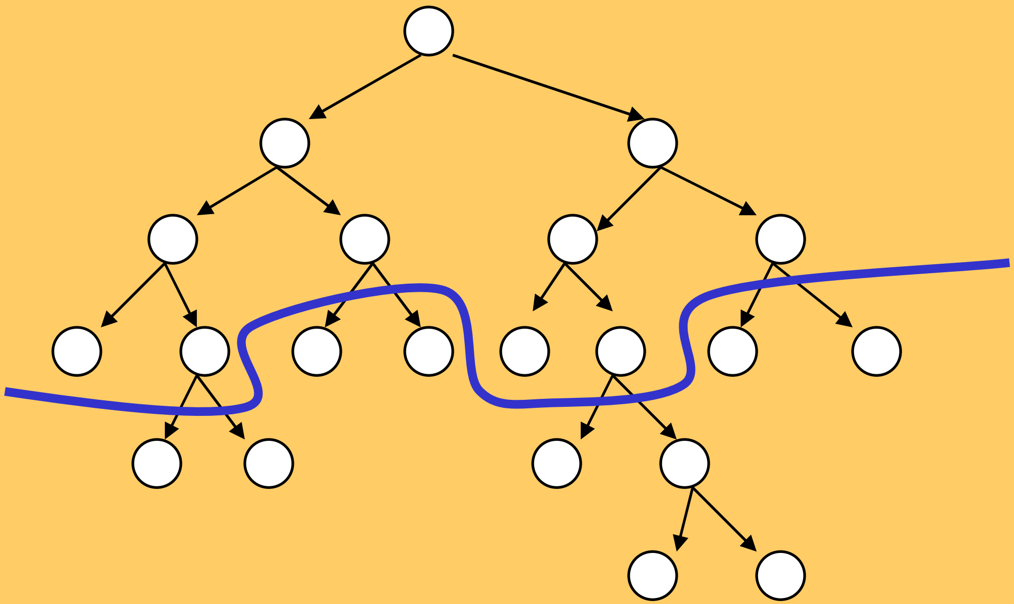
[2+,0-] (2) = 2+1 / 2+2 = 3/4

Heuristic search in ID3

- **Search bias:** Search the space of decision trees from simplest to increasingly complex (greedy search, no backtracking, prefer small trees)
- **Search heuristics:** At a node, select the attribute that is most useful for classifying examples, split the node accordingly
- **Stopping criteria:** A node becomes a leaf
  - if all examples belong to same class C_j, label the leaf with C_j
  - if all attributes were used, label the leaf with the most common value C_k of examples in the node
- **Extension to ID3:** handling noise - tree pruning

Pruning of decision trees

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



Handling noise – Tree pruning

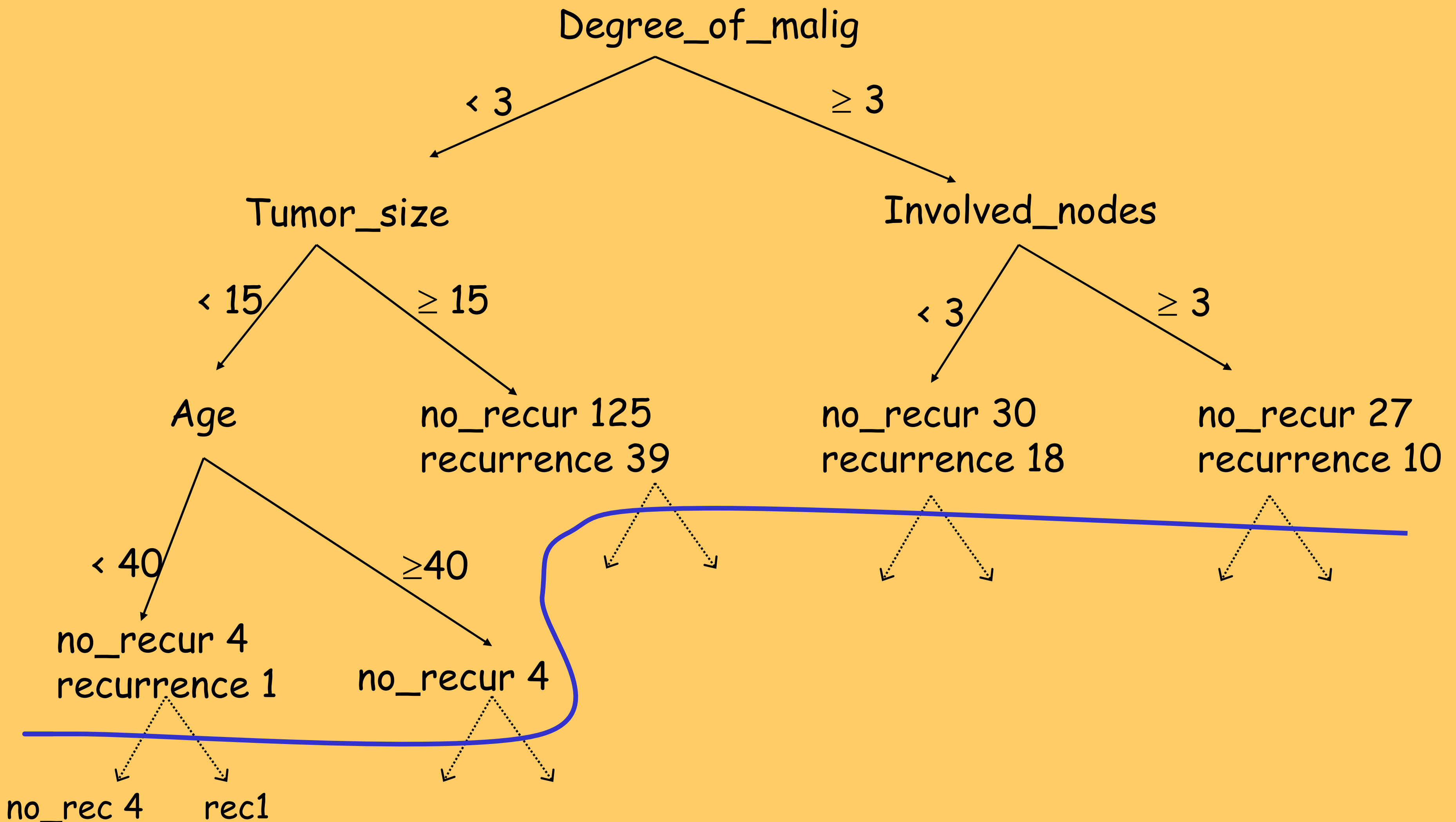
Sources of imperfection

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

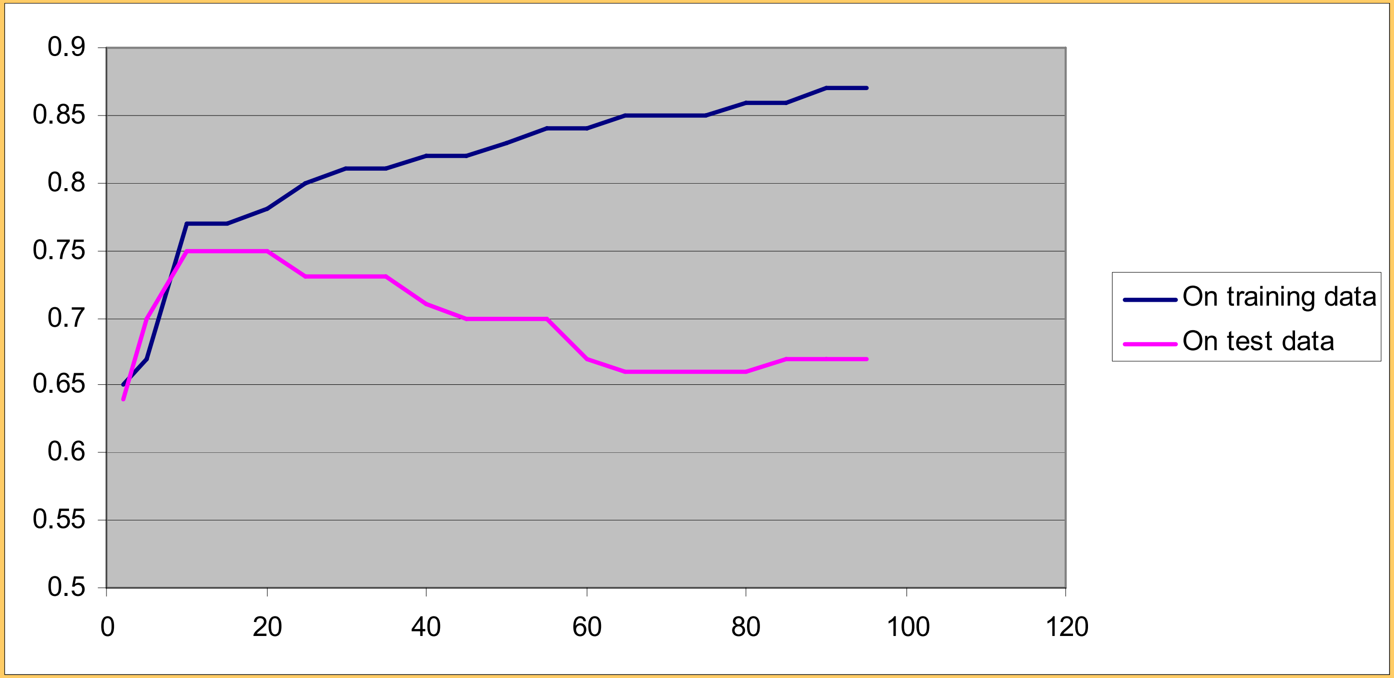


Accuracy and error

- Accuracy: percentage of correct classifications
  - on the training set
  - on unseen instances
- How accurate is a decision tree when classifying unseen instances
  - An estimate of accuracy on unseen instances can be computed, e.g., by averaging over 4 runs:
    - split the example set into training set (e.g. 70%) and test set (e.g. 30%)
    - induce a decision tree from training set, compute its accuracy on test set
- Error = 1 - Accuracy
- High error may indicate data overfitting

Overfitting and accuracy

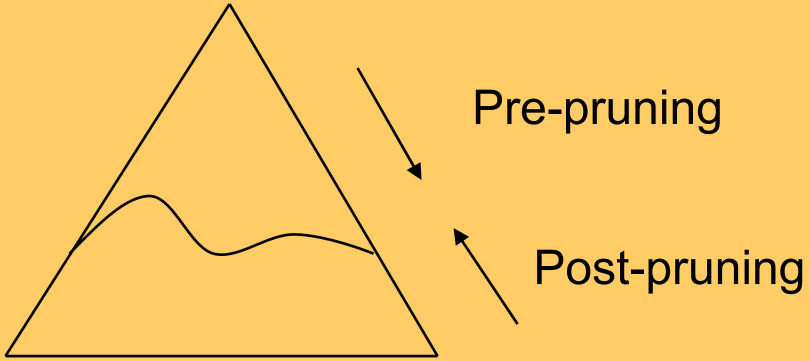
- Typical relation between tree size and accuracy



- Question: how to prune optimally?

Avoiding overfitting

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



- forward pruning considered inferior (myopic)
- post pruning makes use of sub trees

How to select the “best” tree

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

Selected decision/regression tree learners

- Decision tree learners
  - ID3 (Quinlan 1979)
  - CART (Breiman et al. 1984)
  - Assistant (Cestnik et al. 1987)
  - C4.5 (Quinlan 1993), C5 (See5, Quinlan)
  - J48 (available in WEKA)
- Regression tree learners, model tree learners
  - M5, M5P (implemented in WEKA)

Features of C4.5

- Implemented as part of the WEKA data mining workbench
- Handling noisy data: post-pruning
- Handling incompletely specified training instances: ‘unknown’ values (**?**)
  - in learning assign conditional probability of value v: $p(v|C) = p(vC) / p(C)$
  - 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 automaticaly)
  - for discrete values try grouping the values until two groups remain *
- ‘Majority’ classification in NULL leaf (with no corresponding training example)
  - if an example ‘falls’ into a NULL leaf during classification, the class assigned to this example is the majority class of the parent of the NULL leaf

* the basic C4.5 doesn’t support binarisation of discrete attributes, it supports grouping

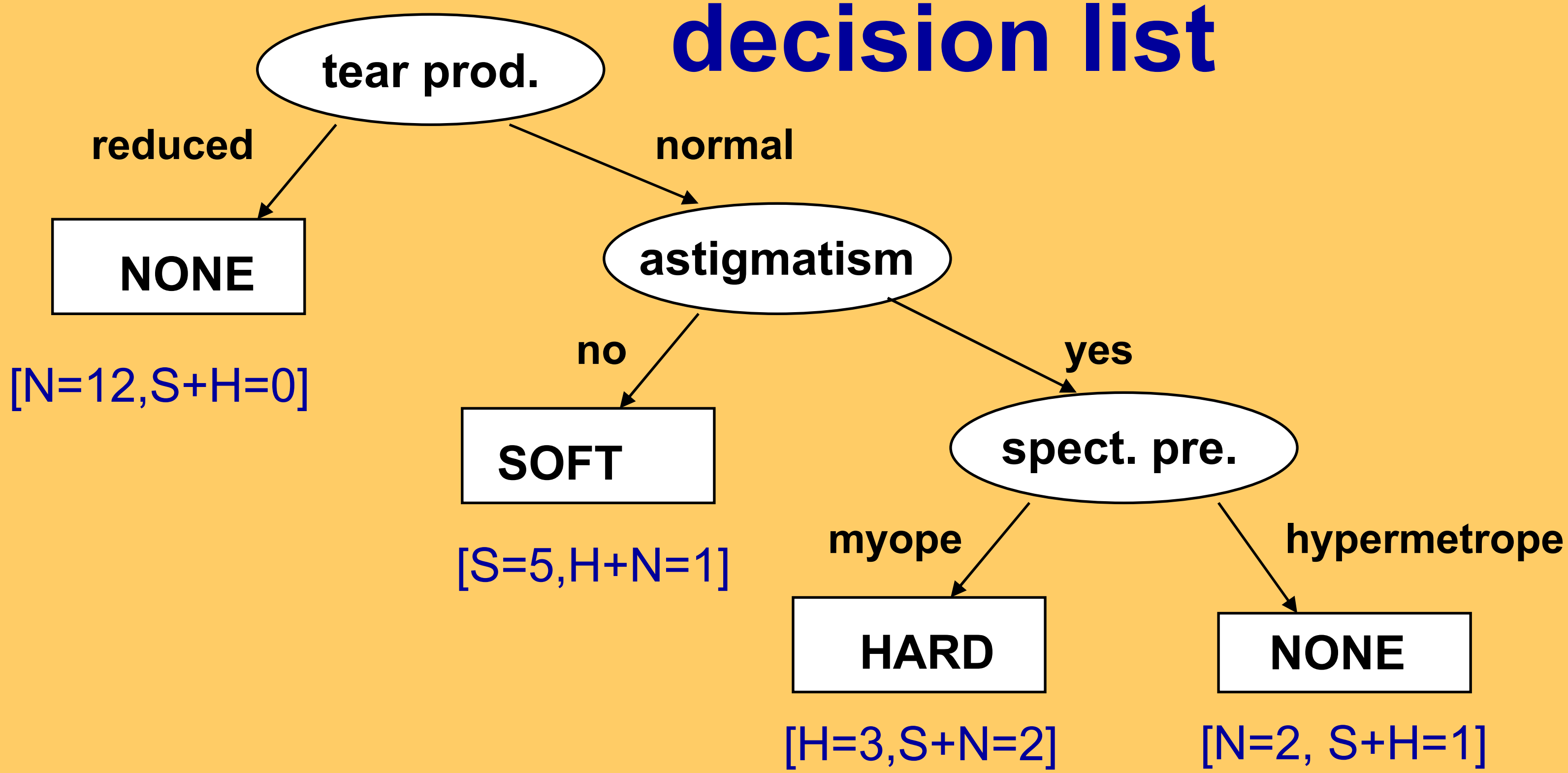
Part II. Predictive DM techniques

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

Rule learning

- 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

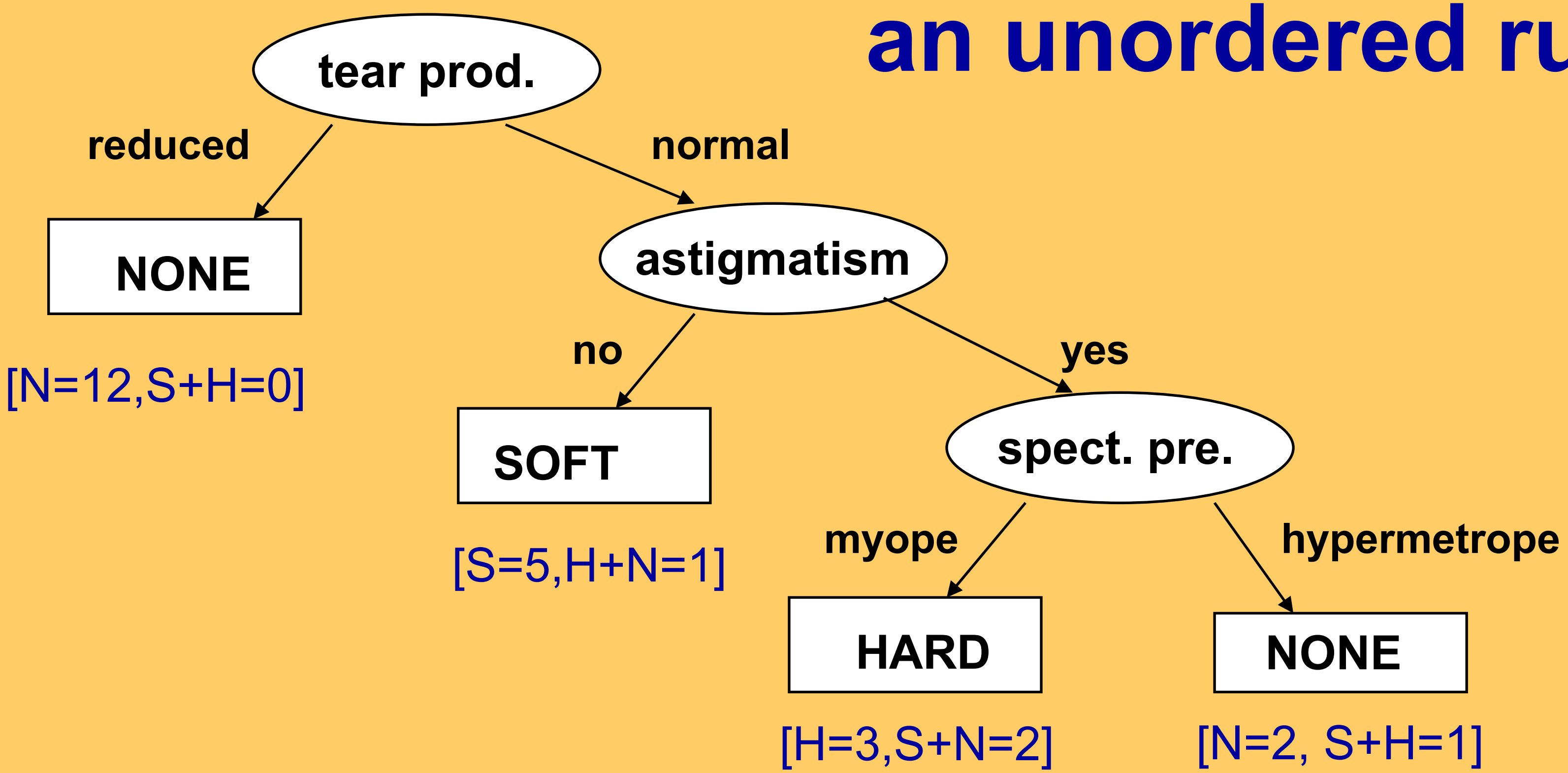
Contact lenses: convert decision tree to decision list



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

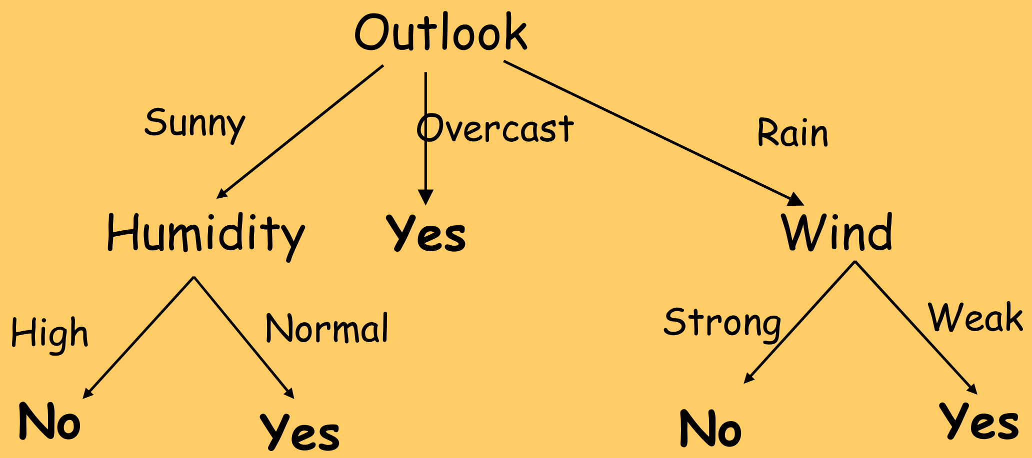
Contact lenses: convert decision tree to an unordered rule set



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)

PlayTennis: Converting a tree to rules



IF Outlook=Sunny $\wedge$ Humidity=Normal **THEN** PlayTennis=Yes

IF Outlook=Overcast **THEN** PlayTennis=Yes

IF Outlook=Rain $\wedge$ Wind=Weak **THEN** PlayTennis=Yes

IF Outlook=Sunny $\wedge$ Humidity=High **THEN** PlayTennis=No

IF Outlook=Rain $\wedge$ Wind=Strong **THEN** PlayTennis=No

Rule post-pruning (Quinlan 1993)

- Very frequently used method, e.g., in C4.5
- 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

Rule set representation

- Rule base is a disjunctive set of conjunctive rules
- Standard form of rules:

  IF Condition THEN Class

  Class IF Conditions

  Class $\leftarrow$ Conditions

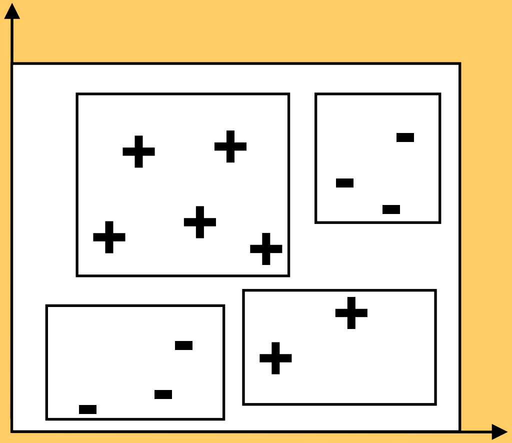
IF Outlook=Sunny $\wedge$ Humidity=Normal **THEN** PlayTennis=Yes
IF Outlook=Overcast **THEN** PlayTennis=Yes
IF Outlook=Rain $\wedge$ Wind=Weak **THEN** PlayTennis=Yes

- Form of CN2 rules:

  IF Conditions THEN MajClass [ClassDistr]

- Rule base: {R1, R2, R3, …, DefaultRule}

Original covering algorithm (AQ, Michalski 1969,86)



Basic covering algorithm

for each class Ci **do**

- Ei := Pi U Ni (Pi pos., Ni neg.)
- RuleBase(Ci) := empty
- **repeat {learn-set-of-rules}**
  - **learn-one-rule** R covering some positive examples and no negatives
  - add R to RuleBase(Ci)
  - delete from Pi all pos. ex. covered by R
- **until** Pi = empty

Heuristics for learn-one-rule: PlayTennis example

PlayTennis = yes [9+,5-] (14)

PlayTennis = yes $\leftarrow$ Wind=weak [6+,2-] (8)

$\leftarrow$ Wind=strong [3+,3-] (6)

$\leftarrow$ Humidity=normal [6+,1-] (7)

$\leftarrow$ …

PlayTennis = yes $\leftarrow$ Humidity=normal

Outlook=sunny [2+,0-] (2)

$\leftarrow$ …

Estimating **rule accuracy (rule precision)** with the **probability** that a covered example is positive

A(Class $\leftarrow$ Cond) = p(Class| Cond)

Estimating the **probability** with the **relative frequency** of covered pos. ex. / all covered ex.

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

Probability estimates

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

$$p(Class \mid Cond) = \\ = \frac{n(Class.Cond)}{n(Cond)}$$

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

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

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

$$= \frac{n(Class.Cond) + 1}{n(Cond) + k} \quad k = 2$$

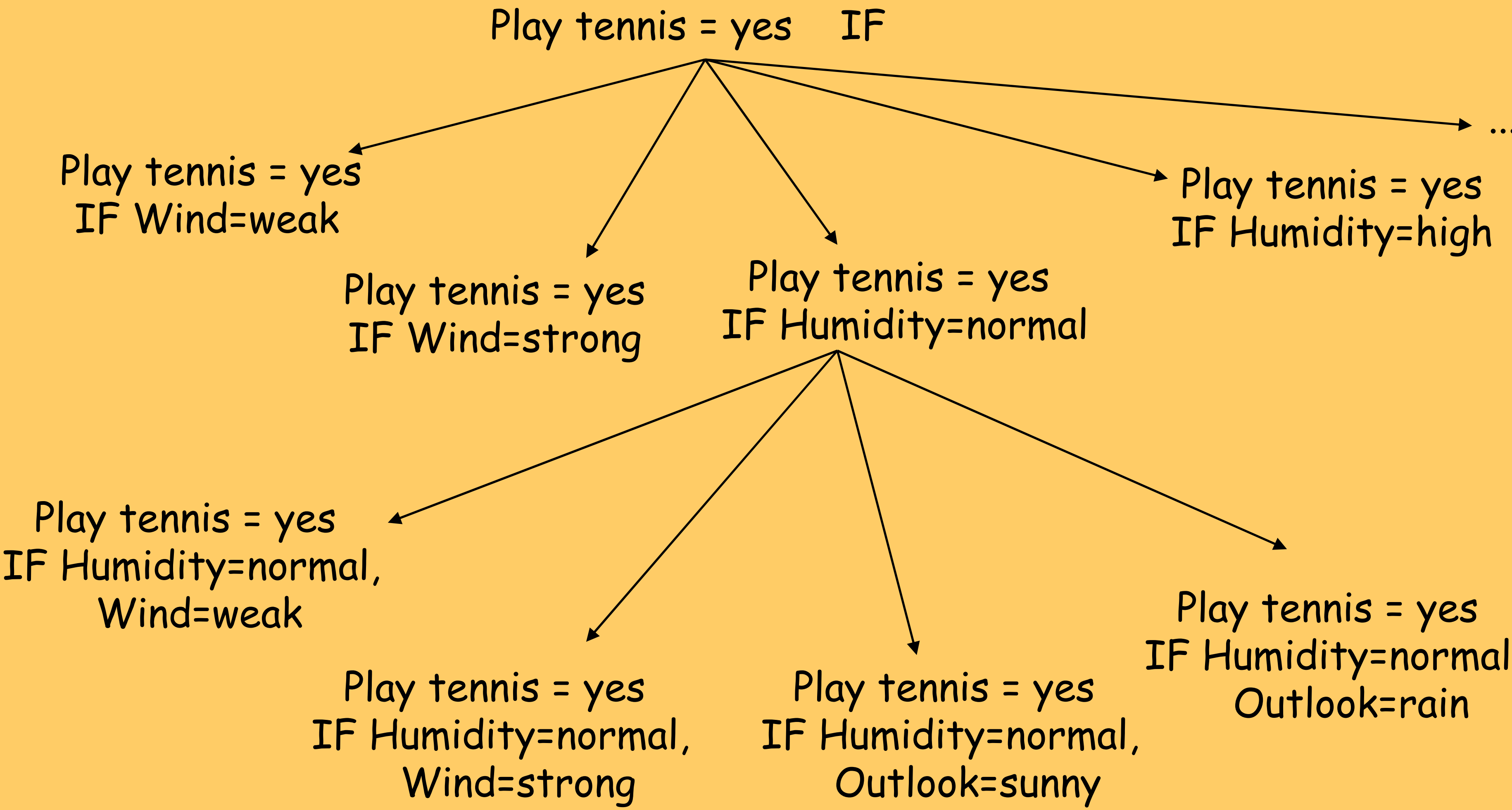
[6+,1-] (7) = 6+1 / 7+2 = 7/9

[2+,0-] (2) = 2+1 / 2+2 = 3/4

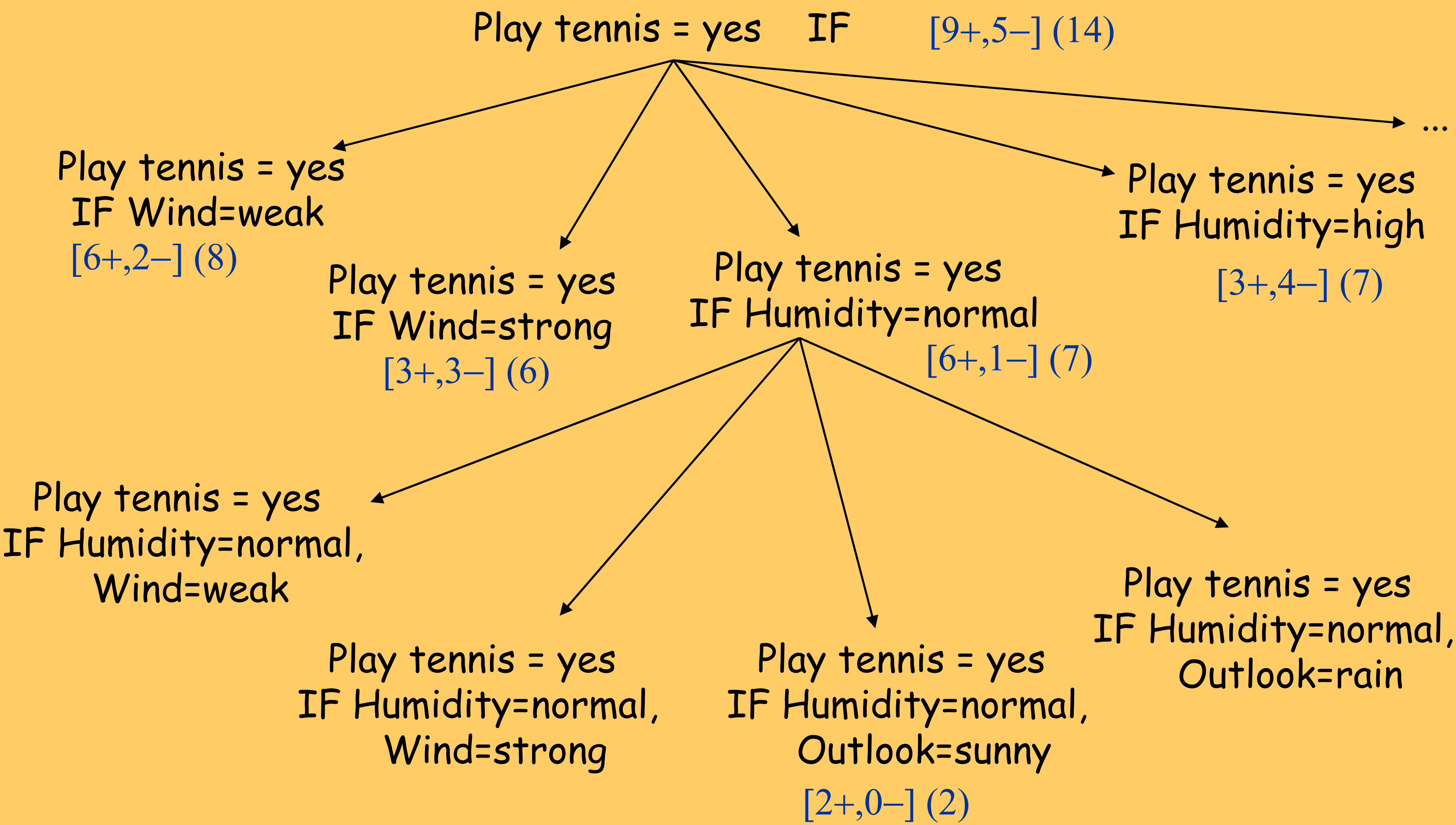
Learn-one-rule: Greedy vs. beam search

- learn-one-rule by greedy general-to-specific search, at each step selecting the `best' descendant, no backtracking
- beam search: maintain a list of k best candidates at each step; descendants (specializations) of each of these k candidates are generated, and the resulting set is again reduced to k best candidates

Learn-one-rule as search: PlayTennis example



Learn-one-rule as heuristic search: PlayTennis example



What is “high” rule accuracy (rule precision) ?

- Rule evaluation measures - aimed at avoiding overfitting
- Predictive evaluation measures: aimed at maximizing classification accuracy, minimizing Error = 1 - Accuracy, avoiding overfitting
- Rule accuracy/precision should be traded off against the “default” accuracy/precision of the rule **Cl ←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***
  - RAcc(Cl ←Cond) = p(Cl | Cond) – p(Cl)

Weighted relative accuracy

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

  WRAcc(Cl←Cond) = p(Cond) . [p(Cl | Cond) – p(Cl)]

- WRAcc is a fundamental rule evaluation measure:
  - WRAcc can be used if you want to assess both accuracy and significance
  - WRAcc can be used if you want to compare rules with different heads **and** bodies

Learn-one-rule: search heuristics

- Assume two classes (+,-), learn rules for + class (Cl). Search for specializations of one rule R = Cl $\leftarrow$ 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) . (p(Cl|Cond') - p(Cl|Cond))
- **Weighted relative accuracy** trades off coverage and relative accuracy WRAcc(R) = p(Cond).(p(Cl|Cond) - p_a(Cl))

Probabilistic classification

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

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

Part II. Predictive DM techniques

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

Classifier evaluation

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

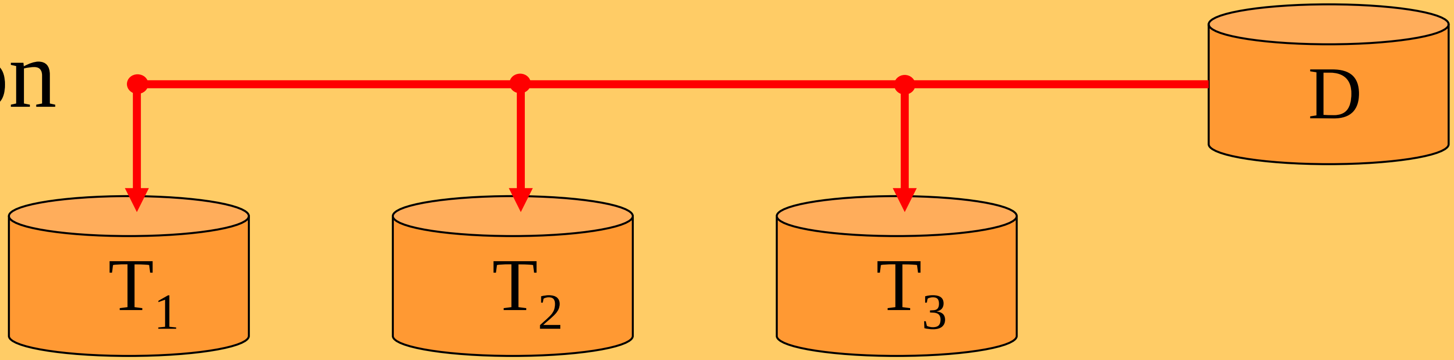
Evaluating hypotheses

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

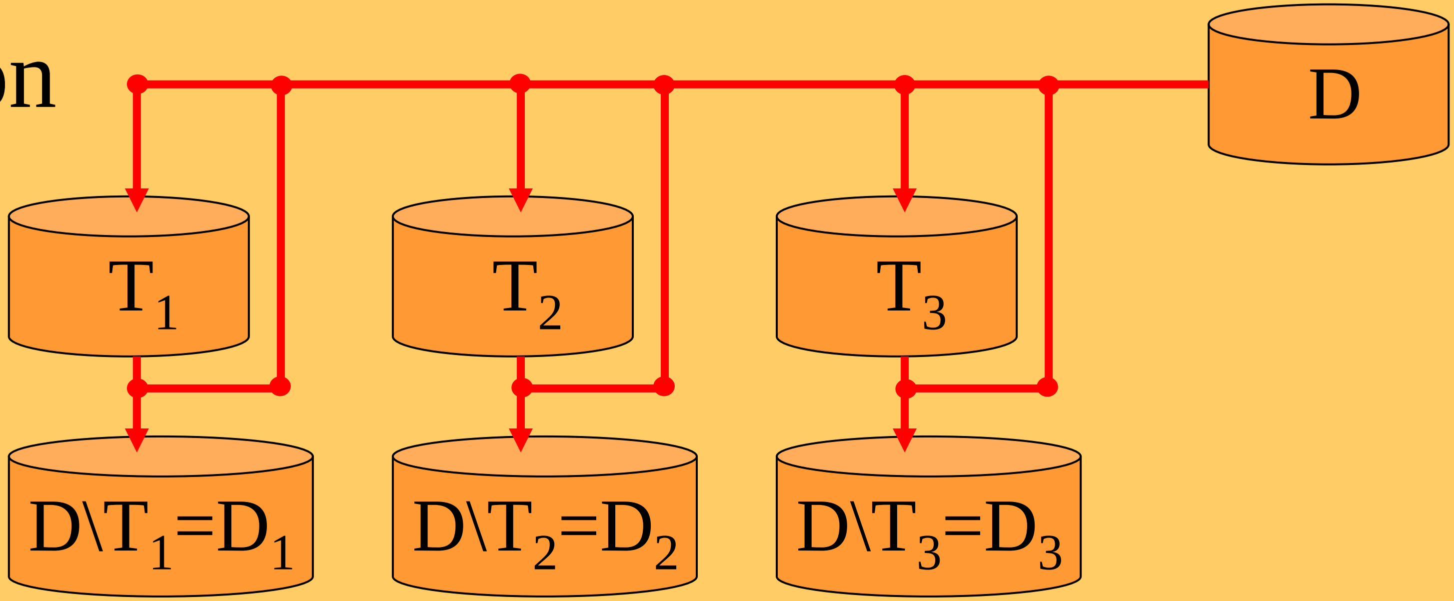
n-fold cross validation

- A method for accuracy estimation of classifiers
- Partition set D into n disjoint, almost equally-sized folds T_i where $\bigcup_i T_i = D$
- **for** i = 1, ..., n **do**
  - form a training set out of n-1 folds: $Di = D \setminus T_i$
  - induce classifier H_i from examples in Di
  - 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

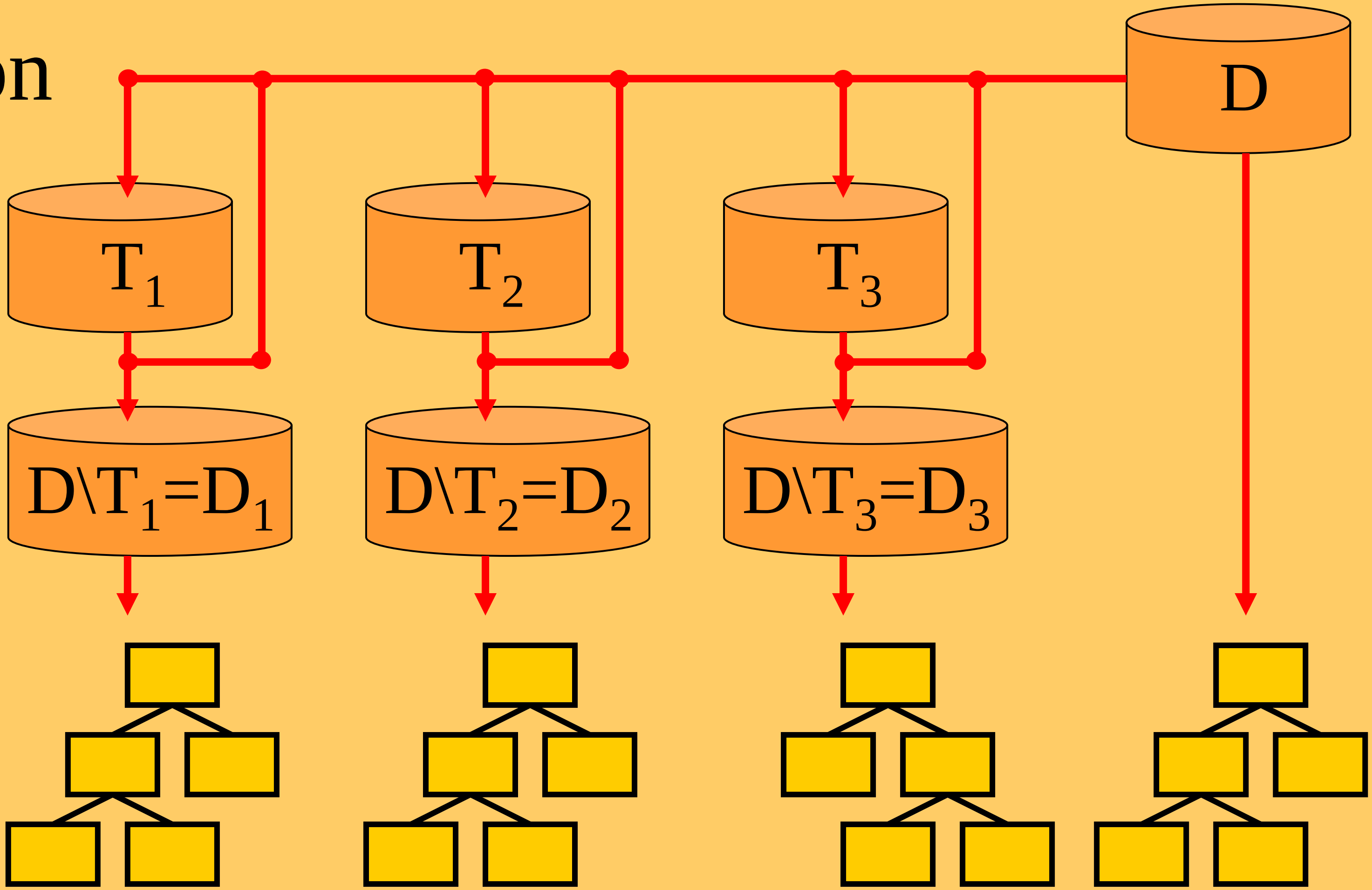
- Partition



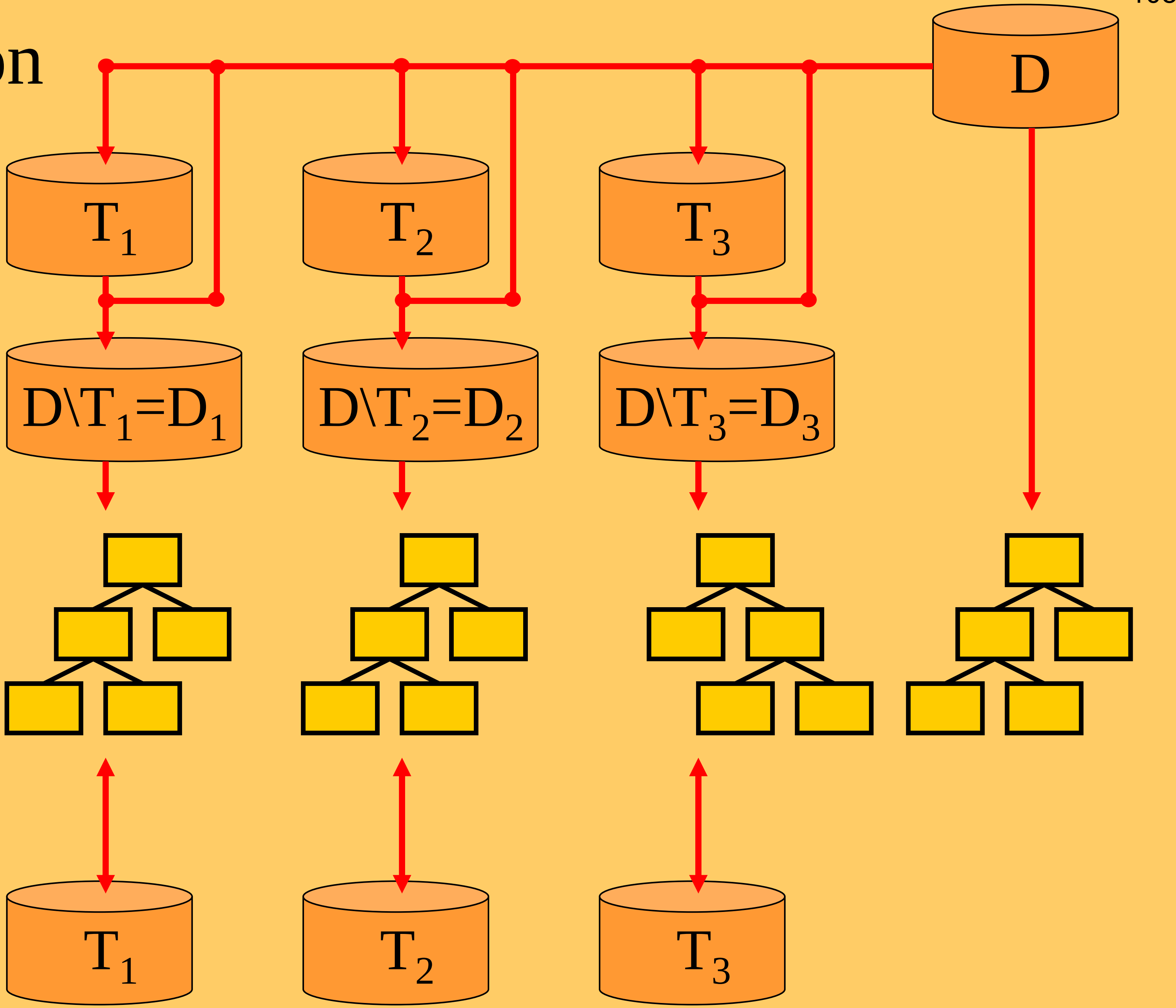
- Partition
- Train



- Partition
- Train



- Partition
- Train
- Test



Confusion matrix and rule (in)accuracy

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

Confusion matrix

	Predicted positive	Predicted negative	
Positive examples	**True positives**	**False negatives**	
Negative examples	**False positives**	**True negatives**	

- also called ***contingency table***

Classifier 1

	Predicted positive	Predicted negative	
Positive examples	**40**	**10**	50
Negative examples	**10**	**40**	50
	50	50	100

Classifier 2

	Predicted positive	Predicted negative	
Positive examples	**30**	**20**	50
Negative examples	**0**	**50**	50
	30	70	100

ROC space

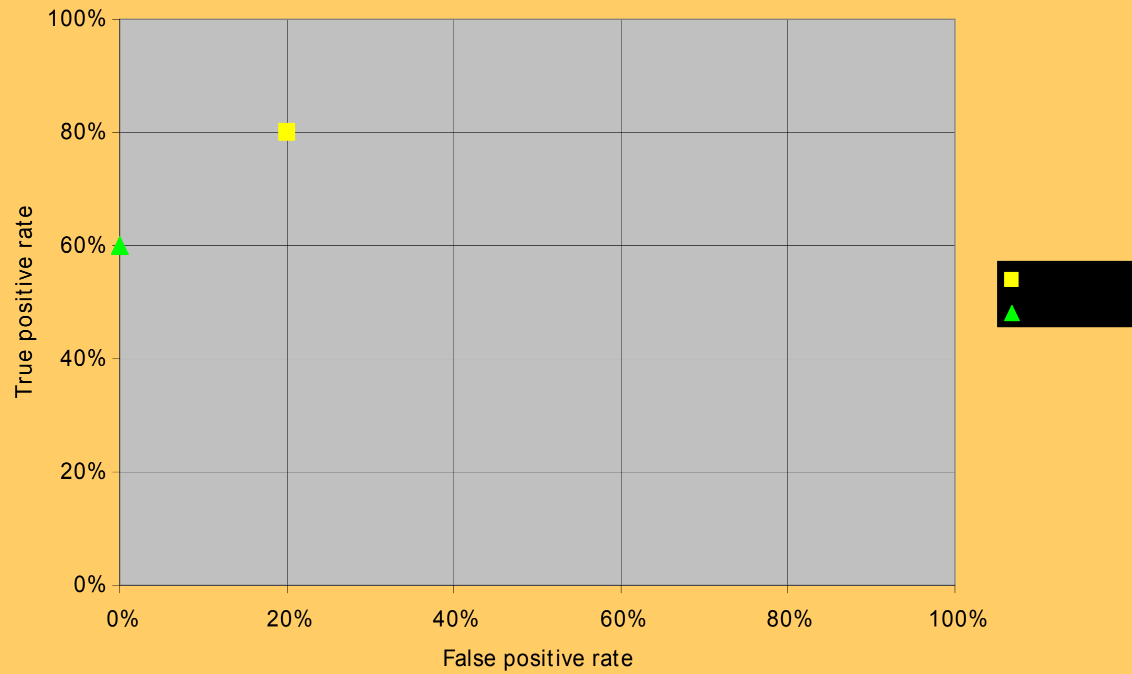
- ***True positive rate*** = #true pos. / #pos.
  - $TPr_1 = 40/50 = 80\%$
  - $TPr_2 = 30/50 = 60\%$
- ***False positive rate*** = #false pos. / #neg.
  - $FPr_1 = 10/50 = 20\%$
  - $FPr_2 = 0/50 = 0\%$
- ***ROC space*** has
  - FPr on X axis
  - TPr on Y axis

Classifier 1

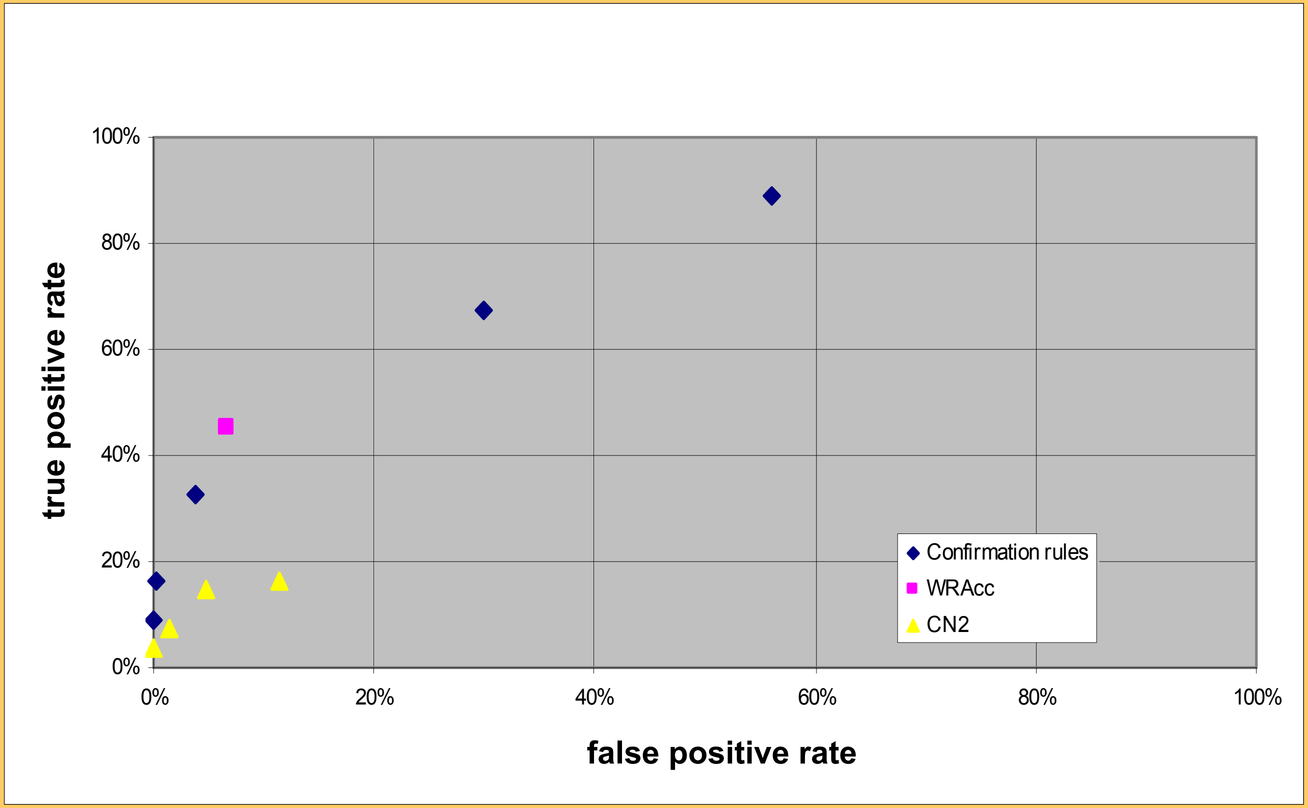
	Predicted positive	Predicted negative	
Positive examples	**40**	**10**	50
Negative examples	**10**	**40**	50
	50	50	100

Classifier 2

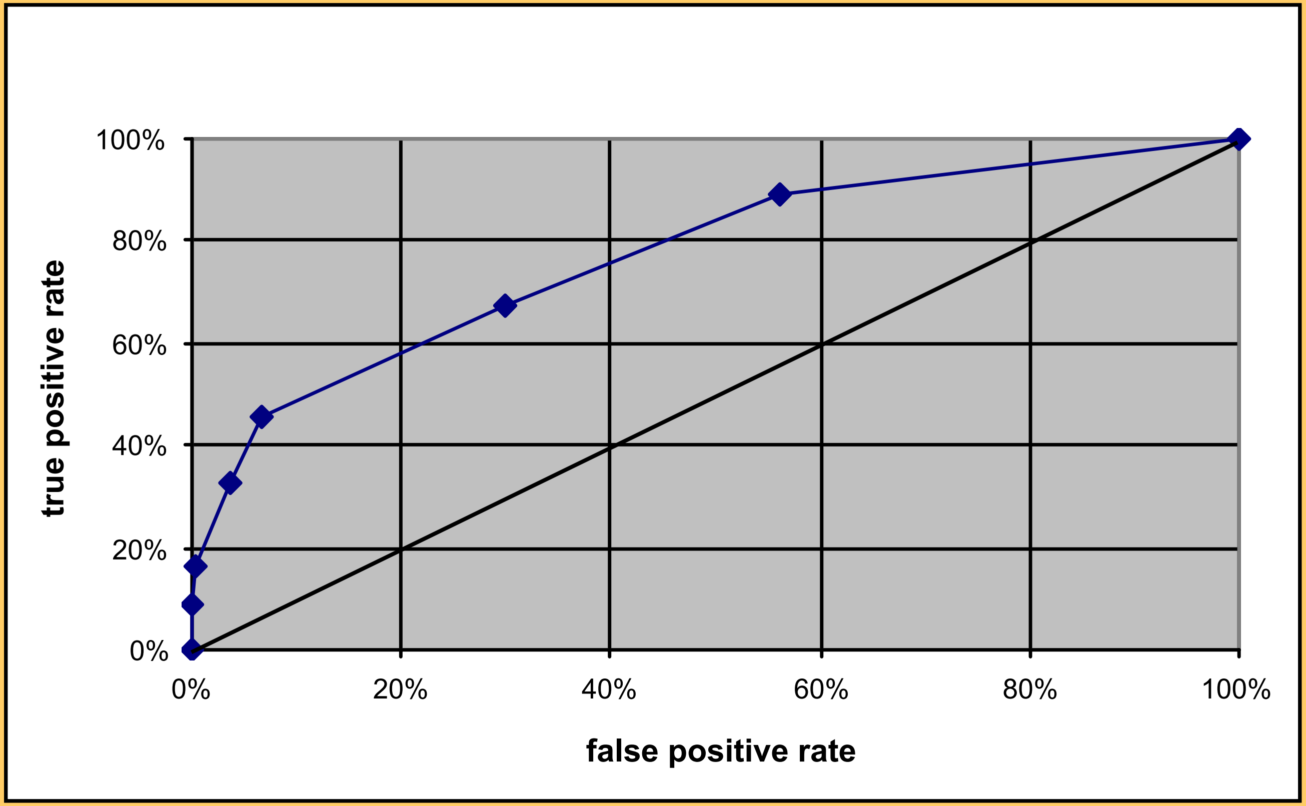
	Predicted positive	Predicted negative	
Positive examples	**30**	**20**	50
Negative examples	**0**	**50**	50
	30	70	100



The ROC space



The ROC convex hull



Part III. Descriptive DM techniques

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

Descriptive DM

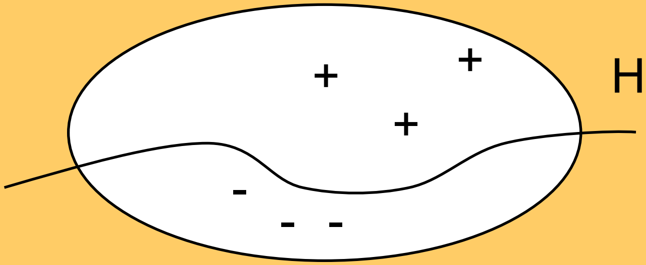
- Often used for preliminary explanatory data analysis
- User gets feel for the data and its structure
- Aims at deriving descriptions of characteristics of the data
- Visualization and descriptive statistical techniques can be used

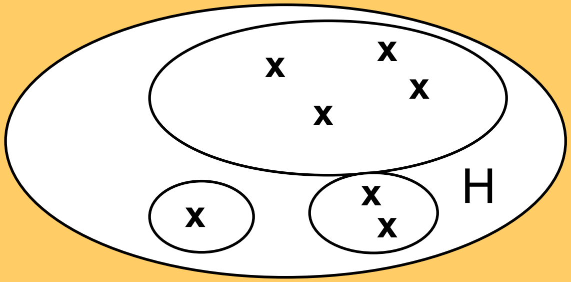
Descriptive DM

- **Description**
  - Data description and summarization: describe elementary and aggregated data characteristics (statistics, …)
  - Dependency analysis:
    - describe associations, dependencies, …
    - discovery of properties and constraints
- **Segmentation**
  - Clustering: separate objects into subsets according to distance and/or similarity (clustering, SOM, visualization, ...)
  - Subgroup discovery: find unusual subgroups that are significantly different from the majority (deviation detection w.r.t. overall class distribution)

Types of DM tasks

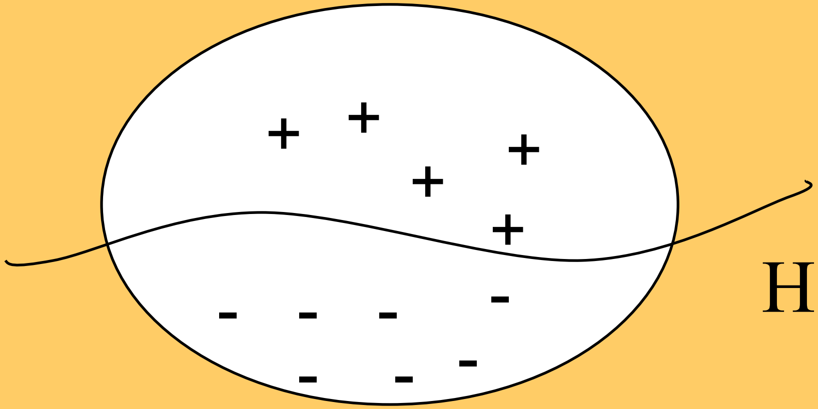
- **Predictive DM:**
  - Classification (learning of rules, decision trees, ...)
  - Prediction and estimation (regression)
  - Predictive relational DM (ILP)
- **Descriptive DM:**
  - description and summarization
  - dependency analysis (association rule learning)
  - discovery of properties and constraints
  - segmentation (clustering)
  - subgroup discovery
- **Text, Web and image analysis**



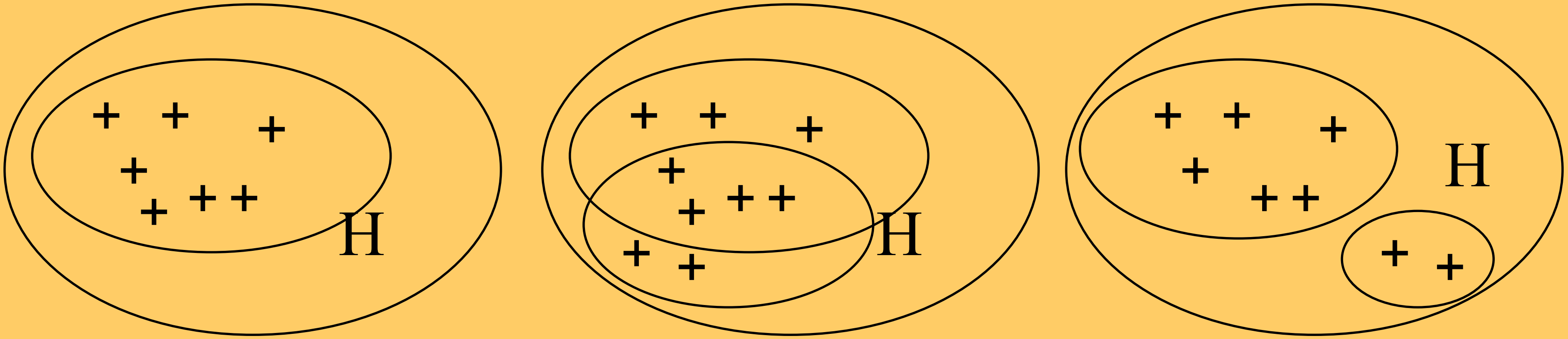


Predictive vs. descriptive induction

Predictive induction



Descriptive induction



Predictive vs. descriptive induction

- **Predictive induction:** Inducing classifiers for solving classification and prediction tasks,
  - Classification rule learning, Decision tree learning, ...
  - Bayesian classifier, ANN, SVM, ...
  - Data analysis through hypothesis generation and testing
- **Descriptive induction:** Discovering interesting regularities in the data, uncovering patterns, ... for solving KDD tasks
  - Symbolic clustering, Association rule learning, Subgroup discovery, ...
  - Exploratory data analysis

Predictive vs. descriptive induction: A rule learning perspective

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

- **Therefore:** Different goals, different heuristics, different evaluation criteria

Supervised vs. unsupervised learning: A rule learning perspective

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

  Discovers **individual rules** describing interesting regularities in the data from **labeled** examples

Part III. Descriptive DM techniques

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

Subgroup Discovery

Given: a population of individuals and a property of individuals we are interested in

Find: population subgroups that are statistically most `interesting’, e.g., are as large as possible and have most unusual statistical (distributional) characteristics w.r.t. the property of interest

Subgroup interestingness

Interestingness criteria:

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

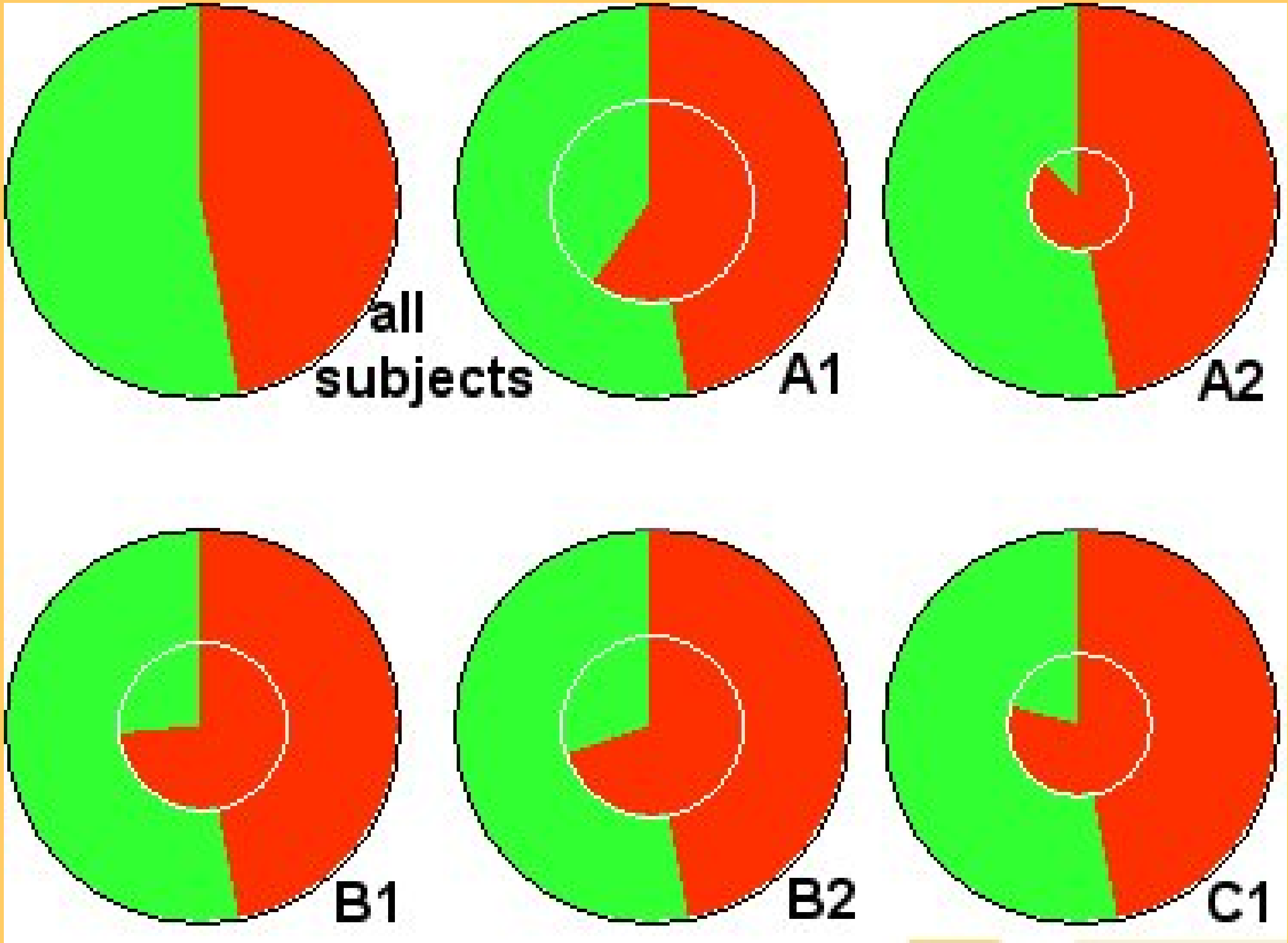
Subgroup Discovery: Medical Case Study

- **Find and characterize population subgroups with high CHD risk** (Gamberger, Lavrač, Krstačić)
- **A1** for males: **principal risk factors**

  CHD ← pos. fam. history & age > 46
- **A2** for females: **principal risk factors**

  CHD ← bodyMassIndex > 25 & age >63
- **A1, A2** (anamnestic info only), **B1, B2** (an. and physical examination), **C1** (an., phy. and ECG)
- **A1: supporting factors** (found by statistical analysis): psychosocial stress, as well as cigarette smoking, hypertension and overweight

Subgroup visualization

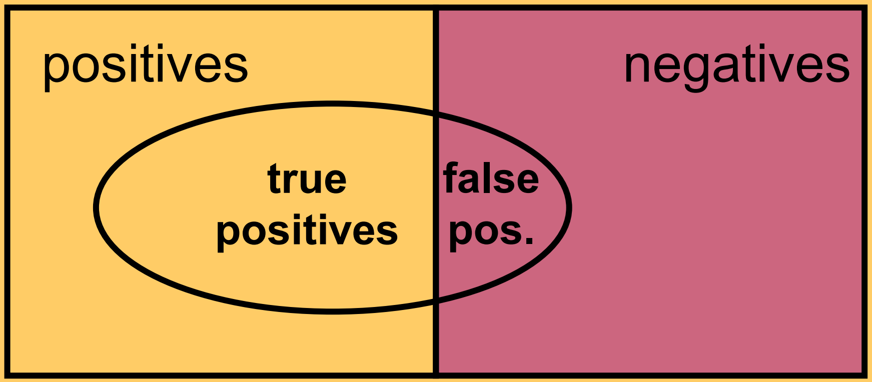


Subgroups of patients with CHD risk

[Gamberger, Lavrač & Wettschereck, IDAMAP2002]

Subgroups vs. classifiers

- Classifiers:
  - Classification rules aim at pure subgroups
  - A set of rules forms a domain model
- Subgroups:
  - Rules describing subgroups aim at significantly higher proportion of positives
  - Each rule is an independent chunk of knowledge
- Link
  - SD can be viewed as cost-sensitive classification
  - Instead of *FNcost* we aim at increased *TPprofit*



Classification Rule Learning for Subgroup Discovery: Deficiencies

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

CN2-SD: Adapting CN2 Rule Learning to Subgroup Discovery

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

CN2-SD: CN2 Adaptations

- General-to-specific search (beam search) for best rules
- Rule quality measure:
  - CN2: Laplace: Acc(Class $\leftarrow$ Cond) =

    = p(Class|Cond) = $(n_c+1)/(n_{rule}+k)$
  - CN2-SD: Weighted Relative Accuracy

    WRAcc(Class $\leftarrow$ Cond) =

    p(Cond) (p(Class|Cond) - p(Class))
- Weighted covering approach (example weights)
- Significance testing (likelihood ratio statistics)
- Output: Unordered rule sets (probabilistic classification)

CN2-SD: Weighted Covering

- Standard covering approach:
  covered examples are deleted from current training set
- Weighted covering approach:
  - weights assigned to examples
  - covered pos. examples are re-weighted:
    in all covering loop iterations, store
    count i how many times (with how many
    rules induced so far) a pos. example has
    been covered: w(e,i), w(e,0)=1
    - **Additive weights:** `w(e,i) = 1/(i+1)`
      `w(e,i)` – pos. example e being covered `i` times
    - **Multiplicative weights:** `w(e,i) =` $\texttt{gamma}^{\texttt{i}}$ `,` **`0<gamma<1`**
      `note:` **`gamma = 1`** ➔ find the same (first) rule again and again
      `gamma = 0` ➔ behaves as standard CN2

CN2-SD: Weighted WRAcc Search Heuristic

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

  WRAcc(Cl $\leftarrow$ Cond) = p(Cond) (p(Cl|Cond) - p(Cl))

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

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

  WRAcc(Cl $\leftarrow$ Cond) = p(Cond) (p(Cl|Cond) - p(Cl)) =

  n'(Cond)/N' (n'(Cl.Cond)/n'(Cond) - n'(Cl)/N')

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

Part III. Descriptive DM techniques

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

Association Rule Learning

Rules: X =>Y, if X then Y

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

Transactions:

itemsets (records)

	i1	**i2**	**.................... i50**
t1	1	1	0
t2	0	1	0
...	...	...	...

Example:

Market basket analysis

beer & coke => peanuts & chips (0.05, 0.65)

- Support: Sup(X,Y) = #XY/#D = p(XY)
- Confidence: Conf(X,Y) = #XY/#X = Sup(X,Y)/Sup(X) =

  = p(XY)/p(X) = p(Y|X)

Association Rule Learning

Given: a set of transactions D

Find: all association rules that hold on the set of transactions that have support > MinSup and confidence > MinConf

Procedure:

- find all large itemsets Z, Sup(Z) > MinSup
- split every large itemset Z into XY,
  compute Conf(X,Y) = Sup(X,Y)/Sup(X),
  if Conf(X,Y) > MinConf then X =>Y
  (Sup(X,Y) > MinSup, as XY is large)

Part III. Descriptive DM techniques

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

Hierarchical clustering

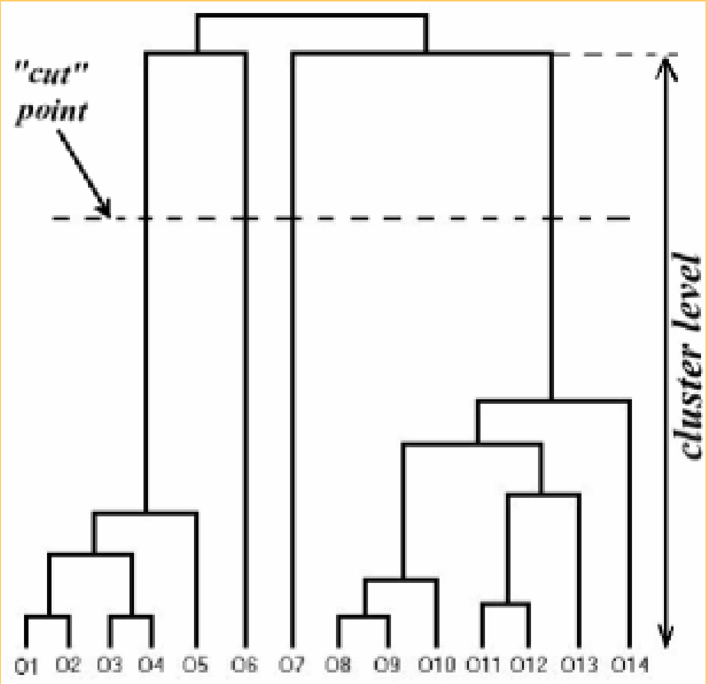
- Algorithm (agglomerative hierarchical clustering):

Each instance is a cluster;

repeat
  find *nearest* pair C_i in C_j;
  ***fuse* C_i in C_j in a new cluster**
    $C_r = C_i \cup C_j$;
  determine *dissimilarities* between
    C_r and other clusters;

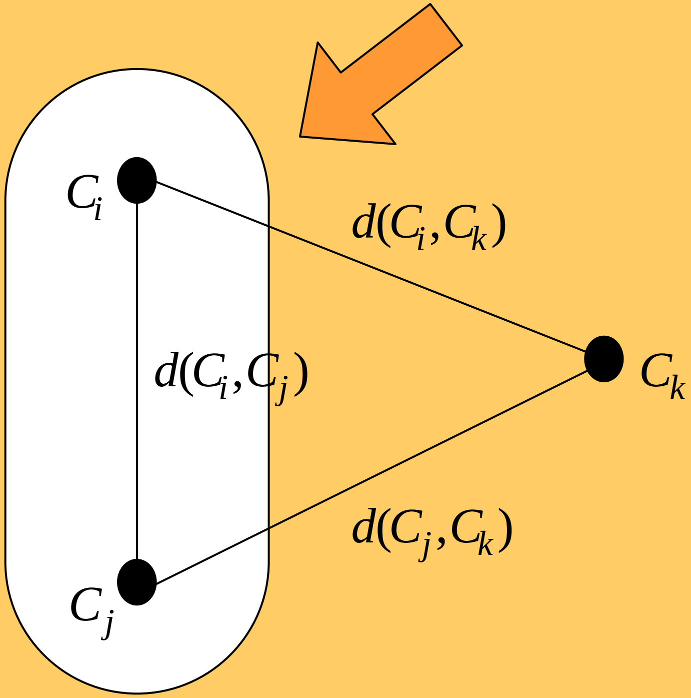
until one cluster left;

- Dendogram:



Hierarchical clustering

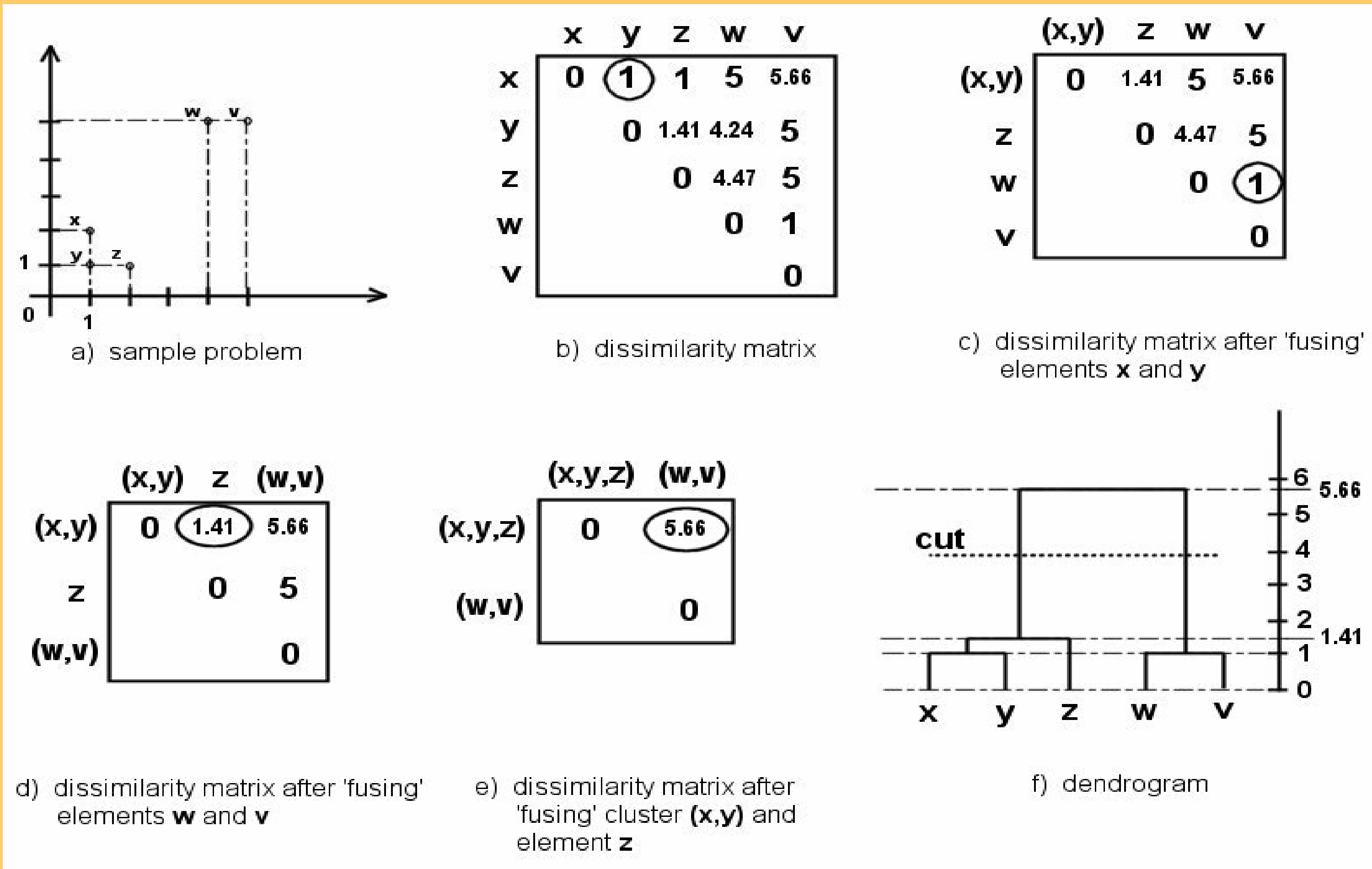
- Fusing the nearest pair of clusters



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

- Computing the dissimilarities from the "new" cluster

Hierarchical clustering: example



a) sample problem

b) dissimilarity matrix

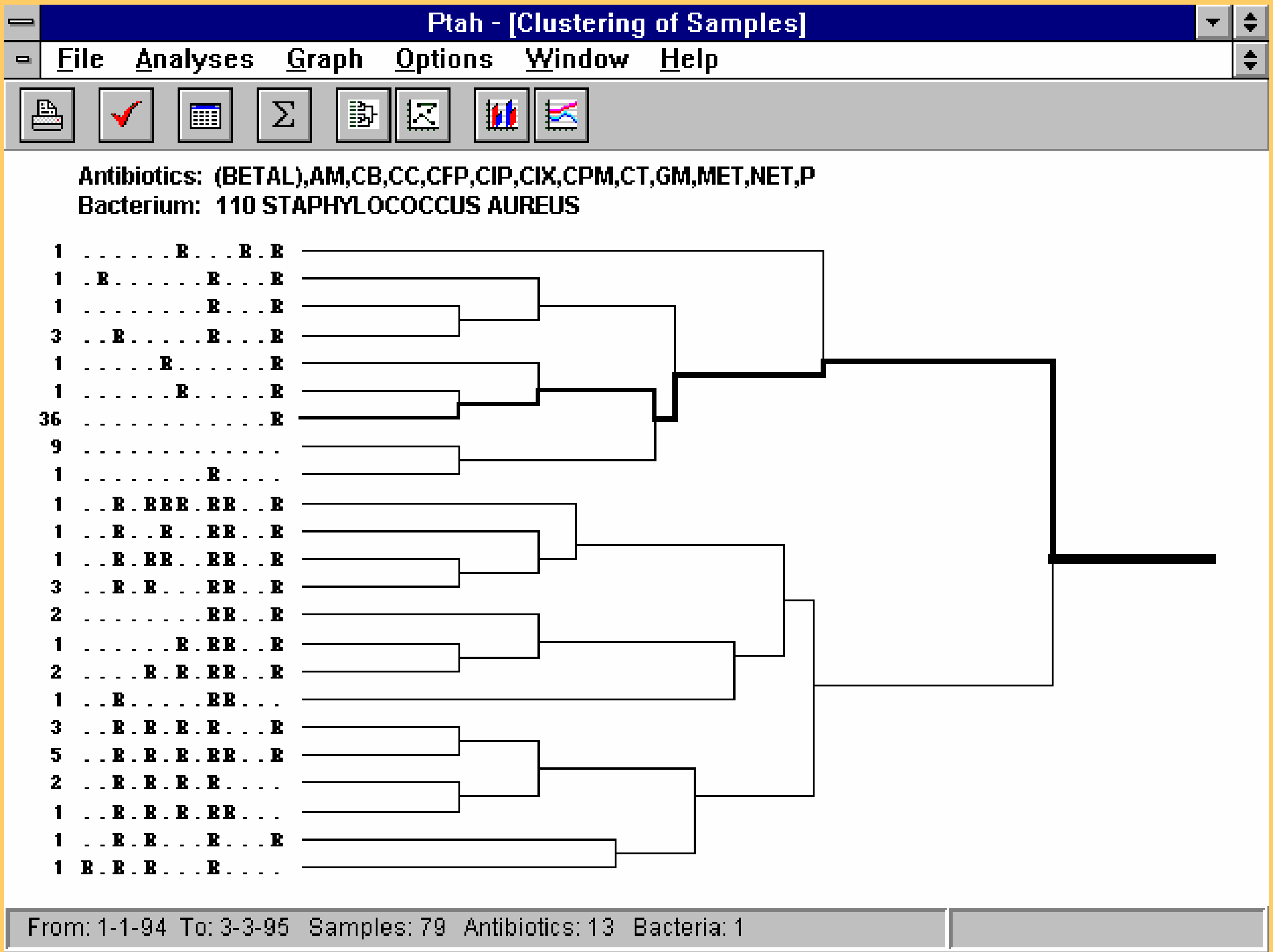
c) dissimilarity matrix after 'fusing' elements **x** and **y**

d) dissimilarity matrix after 'fusing' elements **w** and **v**

e) dissimilarity matrix after 'fusing' cluster **(x,y)** and element **z**

f) dendrogram

Results of clustering



A dendogram of resistance vectors

[Bohanec et al., “PTAH: A system for supporting nosocomial infection therapy”, IDAMAP book, 1997]