Data and Text Mining

Part of
Jožef Stefan IPS Programme – ICT2

2020 / 2021

Nada Lavrač

Jožef Stefan Institute

Ljubljana, Slovenia

2020/2021 Logistics: Course lecturers

Contacts: http://kt.ijs.si/petra_kralj/dmtm2.html

- Data Mining:
  - Nada Lavrač: nada.lavrac@ijs.si,
  - Petra Kralj Novak: petra.kralj.novak@ijs.si,
  - Martin Žnidaršič: martin.znidarsic@ijs.si
- Data prepraration:
  - Bojan Cestnik: bojan.cestnik@temida.si
- Text mining
  - Dunja Mladenić: dunja.mladenic@ijs.si

Course Schedule – 2020/21

ICT2 – see http://kt.ijs.si/petra_kralj/dmtm2.html

4.11.2020	15:00 - 18:00	prof. dr. Nada Lavrač
11.11.2020	15:00 - 18:00	doc. dr. Petra Kralj Novak
18.11.2020	15:00 - 18:00	prof. dr. Nada Lavrač
25.11.2020	15:00 - 18:00	doc. dr. Petra Kralj Novak
2.12.2020	15:00 - 18:00	prof. dr. Bojan Cestnik
9.12.2020	15:00 - 18:00	prof. dr. Dunja Mladenić
10.12.2020	15:00 - 18:00	doc. dr. Petra Kralj Novak - Oral partial exam on Data mining
16.12.2020	15:00 - 18:00	Erik Novak
23.12.2020	15:00 - 18:00	prof. dr. Bojan Cestnik
6.1.2021	15:00 - 18:00	prof. dr. Dunja Mladenić
13.1.2021	15:00 - 18:00	prof. dr. Dunja Mladenić
20.1.2021	15:00 - 18:00	prof. dr. Nada Lavrač - Data mining seminar presentations (partial exam)
3.2.2021	15:00 - 17:00	Erik Novak

Data and Text Mining: ICT2 Credits, Supporting material

- 20 credits
  - 8 credits Nada Lavrač and Petra Kralj Novak
  - 4 credits Bojan Cestnik
  - 8 credits Dunja Mladenić

- Supporting material on videolectures.net:
  Seminar: AI for Industry and Society, Ljubljana 2020
  - http://videolectures.net/AIindustrySeminar2019/
  - Marko Robnik Šikonja: Artificial Intelligence: Techniques, Trends and Applications
  - Nada Lavrač: Data Science, Machine Learning and Big Data: Current trends
  - Blaž Zupan: Data Science with the OrangeToolbox
  - Dunja Mladenić: Text Mining Applications for Industry

Data Mining: MSc Credits and Coursework for Data mining part

Requirements for Data Mining part by Nada Lavrač and Petra Kralj Novak (8 ECTS credits):

- Attending lectures
- Attending practical exercises
  - Theory exercises and hands-on (intro to Orange DM toolbox by dr. Petra Kralj Novak)
- Oral exam (40%)
- Seminar (60%):
  - Data analysis of your own data (e.g., using Orange for questionnaire data analysis)
  - …. own initiatives are welcome …

Data Mining: MSc Credits and coursework

Exam: Oral exam - Theory

Seminar: topic selection + results presentation

- One hour available for seminar topic discussion – one page written proposal defining the task and the selected dataset
- Deliver written report + electronic copy (4 pages in Information Society paper format, instructions on the web)
  - Report on data analysis of own data needs to follow the CRISP-DM methodology
  - Presentation of your seminar results (15 minutes each: 10 minutes presentation + 5 minutes discussion)

Course Outline

I. Introduction

- Data Mining and KDD process
- Introduction to Data Mining
- Data Mining platforms

II. Predictive DM

- Decision Tree learning
- Bayesian classifier
- Classification rule learning
- Classifier evaluation

III. Predictive DM

- Regression

IV. Descriptive DM

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

Part I. Introduction

- Data Mining and the KDD process
- Introduction to Data Mining
- Data Mining platforms

Machine Learning and Data Mining

- Machine Learning (ML) – computer algorithms/machines that learn predictive models from class-labeled data
- Data Mining (DM) – extraction of useful information from data: discovering relationships and patterns that have not previously been known, and use of ML techniques applied to solving real-life data analysis problems
- Knowledge discovery in databases (KDD) – the process of knowledge discovery

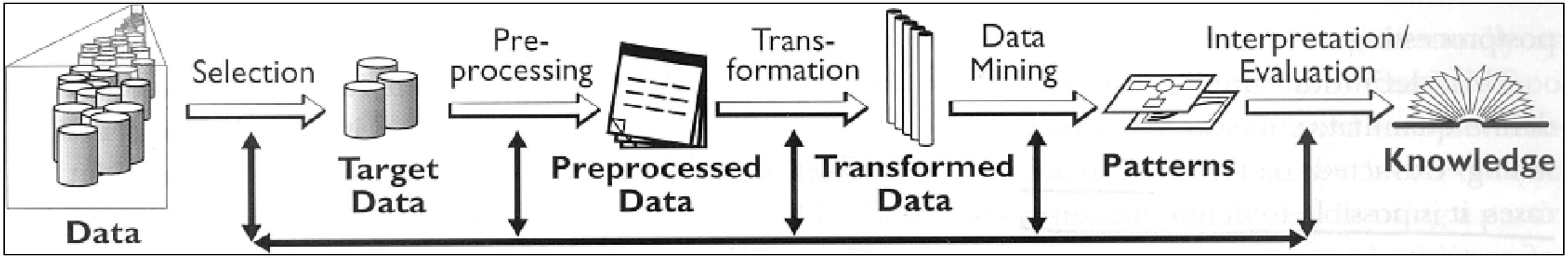
Data Mining and KDD

- Buzzword since 1996
- 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: CRISP-DM

KDD process of discovering useful knowledge from data

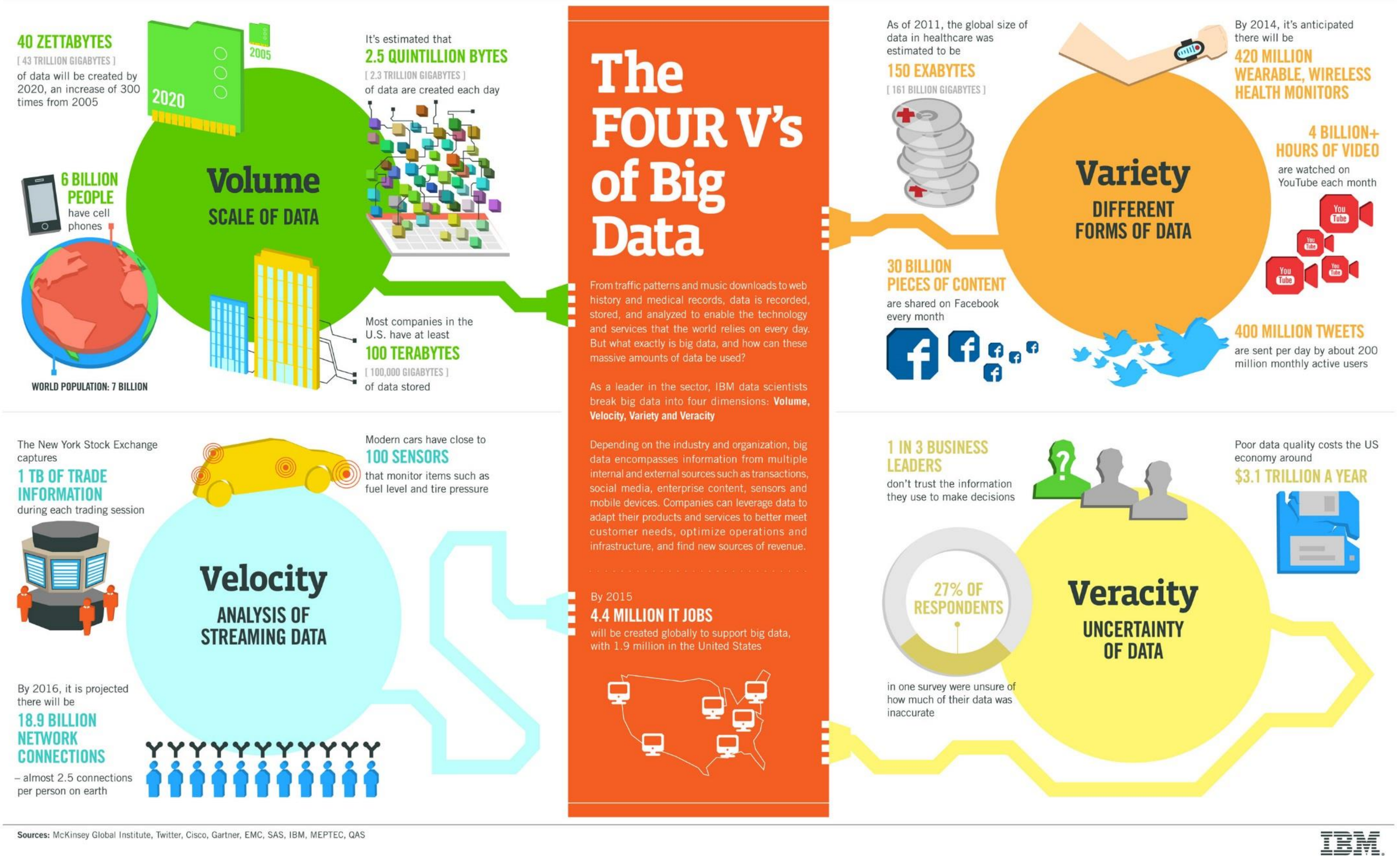


- KDD process involves several phases:
  - data preparation
  - data mining (machine learning, statistics)
  - evaluation and use of discovered patterns
- Data mining is the key step, but represents only 15%-25% of the entire KDD process

Big Data

- Big Data – Buzzword since 2008 (special issue of Nature on Big Data)
  - data and techniques for dealing with very large volumes of data, possibly dynamic data streams
  - requiring large data storage resources, special algorithms for parallel computing architectures.

The 4 Vs of Big Data



Data Science

- Data Science – buzzword since 2012 when Harvard Business Review called it "The Sexiest Job of the 21st Century"
  - – an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, similar to data mining.
  - – used interchangeably with earlier concepts like business analytics, business intelligence, predictive modeling, and statistics.

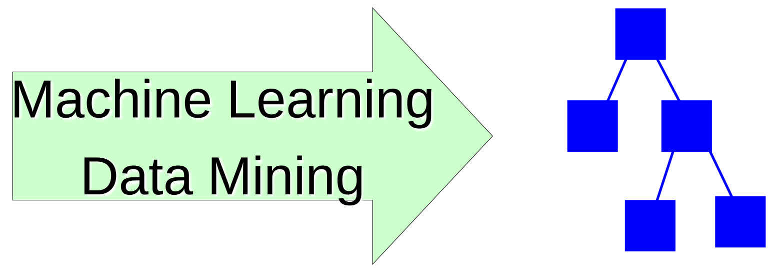
Machine Learning and Data Mining

data

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	17	myope	no	reduced	NONE
O2	23	myope	no	normal	SOFT
O3	22	myope	yes	reduced	NONE
O4	27	myope	yes	normal	HARD
O5	19	hypermetrope	no	reduced	NONE
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	SOFT
O15	43	hypermetrope	yes	reduced	NONE
O16	39	hypermetrope	yes	normal	NONE
O17	54	myope	no	reduced	NONE
O18	62	myope	no	normal	NONE
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	NONE

data

knowledge discovery
from data



model, patterns, …

Given: class labeled data

Find: a classification model, a set of interesting patterns

Machine Learning and Data Mining

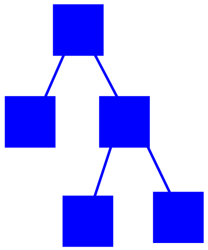
data

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	17	myope	no	reduced	NONE
O2	23	myope	no	normal	SOFT
O3	22	myope	yes	reduced	NONE
O4	27	myope	yes	normal	HARD
O5	19	hypermetrope	no	reduced	NONE
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	SOFT
O15	43	hypermetrope	yes	reduced	NONE
O16	39	hypermetrope	yes	normal	NONE
O17	54	myope	no	reduced	NONE
O18	62	myope	no	normal	NONE
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	NONE

data

knowledge discovery from data

Machine Learning
Data Mining

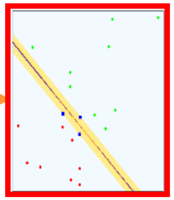


model, patterns, …

Given: class labeled data

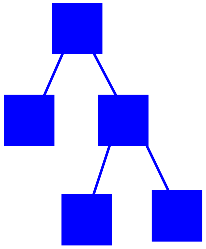
Find: a classification model, a set of interesting patterns

new unclassified instance



classified instance

black box classifier
no explanation



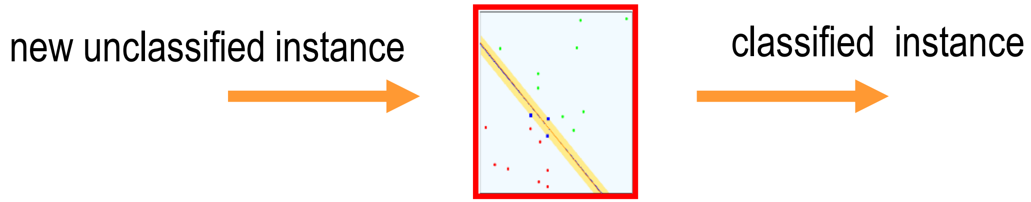
symbolic model
symbolic patterns

explanation

Why learn and use black-box models

Given: the learned classification model
(e.g, a linear classifier, a deep neural network, …)

Find: - the class label for a new unlabeled instance



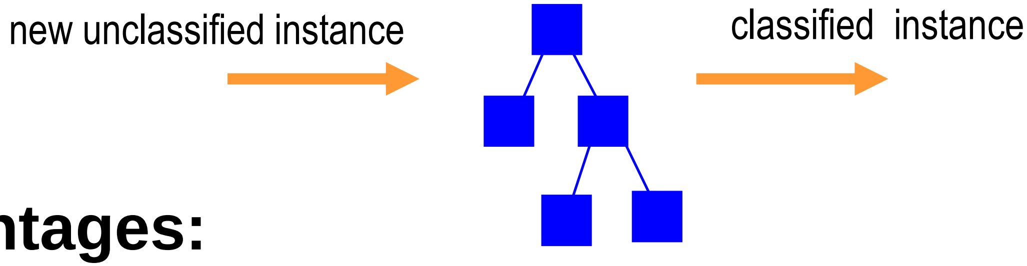
Advantages:
- best classification results in image recognition and other complex classification tasks

Drawbacks:
- poor interpretability of results
- can not be used for pattern analysis

Why learn and use symbolic models

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

Find: - the class label for a new unlabeled instance



Advantages:

- use the model for the explanation of classifications of new data instances
- use the discovered patterns for data exploration

Drawbacks:

- lower accuracy than deep NNs

Simplified example: Learning a classification model from contact lens data

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	17	myope	no	reduced	NONE
O2	23	myope	no	normal	SOFT
O3	22	myope	yes	reduced	NONE
O4	27	myope	yes	normal	HARD
O5	19	hypermetrope	no	reduced	NONE
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	SOFT
O15	43	hypermetrope	yes	reduced	NONE
O16	39	hypermetrope	yes	normal	NONE
O17	54	myope	no	reduced	NONE
O18	62	myope	no	normal	NONE
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	NONE

Pattern discovery in Contact lens data

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	17	myope	no	reduced	NONE
O2	23	myope	no	normal	SOFT
O3	22	myope	yes	reduced	NONE
O4	27	myope	yes	normal	HARD
O5	19	hypermetrope	no	reduced	NONE
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	SOFT
O15	43	hypermetrope	yes	reduced	NONE
O16	39	hypermetrope	yes	normal	NONE
O17	54	myope	no	reduced	NONE
O18	62	myope	no	normal	NONE
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	NONE

PATTERN

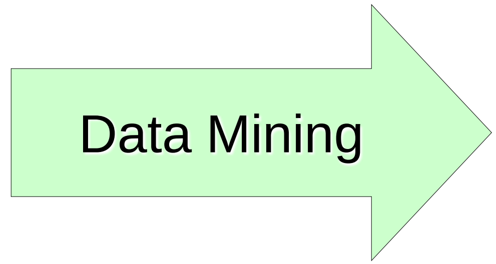
Rule:

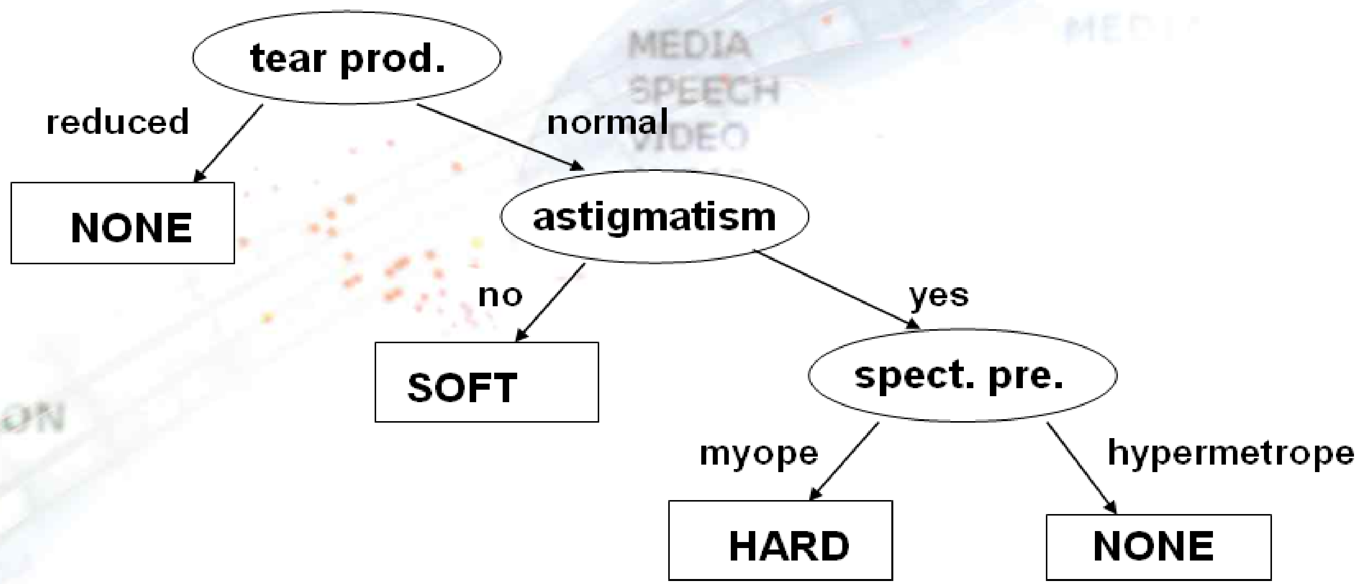
IF
Tear prod. =
reduced

THEN
Lenses =
NONE

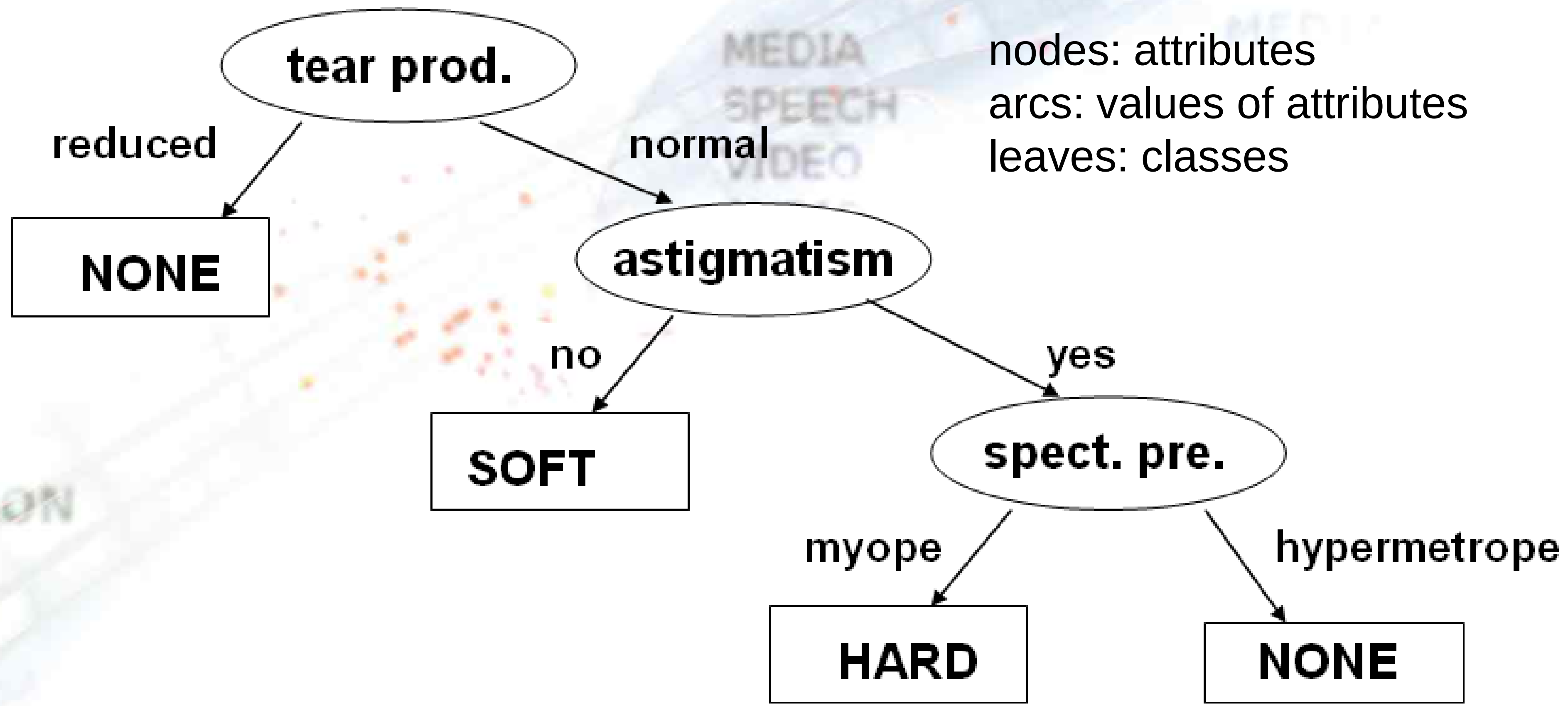
Learning a classification model from contact lens data

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



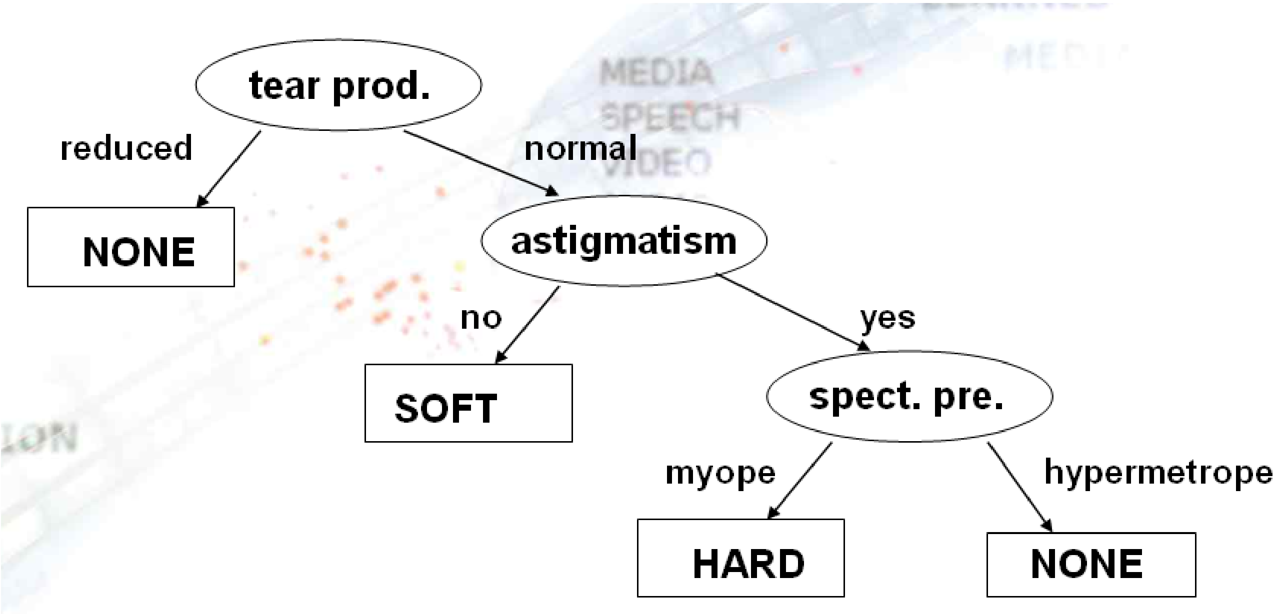


Decision tree classification model learned from contact lens data



nodes: attributes
arcs: values of attributes
leaves: classes

Learning a decision tree classification model



Using Gain(S,A) heuristic for determining the most informative attribute

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

Gain(S,A) estimates the reduction of entropy of set S after splitting into subsets based on values of attribute A

Heuristics for estimating the informativity of attributes and features

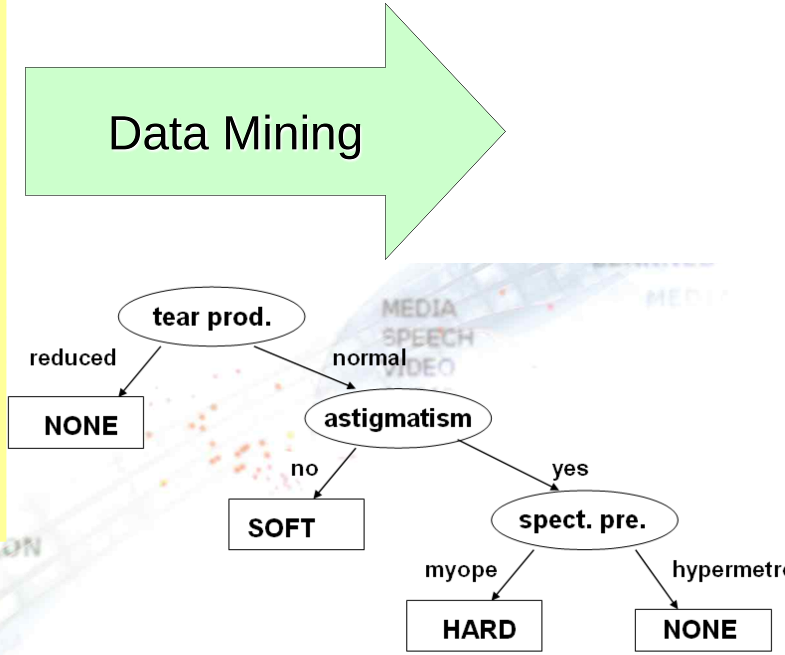
- **Search heuristics:** Which attribute to test at each node in the tree ? The attribute that is most useful for classifying examples.
- Define a statistical property, called **information gain**, measuring how well a given attribute separates the training examples w.r.t their target classification.
- First define a measure commonly used in information theory, called **entropy**, to characterize the (im)purity of an arbitrary collection of examples, and **Informativity of an attribute** merimois measured as **reduction of entropy of a training set**
- **Entropy:** $E(S) = - p_+ \log_2 p_+ - p_- \log_2 p_-$
- **Most informative attribute**:
  - Select S
  - Select A to split S into $S_1, S_2, \ldots, S_v$
  - Select A, which maximizes info. Gain

    max Gain(S,A)

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

Learning a classification model from contact lens data

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	17	myope	no	reduced	NONE
O2	23	myope	no	normal	SOFT
O3	22	myope	yes	reduced	NONE
O4	27	myope	yes	normal	HARD
O5	19	hypermetrope	no	reduced	NONE
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	SOFT
O15	43	hypermetrope	yes	reduced	NONE
O16	39	hypermetrope	yes	normal	NONE
O17	54	myope	no	reduced	NONE
O18	62	myope	no	normal	NONE
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	NONE



lenses=NONE ← tear production=red

lenses=NONE ← tear production=normal AND astigmatism=yes AND spect. pre.=hypermetrope

lenses=SOFT ← tear production=normal AND astigmatism=no

lenses=HARD ← tear production=normal AND astigmatism=yes AND spect. pre.=myope

lenses=NONE ←

Classification rules model learned from contact lens data

lenses=NONE ← tear production=reduced
lenses=NONE ← tear production=normal AND
astigmatism=yes AND
spect. pre.=hypermetrope
lenses=SOFT ← tear production=normal AND
astigmatism=no
lenses=HARD ← tear production=normal AND
astigmatism=yes AND
spect. pre.=myope
lenses=NONE ←

Learning from Unlabeled Data

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	17	myope	no	reduced	NONE
O2	23	myope	no	normal	SOFT
O3	22	myope	yes	reduced	NONE
O4	27	myope	yes	normal	HARD
O5	19	hypermetrope	no	reduced	NONE
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	SOFT
O15	43	hypermetrope	yes	reduced	NONE
O16	39	hypermetrope	yes	normal	NONE
O17	54	myope	no	reduced	NONE
O18	62	myope	no	normal	NONE
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	NONE

Unlabeled data - clustering: grouping of similar instances
- association rule learning

Learning from Numeric Class Data

Person	Age	Spect. presc.	Astigm.	Tear prod.	LensPrice
O1	17	myope	no	reduced	0
O2	23	myope	no	normal	8
O3	22	myope	yes	reduced	0
O4	27	myope	yes	normal	5
O5	19	hypermetrope	no	reduced	0
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	5
O15	43	hypermetrope	yes	reduced	0
O16	39	hypermetrope	yes	normal	0
O17	54	myope	no	reduced	0
O18	62	myope	no	normal	0
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	0

Numeric class values – regression analysis

Task reformulation: Binary Class Values

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	17	myope	no	reduced	NO
O2	23	myope	no	normal	YES
O3	22	myope	yes	reduced	NO
O4	27	myope	yes	normal	YES
O5	19	hypermetrope	no	reduced	NO
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	YES
O15	43	hypermetrope	yes	reduced	NO
O16	39	hypermetrope	yes	normal	NO
O17	54	myope	no	reduced	NO
O18	62	myope	no	normal	NO
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	NO

Binary classes (positive vs. negative examples of Target class)

- for Concept learning – classification and class description
  - for Subgroup discovery – exploring patterns characterizing groups of instances of target class

Task reformulation: Binary Class and Feature Values

Person	Young	Myope	Astigm.	Reuced tea	Lenses
O1	1	1	0	1	NO
O2	1	1	0	0	YES
O3	1	1	1	1	NO
O4	1	1	1	0	YES
O5	1	0	0	1	NO
O6-O13	...	...	...	...	...
O14	0	0	0	0	YES
O15	0	0	1	1	NO
O16	0	0	1	0	NO
O17	0	1	0	1	NO
O18	0	1	0	0	NO
O19-O23	...	...	...	...	...
O24	0	0	1	0	NO

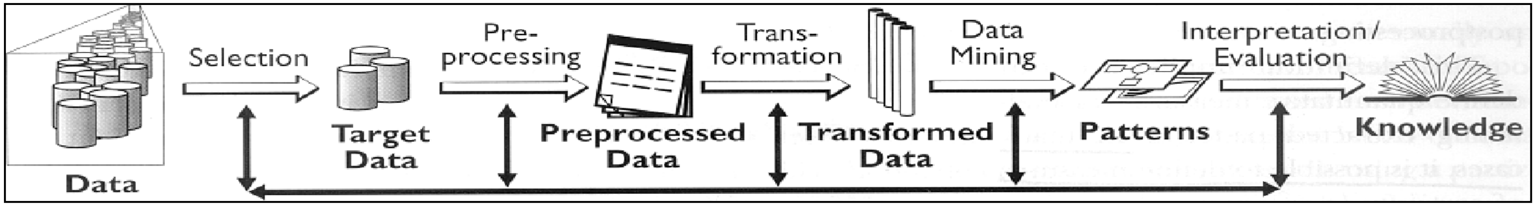
Binary features and class values

First Generation Data Mining

- **First machine learning algorithms for**
  - Decision tree and rule learning in 1970s and early 1980s by Quinlan, Michalski et al., Breiman et al., …
- **Characterized by**
  - Learning from data stored in a single data table
  - Relatively small set of instances and attributes
- **Lots of ML research followed in 1980s**
  - Numerous conferences ICML, ECML, … and ML sessions at AI conferences IJCAI, ECAI, AAAI, …
  - Extended set of learning tasks and algorithms addressed

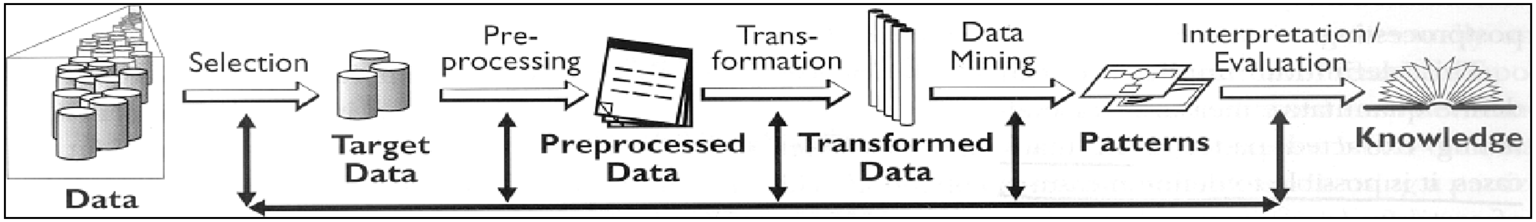
Second Generation Data Mining

- **Developed since 1990s:**
  - Focused on data mining tasks characterized by large datasets described by large numbers of attributes
  - Industrial standard: CRISP-DM methodology (1997)



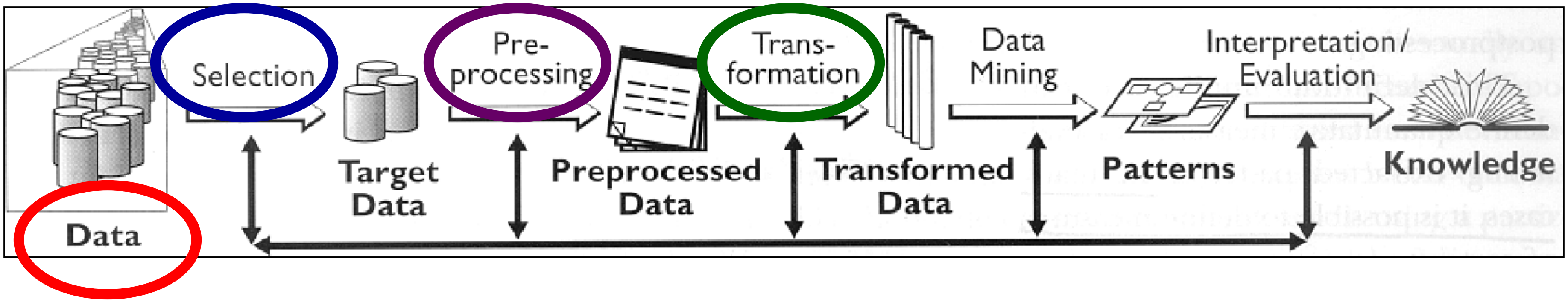
Second Generation Data Mining

- **Developed since 1990s:**
  – Focused on data mining tasks characterized by large datasets described by large numbers of attributes
  – Industrial standard: CRISP-DM methodology (1997)



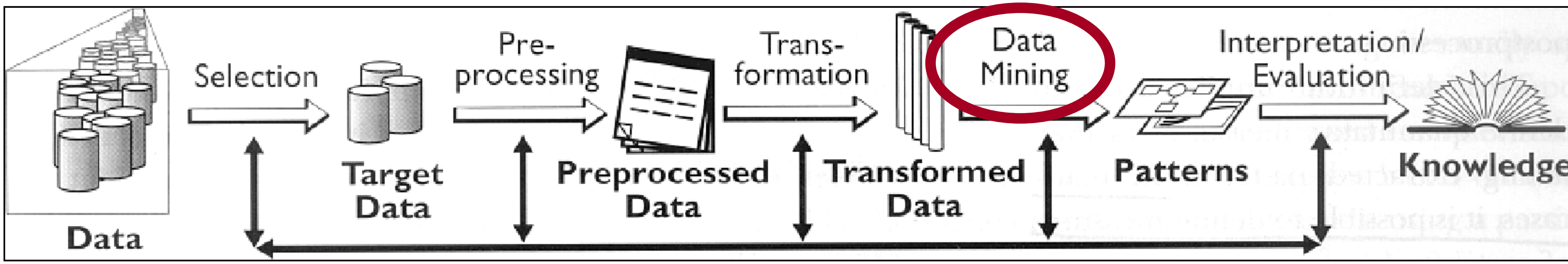
  – New conferences on practical aspects of data mining and knowledge discovery: KDD, PKDD, …
  – New learning tasks and efficient learning algorithms:
    - Learning predictive models: Bayesian network learning,, relational data mining, statistical relational learning, SVMs, …
    - Learning descriptive patterns: association rule learning, subgroup discovery, …

MEDIANA – analysis of media research data



- Questionnaires about journal/magazine reading, watching of TV programs and listening of radio programs, about 1200 questions. Yearly publication: frequency of reading/listening/watching, distribution w.r.t. Sex, Age, Education, Buying power,..
- Data 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)

Simplified association rules

Finding profiles of readers of the Delo daily newspaper

1. reads_Marketing_magazine 116 ➔ reads_Delo 95 (0.82)
2. reads_Finance 223 ➔ reads_Delo 180 (0.81)
3. reads_Views 201 ➔ reads_Delo 157 (0.78)
4. reads_Money 197 ➔ reads_Delo 150 (0.76)
5. reads_Vip 181 ➔ reads_Delo 134 (0.74)

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

Simplified association rules

1. reads_Sara 332 ➔ reads_Slovenian_news 211 (0.64)
2. reads_Love_stories 283 ➔
   reads_Slovenian_news 174 (0.61)
3. reads_Dolenjska_news 520 ➔
   reads_Slovenian_news 310 (0.6)
4. reads_Omama 154 ➔ reads_Slovenian_news 90 (0.58)
5. reads_Workers_news 177 ➔
   reads_Slovenian_news 102 (0.58)

Most of the readers of Sara, Love stories, Dolenjska news, Omama in Workers news read also Slovenian news.

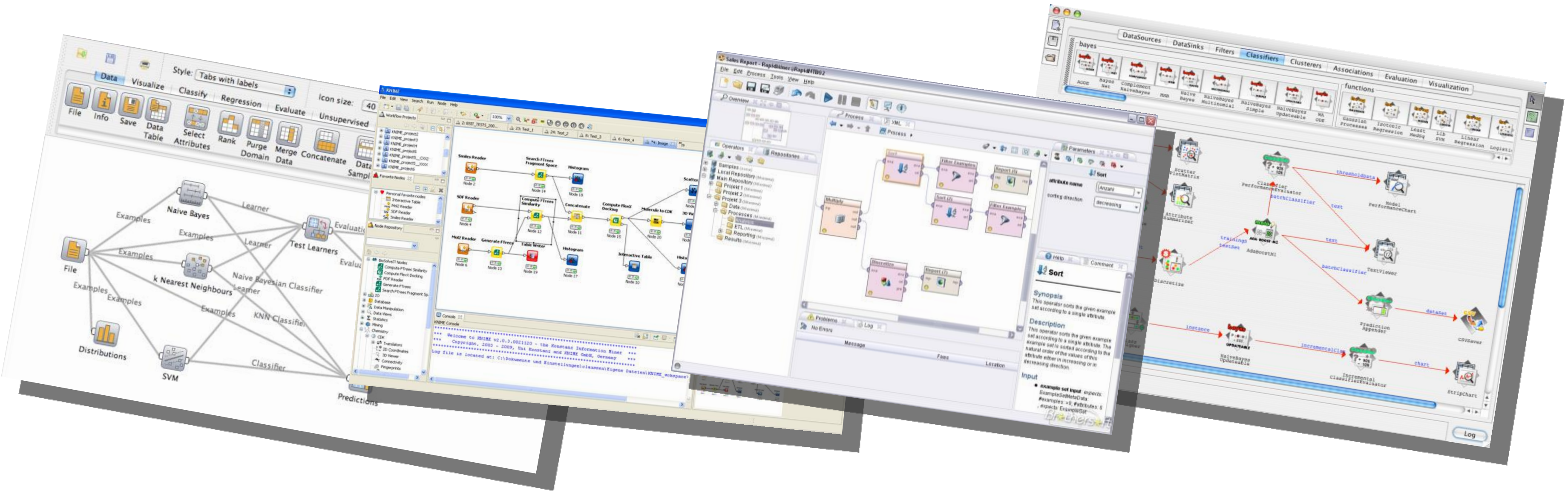
Simplified association rules

1. reads_Sports_news 303 ➔
   reads_Slovenian_shareholders_magazine 164 (0.54)
2. reads_Sports_news 303 ➔
   reads_Salomon_advertisemens 155 (0.51)
3. reads_Sports_news 303 ➔
   reads_Lady 152 (0.5)

More than half of readers of Sports news reads also Slovenian shareholders magazine, Solomon advertisements and Lady.

Second Generation Data Mining Platforms

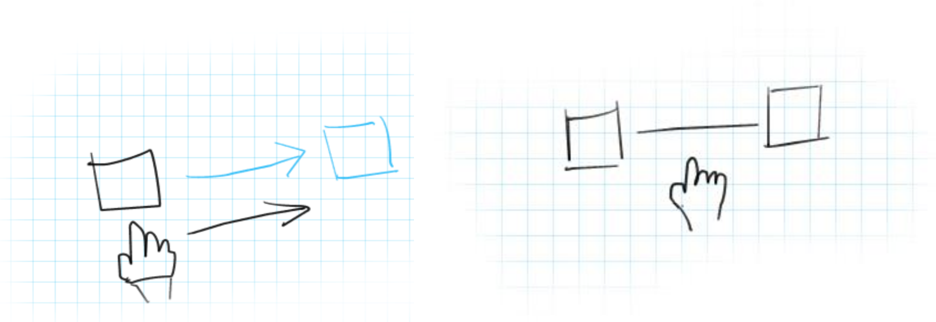
Orange, WEKA, KNIME, RapidMiner, …



– include numerous data mining algorithms
– enable data and model visualization
– like Orange, Taverna, WEKA, KNIME, RapidMiner, also enable complex **workflow** construction

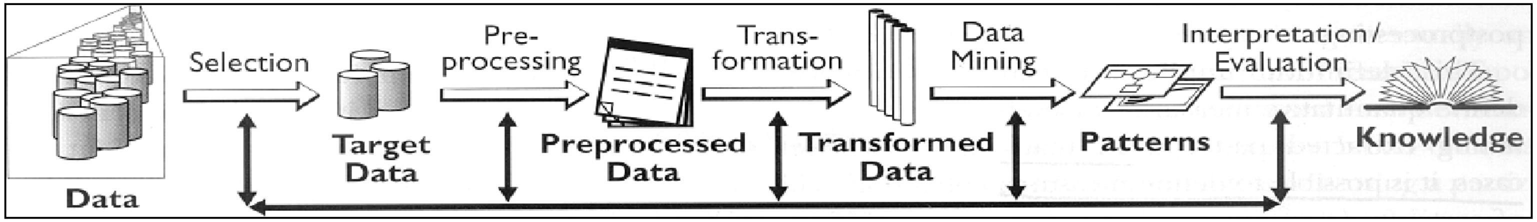
Data Mining Workflows for Open Data Science

- Workflows are executable visual representations of procedures
  - divided into smaller chunks of code (components)
  - organized as sequences of connected components.
- Suitable for representing complex scientific pipelines
  - by explicitly modeling dependencies of components
- Building scientific workflows consists of simple operations on workflow elements (drag, drop, connect), suitable for non-experts



Third Generation Data Mining

- **Developed since 2010s:**
  - Focused on big data analytics
  - Addressing complex data mining tasks and scenarios
  - New conferences on data science and big data analytics; e.g., IEEE Big Data, Complex networks, …
  - New learning tasks and efficient learning algorithms:
    - Analysis of dynamic data streams, Network analysis, Text mining, Semantic data analysis, …
  - Lots of emphasis on automated **data transformation**
    - Propositionalization of relational data, of heterogeneous information networks, …
    - Embedding of texts, networks, knowledge graphs, entities (features), … is highly popular in the last few years



Propositionalization: Data transformation for Relational Data Mining

customer							
ID	Zip	Sex	So St	In come	Age	Club	Re sp
...	...	...	...	...	...	...	...
3478	34677	m	si	60-70	32	me	nr
3479	43666	f	ma	80-90	45	nm	re
...	...	...	...	...	...	...	...

order				
Customer ID	Order ID	Store ID	Delivery Mode	Paymt Mode
...	...	...	...	...
3478	2140267	12	regular	cash
3478	3446778	12	express	check
3478	4728386	17	regular	check
3479	3233444	17	express	credit
3479	3475886	12	regular	credit
...	...	...	...	...

store			
Store ID	Size	Type	Location
...	...	...	...
12	small	franchise	city
17	large	indep	rural
...	...	...	...

Relational representation of customers, orders and stores.

Step 1

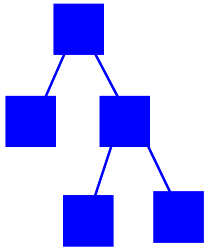
Propositionalization

	f1	f2	f3	f4	f5	f6	...				...	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

	f1	f2	f3	f4	f5	f6	...				...	fn
g1	1	0	0	1	1	1	0	0	1	0	1	1
g2	0	1	1	0	1	1	0	0	0	1	1	0
g3	0	1	1	1	0	0	1	1	0	0	0	1
g4	1	1	1	0	1	1	0	0	1	1	1	0
g5	1	1	1	0	0	1	0	1	1	0	1	0
g1	0	0	1	1	0	0	0	1	0	0	0	1
g2	1	1	0	0	1	1	0	1	0	1	1	1
g3	0	0	0	0	1	0	0	1	1	1	0	0
g4	1	0	1	1	1	0	1	0	0	1	0	1

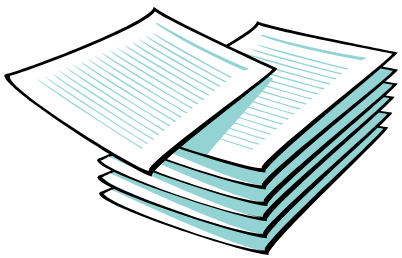
Step 2

Data Mining



model, patterns, …

Bag-of-Words Data Transformation for Text mining



Step 1

BoW vector construction

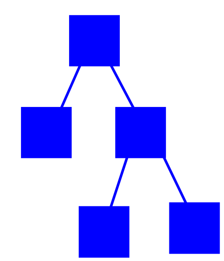
1. BoW features construction
2. Table of BoW vectors construction

Document	Word1	Word2	…	WordN	Class
d1	1	1	0	1	NO
d2	1	1	0	0	YES
d3	1	1	1	1	NO
d4	1	1	1	0	YES
d5	1	0	0	1	NO
d6-d13	…	…	…	…	…
d14	0	0	0	0	YES
d15	0	0	1	1	NO
d16	0	0	1	0	NO
d17	0	1	0	1	NO
d18	0	1	0	0	NO
d19-d23	…	…	…	…	…
d24	0	0	1	0	NO

Document	Word1	Word2	…	WordN	Class
d1	1	1	0	1	NO
d2	1	1	0	0	YES
d3	1	1	1	1	NO
d4	1	1	1	0	YES
d5	1	0	0	1	NO
d6-d13	…	…	…	…	…
d14	0	0	0	0	YES
d15	0	0	1	1	NO
d16	0	0	1	0	NO
d17	0	1	0	1	NO
d18	0	1	0	0	NO
d19-d23	…	…	…	…	…
d24	0	0	1	0	NO

Step 2

Data Mining



model, patterns, clusters, …

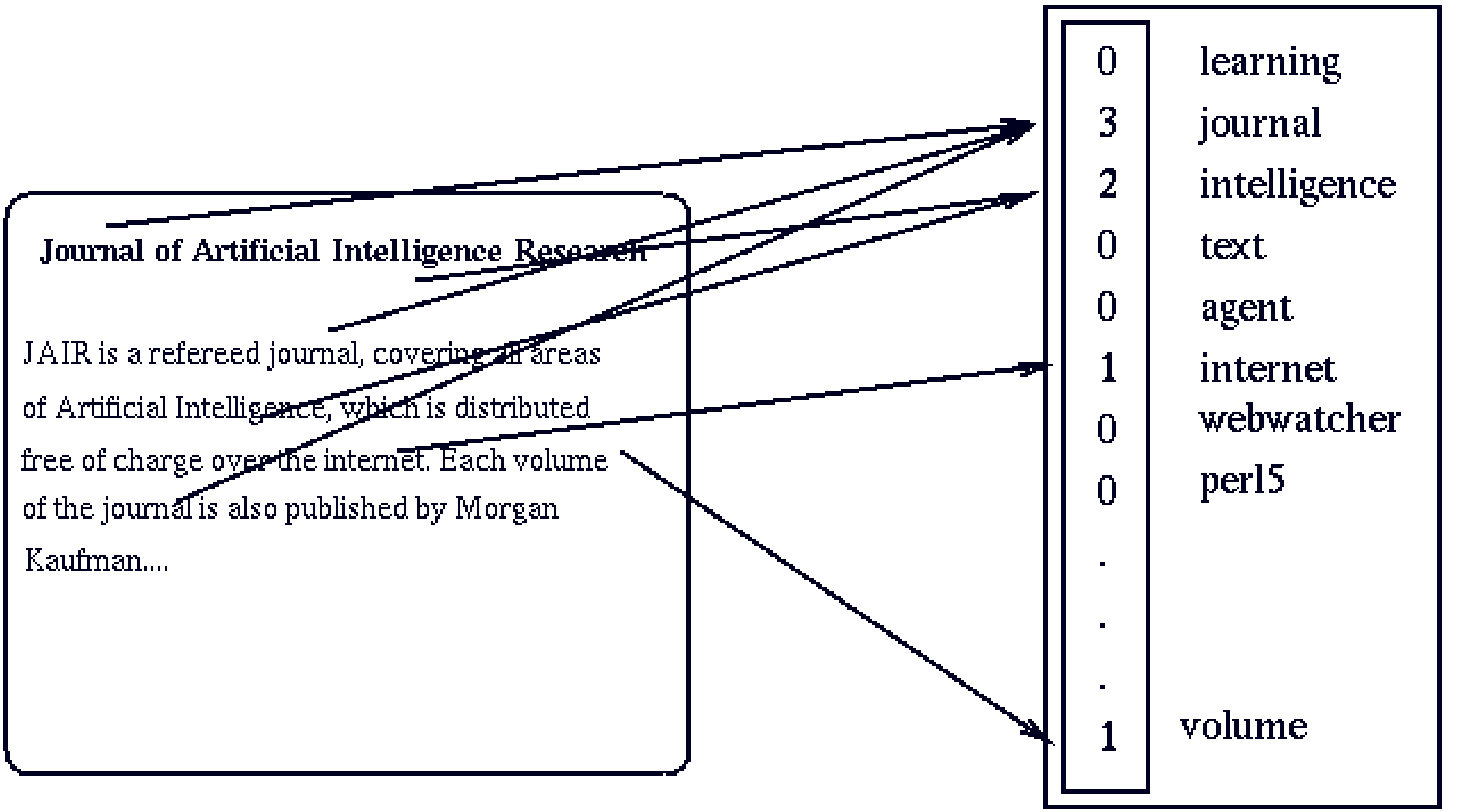
Text mining:
Words/terms as binary features

Document	Word1	Word2	…	WordN	Class
d1	1	1	0	1	NO
d2	1	1	0	0	YES
d3	1	1	1	1	NO
d4	1	1	1	0	YES
d5	1	0	0	1	NO
d6-d13	...	...	...	...	...
d14	0	0	0	0	YES
d15	0	0	1	1	NO
d16	0	0	1	0	NO
d17	0	1	0	1	NO
d18	0	1	0	0	NO
d19-d23	...	...	...	...	...
d24	0	0	1	0	NO

Instances = documents

Words and terms = Binary features

Bag-of-Words document representation



Word weighting for BoW document representation

- In bag-of-words representation each word is represented as a separate variable having numeric weight.
- The most popular weighting schema is normalized word frequency TFIDF:

$$tfidf(w) = tf \,.\, \log\left(\frac{N}{df(w)}\right)$$

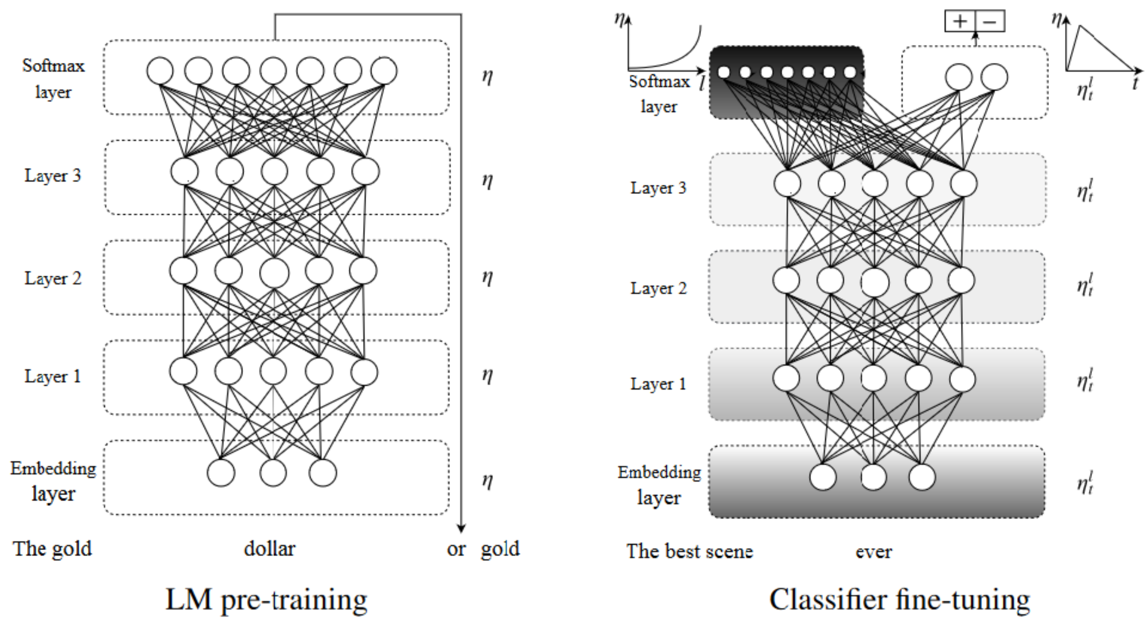
  - Tf(w) – term frequency (number of word occurrences in a document)
  - Df(w) – document frequency (number of documents containing the word)
  - N – number of all documents
  - Tfidf(w) – relative importance of the word in the document

The word is more important if it appears several times in a target document

The word is more important if it appears in less documents

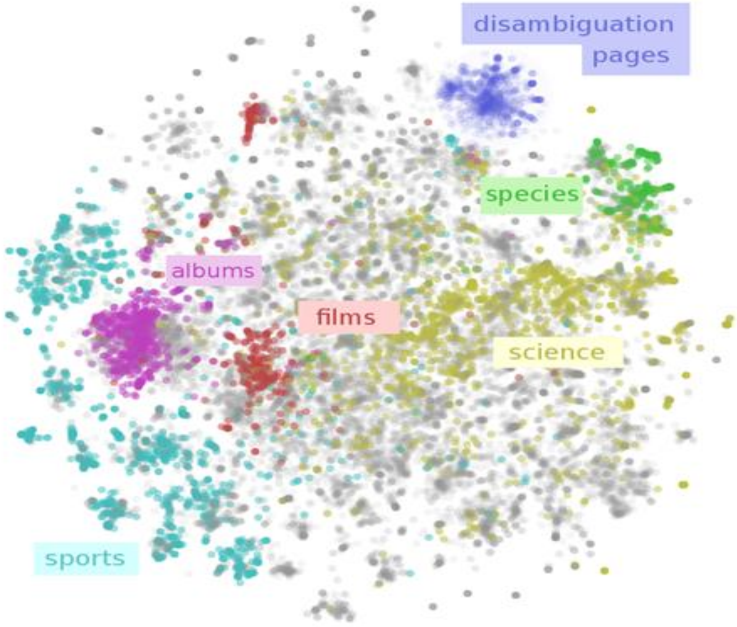
Embeddings-based Data Transformations

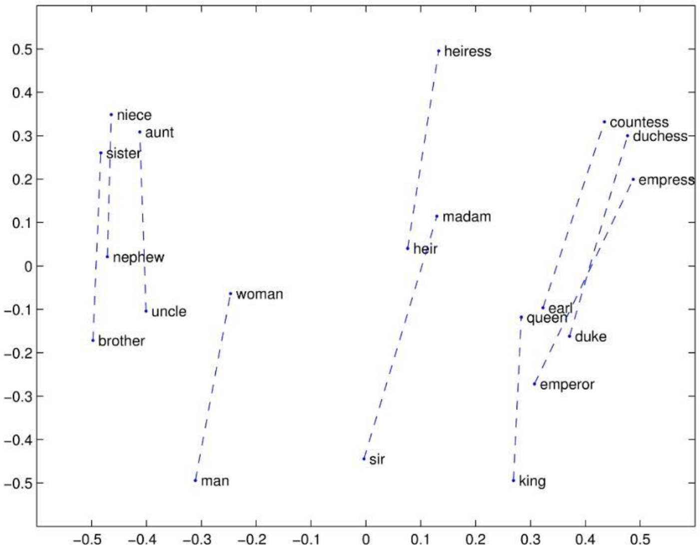
- Embedding networks, knowledge graphs, relational data, entities (features), texts …
  - Transforming data by projecting individual data instances into vectors (rows of a data table) – **dense data representation**
  - Weights correspond to weights in the embedding layer of a neural network



Embedding-based Data Transformation for Text mining

- Corpus embedding, Document embedding, Sentence embedding, **word embedding**, …
  - Representations of word meaning obtained from corpus statistics
  - Spatial relationships correspond to linguistic relationships



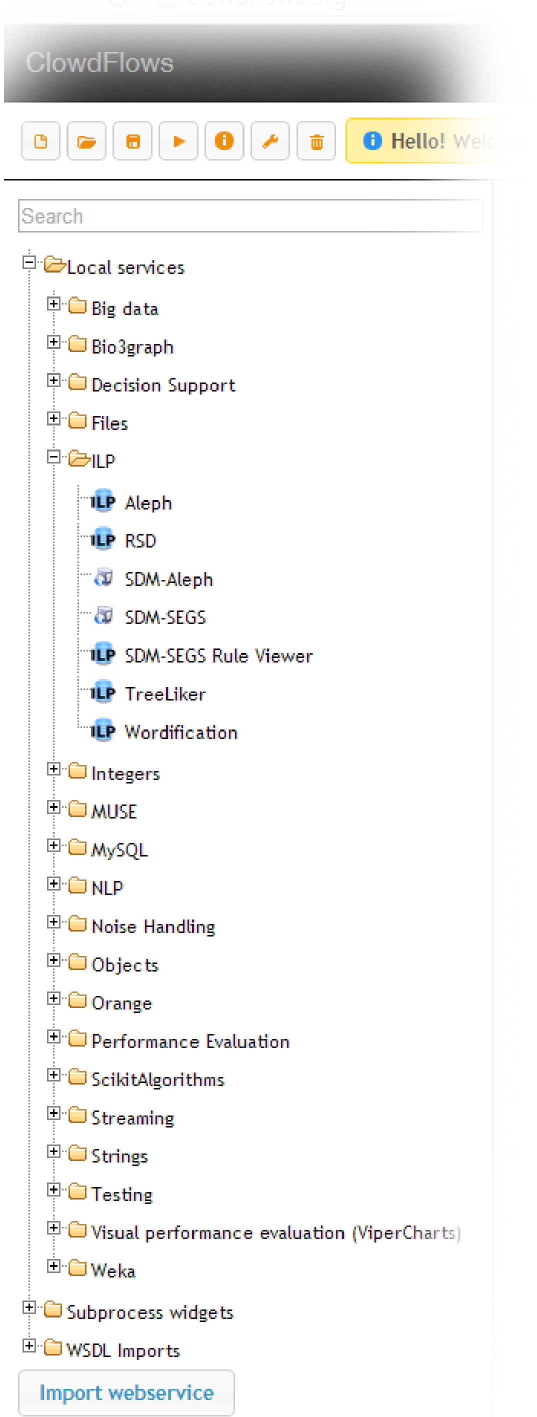


Third Generation Data Mining Platforms

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

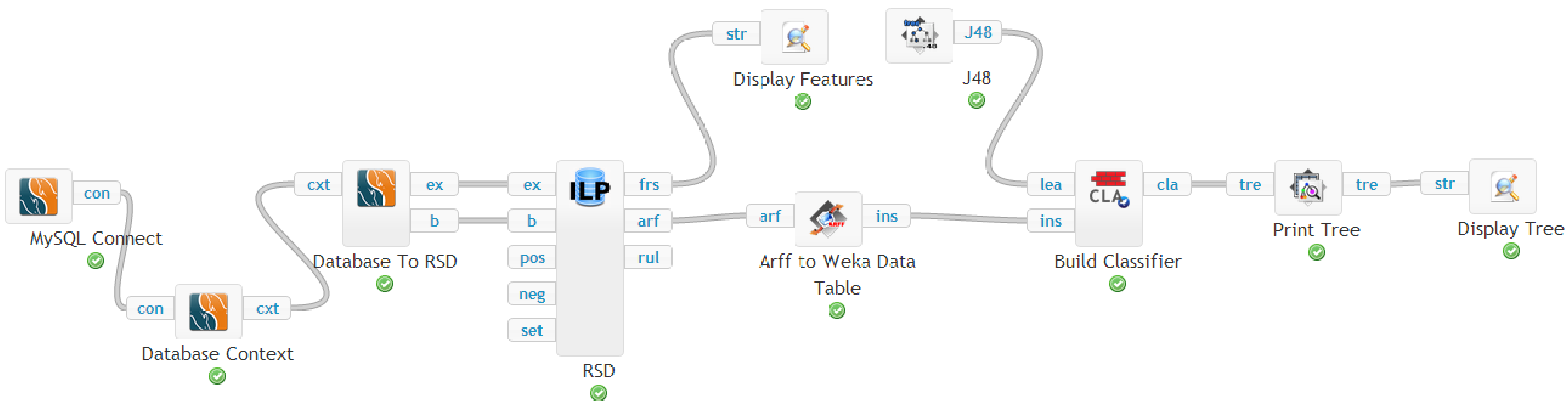
ClowdFlows platform

- **Large algorithm repository**
  – Relational data mining
  – All Orange algorithms
  – WEKA algorithms as web services
  – Data and results visualization
  – Text analysis
  – Social network analysis
  – Analysis of big data streams
- **Large workflow repository**
  – Enables access to our technology heritage



ClowdFlows platform

- Large repository of algorithms
- Large repository of workflows



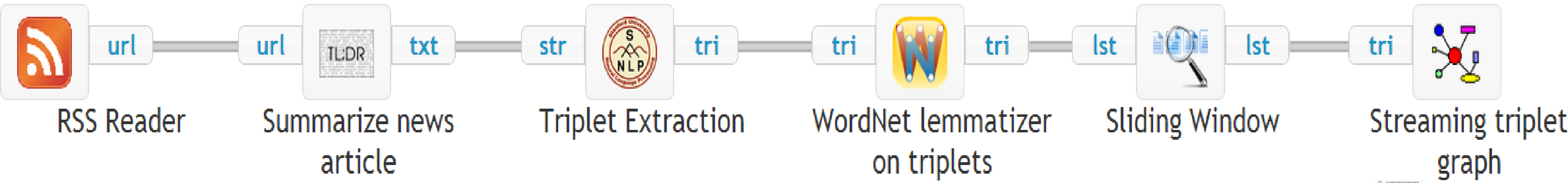
Example workflow:
Propositionalization with RSD
available in ClowdFlows at
http://clowdflows.org/workflow/611/

TextFlows

- Motivation:
  - Develop an online text mining platform for composition, execution and sharing of text mining workflows
- TextFlows platform – fork of ClowdFlows.org:
  - Specialized on text mining
  - Web-based user interface
  - Visual programming
  - Big roster of existing workflow (mostly text mining) components
  - Cloud-based service-oriented architecture

"Big Data" Use Case

- Real-time analysis of big data streams
- Example: semantic graph construction from news streams. http://clowdflows.org/workflow/1729/.



- Example: news monitoring by graph visualization (graph of CNN RSS feeds)

http://clowdflows.org/streams/data/31/1

Part I: Summary

- KDD is the overall process of discovering useful knowledge in data
  - many steps including data preparation, cleaning, transformation, pre-processing
- Data Mining is the data analysis phase in KDD
  - DM takes only 15%-25% of the effort of the overall KDD process
  - employing techniques from machine learning and statistics
- Predictive and descriptive induction have different goals: classifier vs. pattern discovery
- Many application areas, many powerful tools available

Summary of types of learning tasks

- **Supervised learning vs. Unsupervised learning = Learning from Labeled vs. Learning from Unlabeled data,** i.e. depending whether the data includes class labels for a predefined target class attribute or not.
- **Prediction (classification, predictive modeling, classifier learning)** - learning classifiers from class labeled data, e.g., decision tree learning
- **Concept learning** – learning classifiers for a preselected target class from binary labeled data
- **Regression** – learning classifiers from data with numeric class labels
- **Multi-label prediction** - learning classifiers from data labeled by several target class attributes
- **Description (descriptive pattern mining)** - learning individual rules/patterns, describing properties of parts of the data set, e.g. association rule learning
- **Subgroup discovery** – combining supervised learning from class labeled data and descriptive pattern mining
- **Clustering** – grouping of unlabeled data, based on data similarity

Technical paper outline

Book: Foundations of Rule Learning

Publisher: Springer, 2012

Authors: J. Fuernkranz, D. Gamberger and N. Lavrač

Chapter: Machine Learning and Data Mining

Course Outline

I. Introduction

- Data Mining and KDD process
- Introduction to Data Mining
- Data Mining platforms

II. Predictive DM

- Decision Tree learning
- Bayesian classifier
- Classification rule learning
- Classifier evaluation

III. Predictive DM

- Regression

IV. Descriptive DM

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

Part II. Predictive DM techniques

- Decision tree learning
- Bayesian Classifier
- Rule learning
- 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, ...

Predictive DM - classification formulated as a machine learning task

- Given a set of labeled **training examples** (n-tuples of attribute values, labeled by class name)

	A1	A2	A3	Class
example1	$v_{1,1}$	$v_{1,2}$	$v_{1,3}$	C_1
example2	$v_{2,1}$	$v_{2,2}$	$v_{2,3}$	C_2

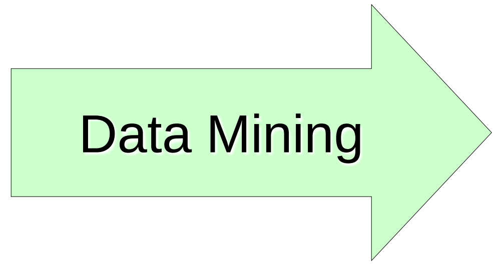
. .

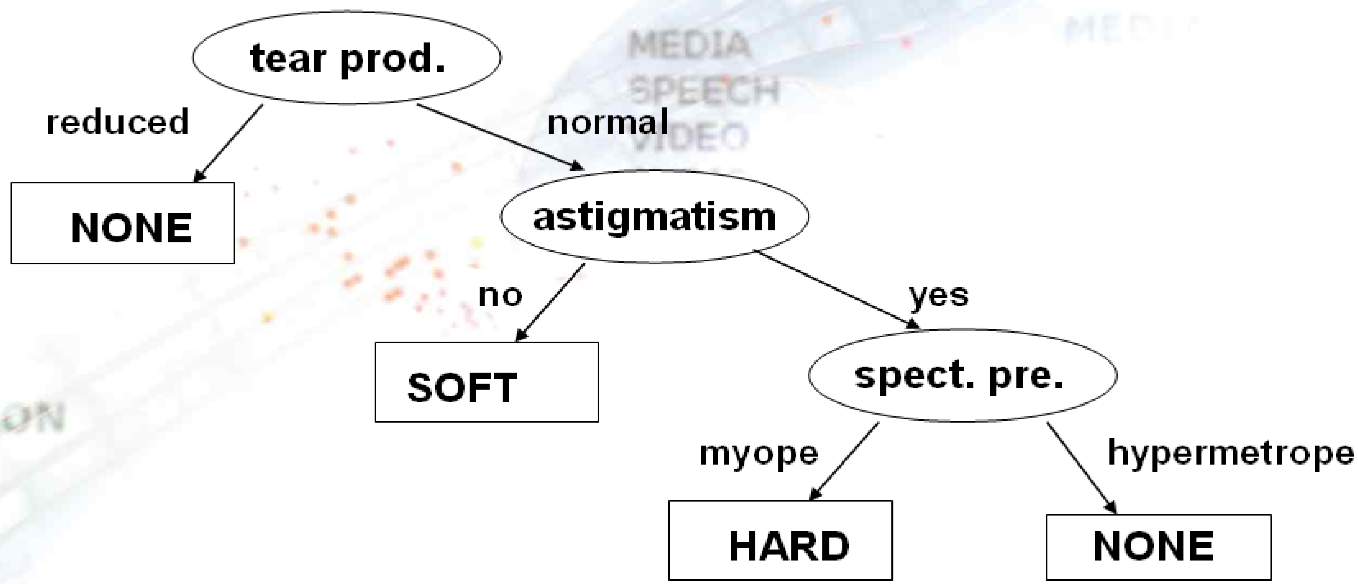
- Performing generalization from examples (induction)
- Find a **hypothesis** (a decision tree or classification rules) which explains the training examples, e.g. decision trees or classification rules of the form:

IF $(A_i = v_{i,k})$ & $(A_j = v_{j,l})$ & ... THEN Class = C_n

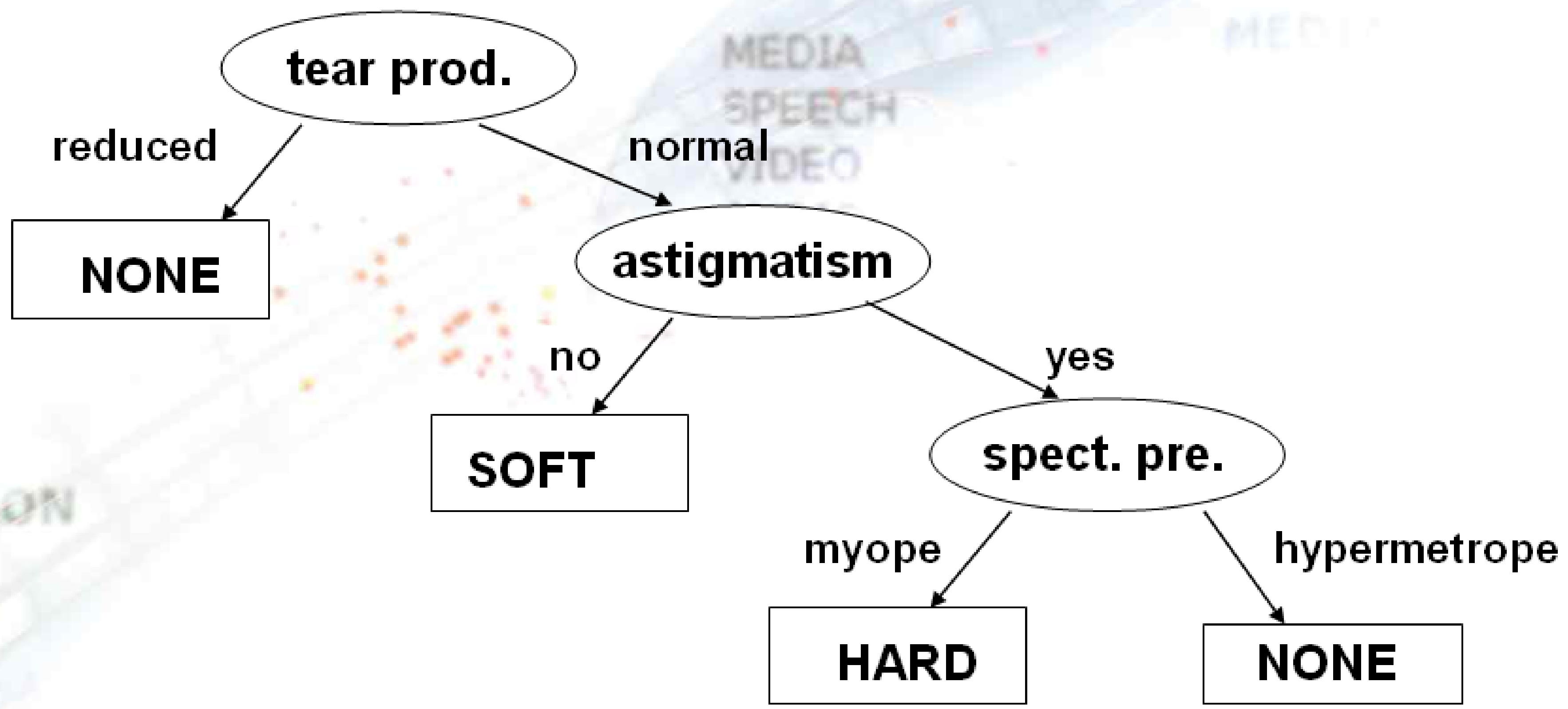
Decision Tree Learning

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



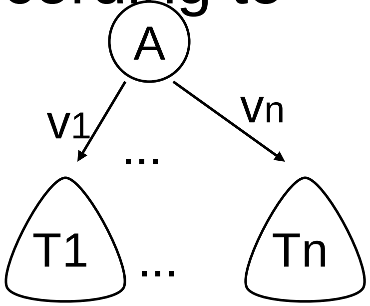


Decision Tree classifier



Decision tree learning algorithm

- ID3 (Quinlan 1979), CART (Breiman et al. 1984), C4.5, J48 in WEKA, ...
  - create the root node of the tree
  - if all examples from S belong to the same class Cj
    - then label the root with Cj
  - else
    - select the ‘most informative’ attribute **A** with values **v1, v2, … vn**
    - divide training set **S** into **S1,… , Sn** according to values **v1,…,vn**
    - recursively build sub-trees **T1,…,Tn** for **S1,…,Sn**



Decision tree search heuristics

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

Entropy

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

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

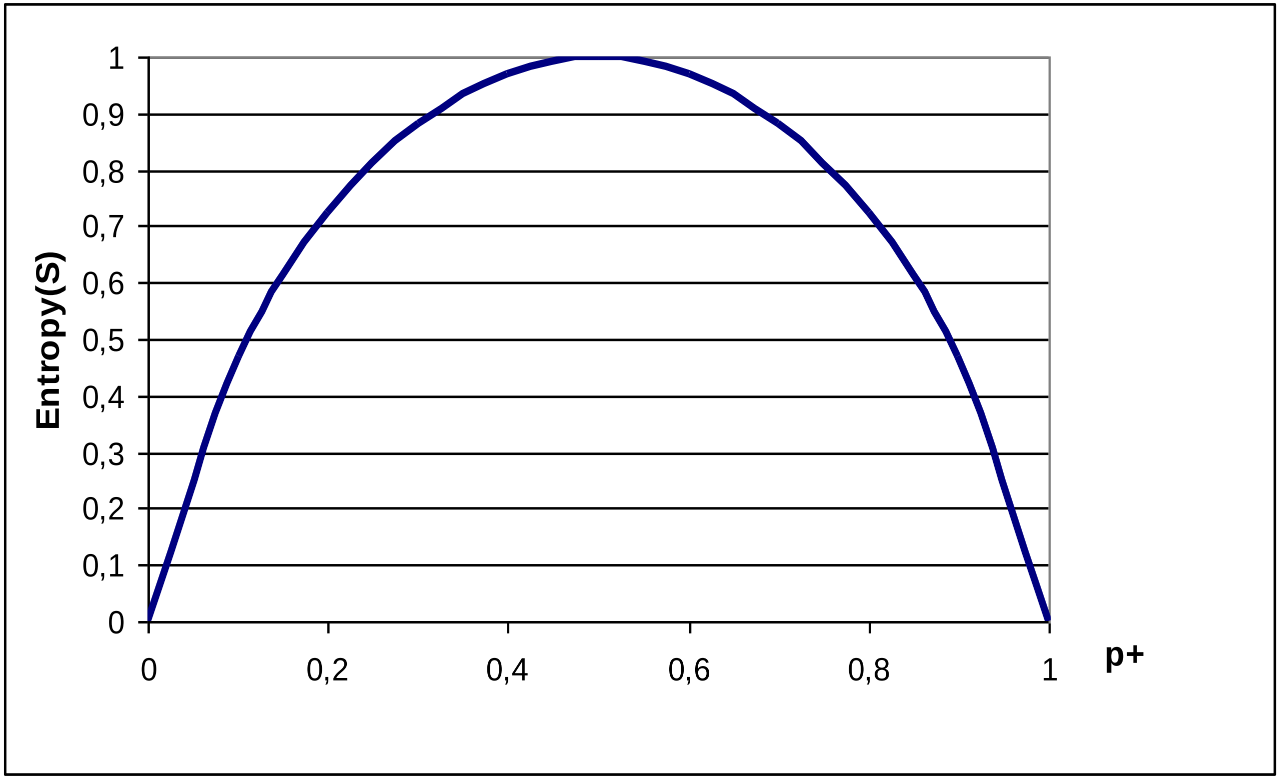
$\mathbf{p_c}$ - prior probability of class $\mathbf{C_c}$ (relative frequency of $\mathbf{C_c}$ in **S**)

- 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



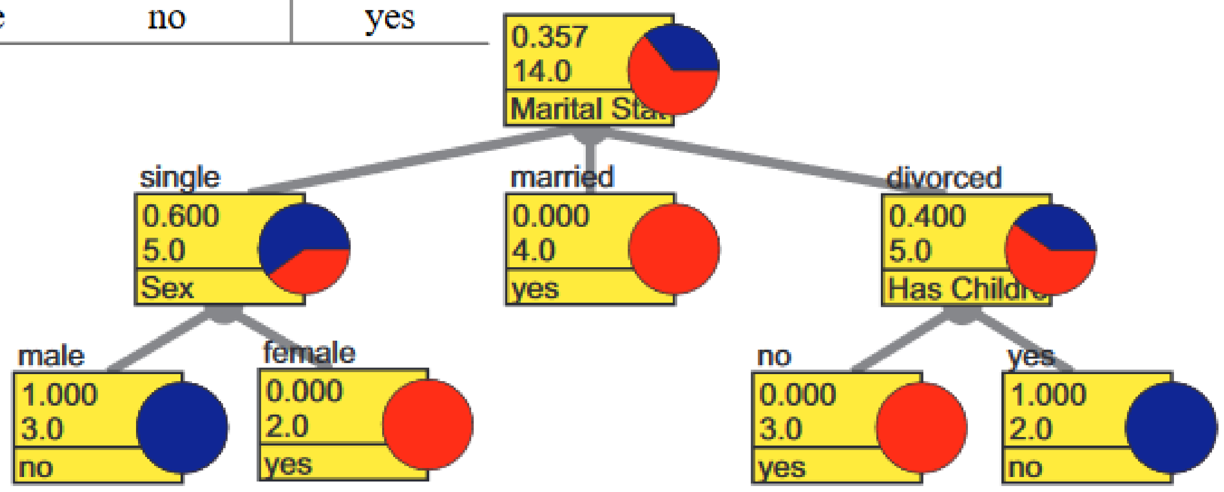
Entropy – why ?

- **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:

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

Binary classification problem: Survey data

Education	Marital Status	Sex	Has Children	Approved
primary	single	male	no	no
primary	single	male	yes	no
primary	married	male	no	yes
university	divorced	female	no	yes
university	married	female	yes	yes
secondary	single	male	no	no
university	single	female	no	yes
secondary	divorced	female	no	yes
secondary	single	female	yes	yes
secondary	married	male	yes	yes
primary	married	female	no	yes
secondary	divorced	male	yes	no
university	divorced	female	yes	no
secondary	divorced	male	no	yes



Entropy – example calculation

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

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

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

  $= 0.940$

Survey data: Entropy

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

Marital status ? $\xrightarrow{\text{single}}$ {e1,e2,e6,e7,e9} [2+, 3-] E=0.970

Marital status ? $\xrightarrow{\text{married}}$ {e3,e5,e10,e11} [4+, 0-] E=0

Marital status ? $\xrightarrow{\text{divorced}}$ {e4,e8,e12,e13, e14} [3+, 2-] E=0.970

Sex ? $\xrightarrow{\text{male}}$ [3+, 4-] E=0.985

Sex ? $\xrightarrow{\text{female}}$ [6+, 1-] E=0.592

Has children ? $\xrightarrow{\text{no}}$ [6+, 2-] E=0.811

Has children ? $\xrightarrow{\text{yes}}$ [3+, 3-] E=1.00

Information gain search heuristic

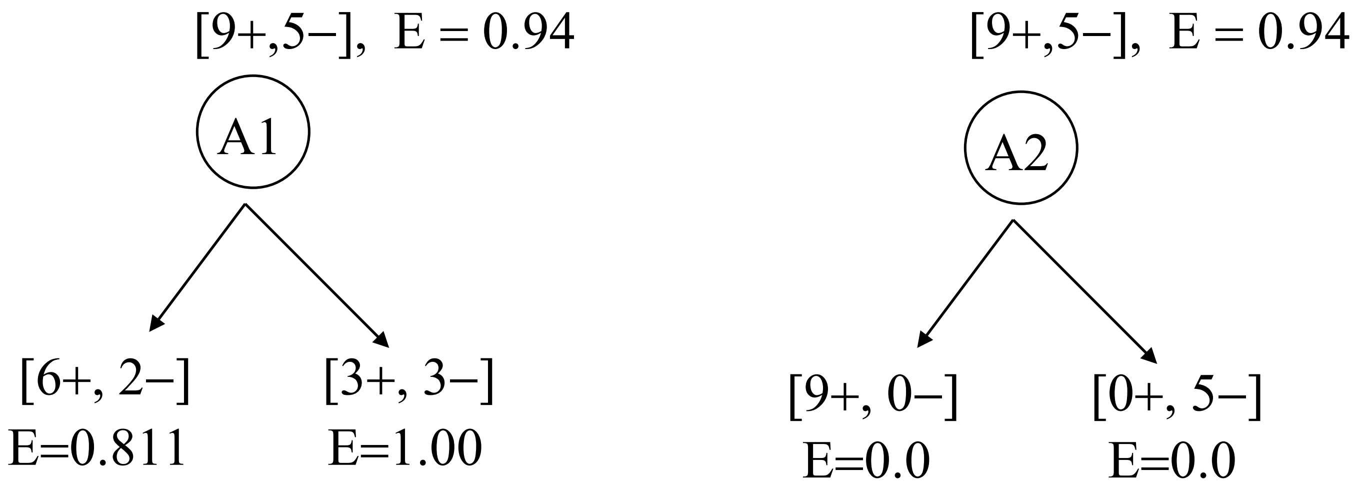
- **Information gain** measure is aimed to minimize the number of tests needed for the classification of a new object
- **Gain(S,A)** – expected reduction in entropy of S due to sorting on A

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

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

Information gain search heuristic

- Which attribute is more informative, A1 or A2 ?

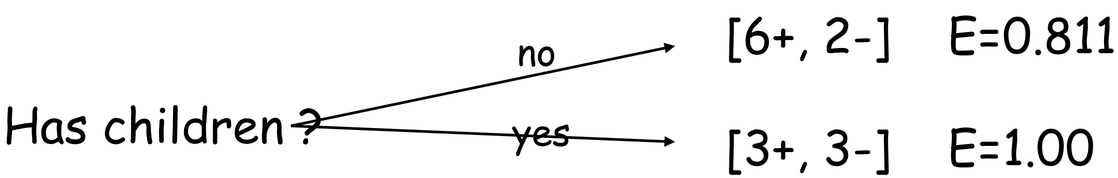


- Gain(S,A1) = 0.94 – (8/14 x 0.811 + 6/14 x 1.00) = 0.048
- Gain(S,A2) = 0.94 – 0 = 0.94 A2 has max Gain

Survey data: Information gain

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

- Values(Has children) = {no, yes}



  – S = [9+,5-], E(S) = 0.940
  – S_{no} = [6+,2-], E(S_{no}) = 0.811
  – S_{yes} = [3+,3-], E(S_{yes}) = 1.0
  – **Gain(S, Has children) =** E(S) - (8/14)E(S_{no}) - (6/14)E(S_{yes}) = 0.940 - (8/14)x0.811 - (6/14)x1.0=**0.048**

Survey data: Information gain

- **Which attribute is the best?**
  - Gain(S, Marital status)=0.246 *MAX !*
  - Gain(S, Sex)=0.151
  - Gain(S, Has children)=0.048
  - Gain(S, Education)=0.029

Survey data: Information gain

Marital status ?

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

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

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

- Which attribute should be tested here?
  - Gain(S_{sunny}, Sex) = 0.97-(3/5)0-(2/5)0 = 0.970 ***MAX !***
  - Gain(S_{sunny},Has children) = 0.97-(2/5)0-(2/5)1-(1/5)0 = 0.570
  - Gain(S_{sunny},Education) = 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

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

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

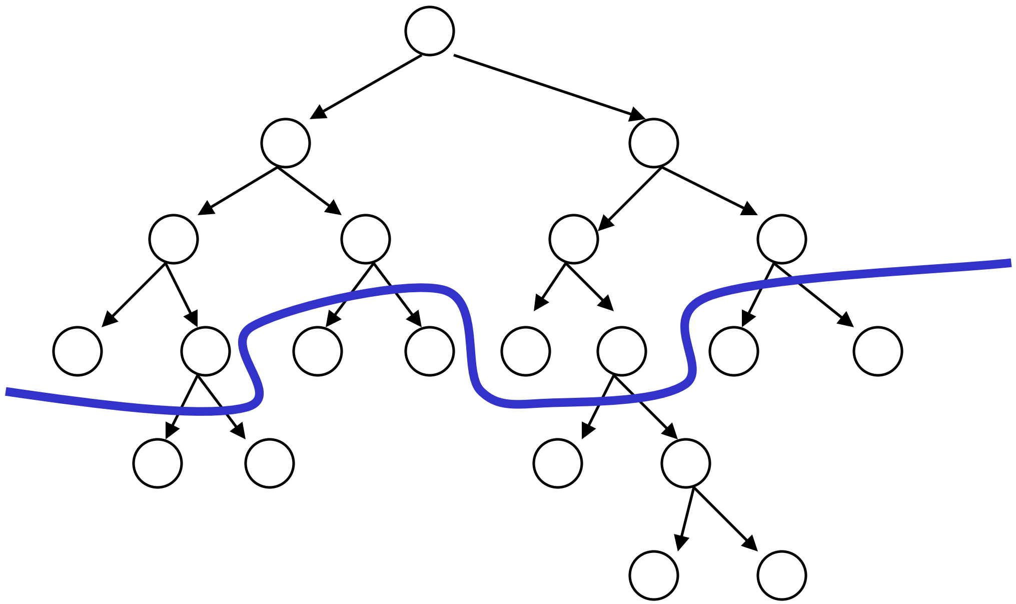
[2+,0-] (2) = 2+1 / 2+2 = 3/4

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

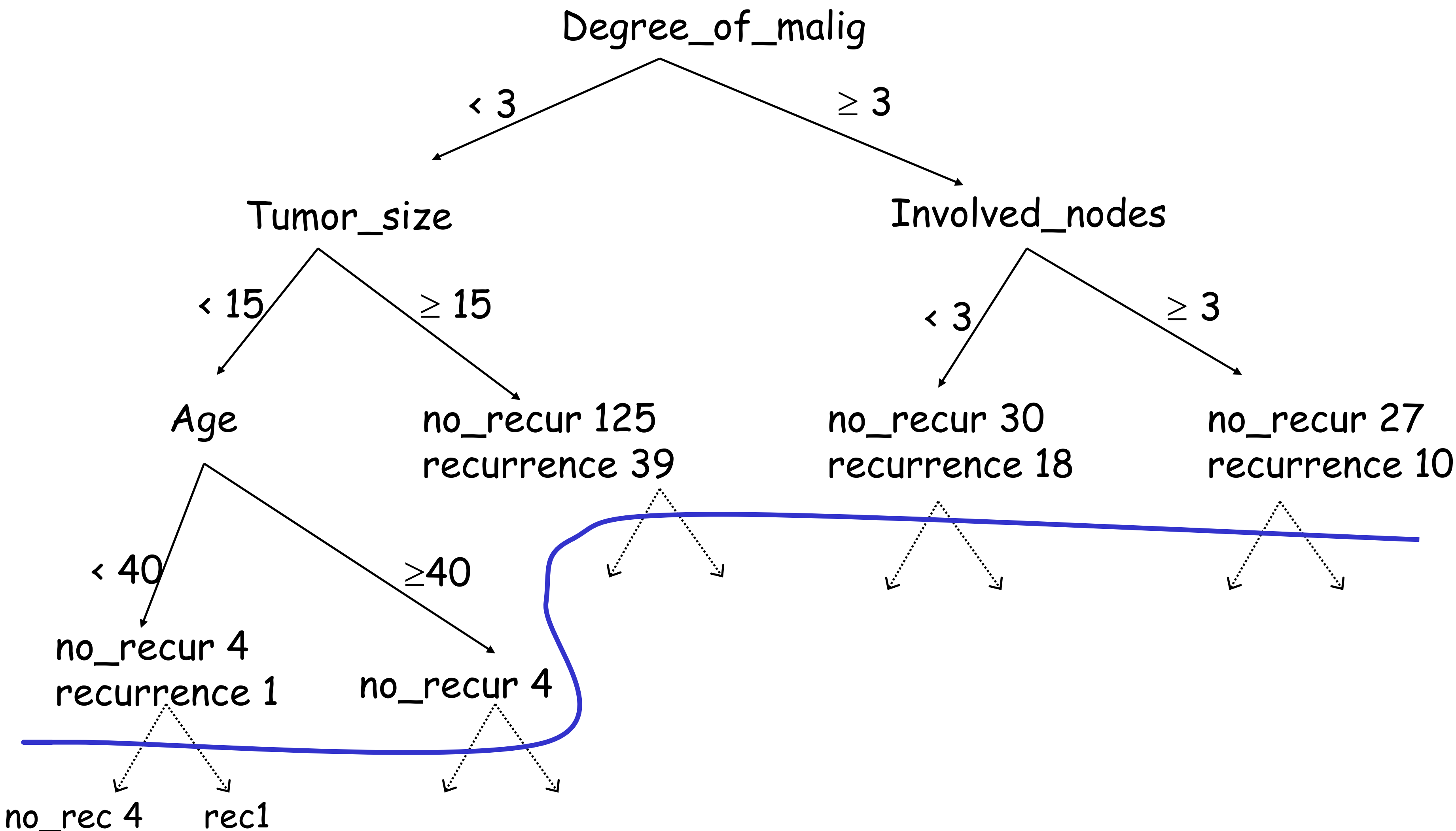
Sources of imperfection

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

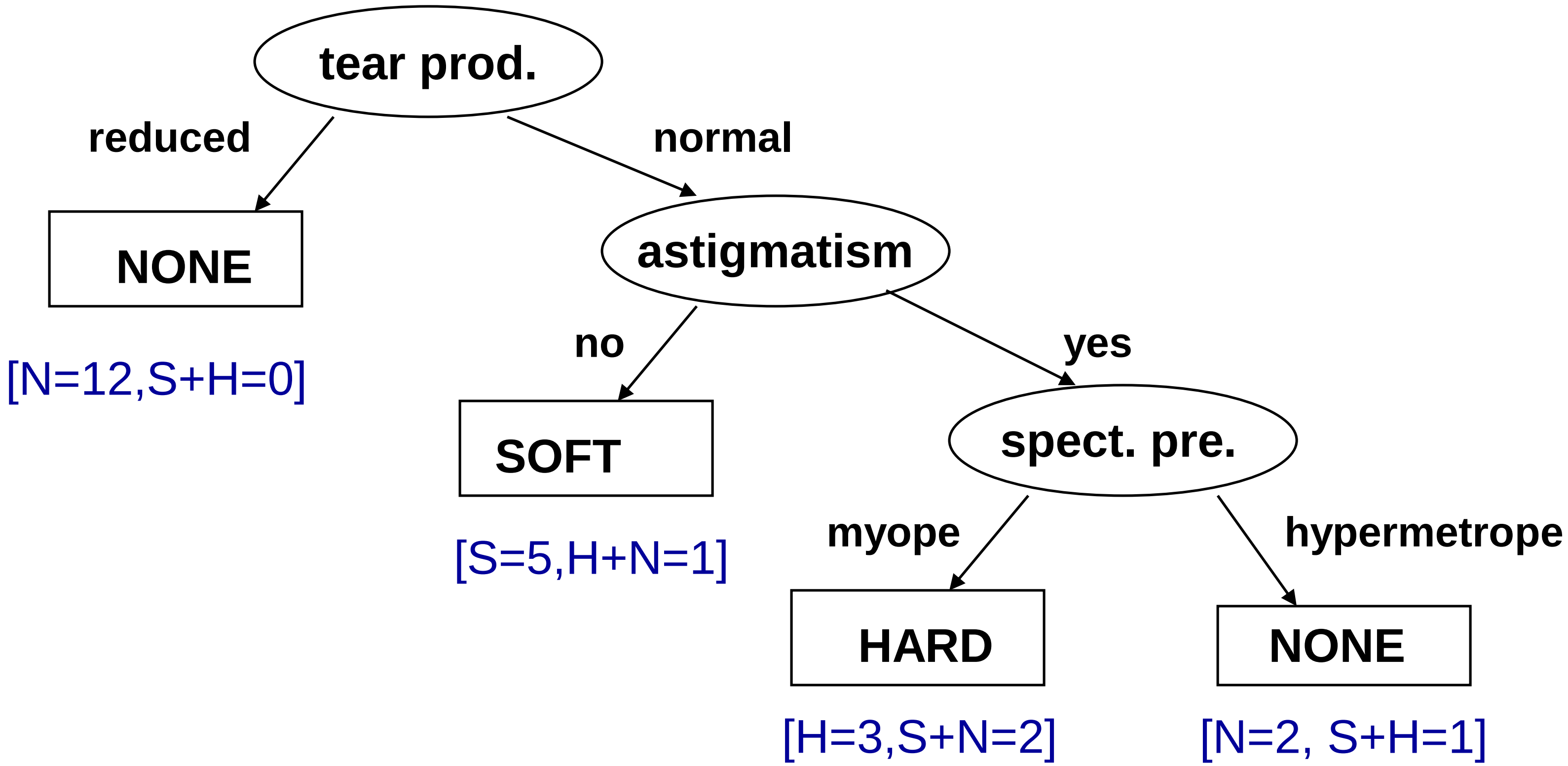
Handling noise – Tree pruning

- 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



Pruned decision tree for contact lenses recommendation

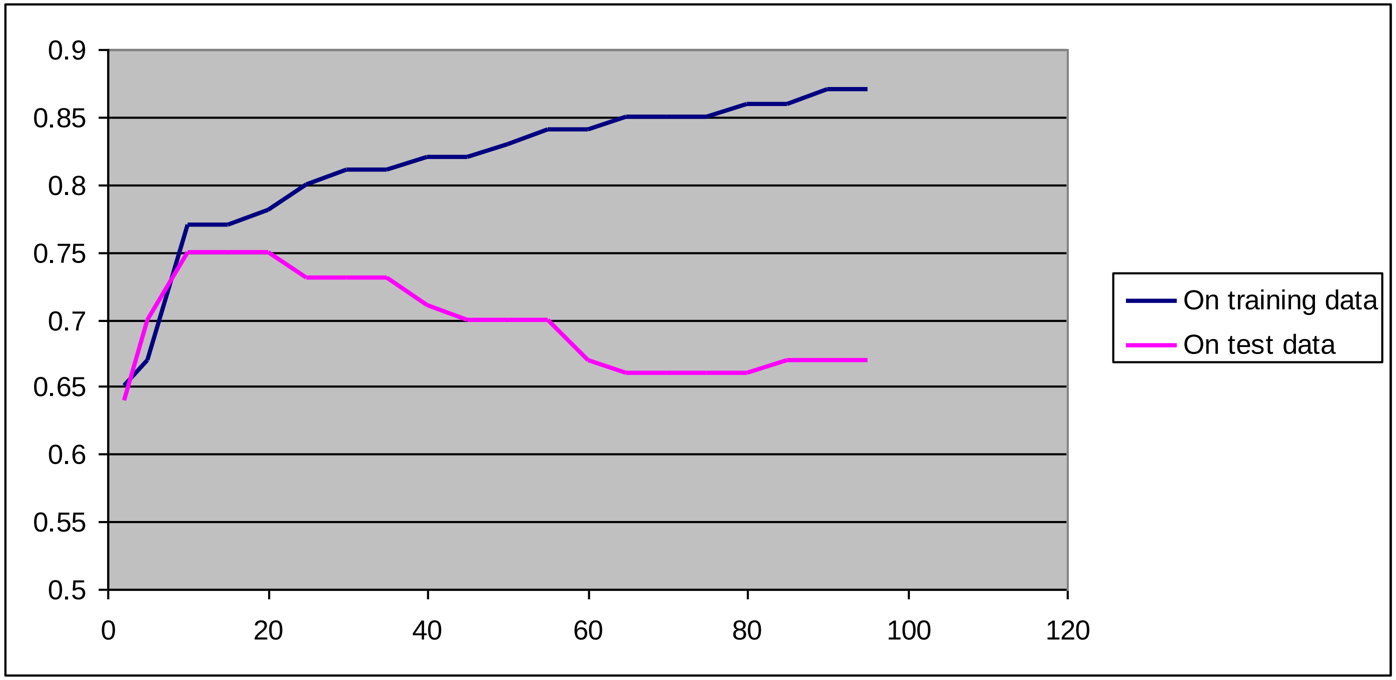


Accuracy and error

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

Overfitting and accuracy

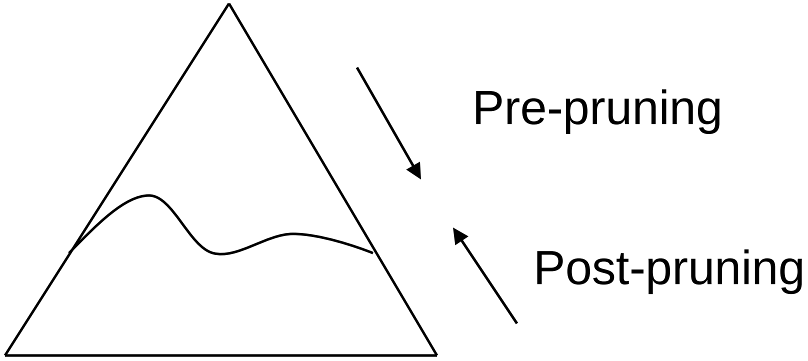
- Typical relation between tree size and accuracy



- Question: how to prune optimally?

Avoiding overfitting

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



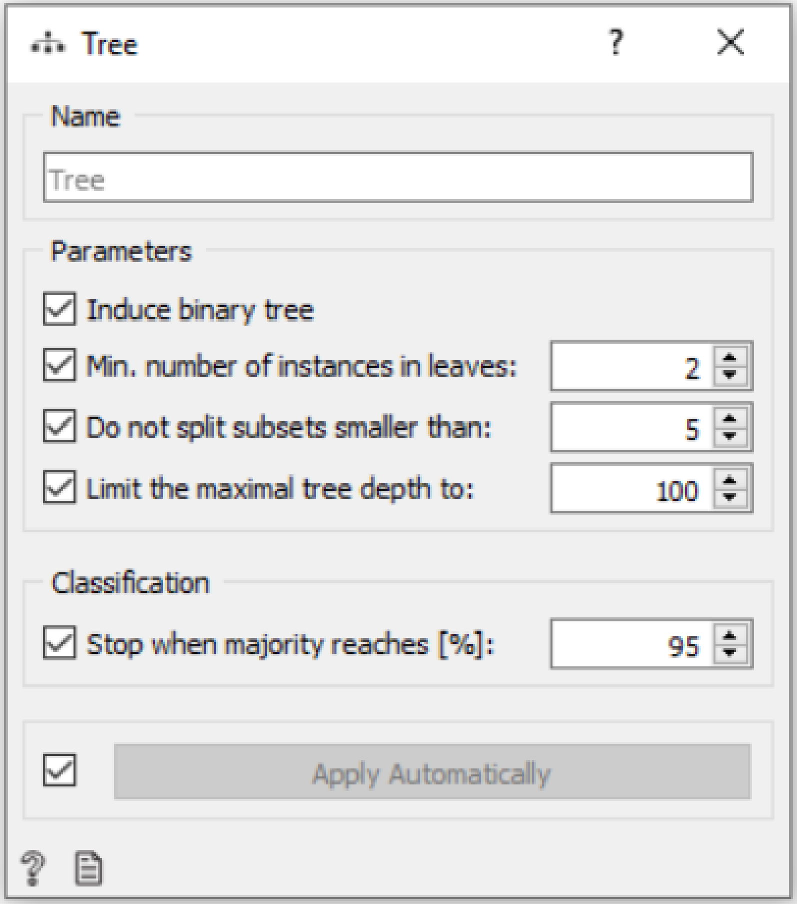
- forward pruning considered inferior (myopic)
- post pruning makes use of sub trees

Selected decision/regression tree learners

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

Selected decision tree learners

- Decision tree learners: Tree (in Orange)



Selected decision tree learners

- Homework
  - To prepare for the lecture of Petra Kralj Novak on Nov. 11, 2020 on using Tree software in Orange
  - See Blaž Zupan: Data Science with the OrangeToolbox
    http://videolectures.net/AIindustrySeminar2019_zupan_data_science/

Features of C4.5 and J48

- Implemented as part of the WEKA data mining workbench
- Handling noisy data: post-pruning
- Handling incompletely specified training instances: ‘unknown’ values (**?**)
  - in learning assign conditional probability of value v: $p(v|C) = p(vC) / p(C)$
  - 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

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

Classifier evaluation

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

n-fold cross validation

- A method for accuracy estimation of classifiers
- Partition set D into n disjoint, almost equally-sized folds T_i where U_i T_i = D
- **for** i = 1, ..., n **do**
  - form a training set out of n-1 folds: Di = D\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

Part II. Predictive DM techniques

- Decision tree learning
- Bayesian Classifier
- Rule learning
- 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:

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

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

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

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

problems with small samples

- Laplace estimate (prior probability):

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

assumes uniform prior distribution of k classes

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

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

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:

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

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

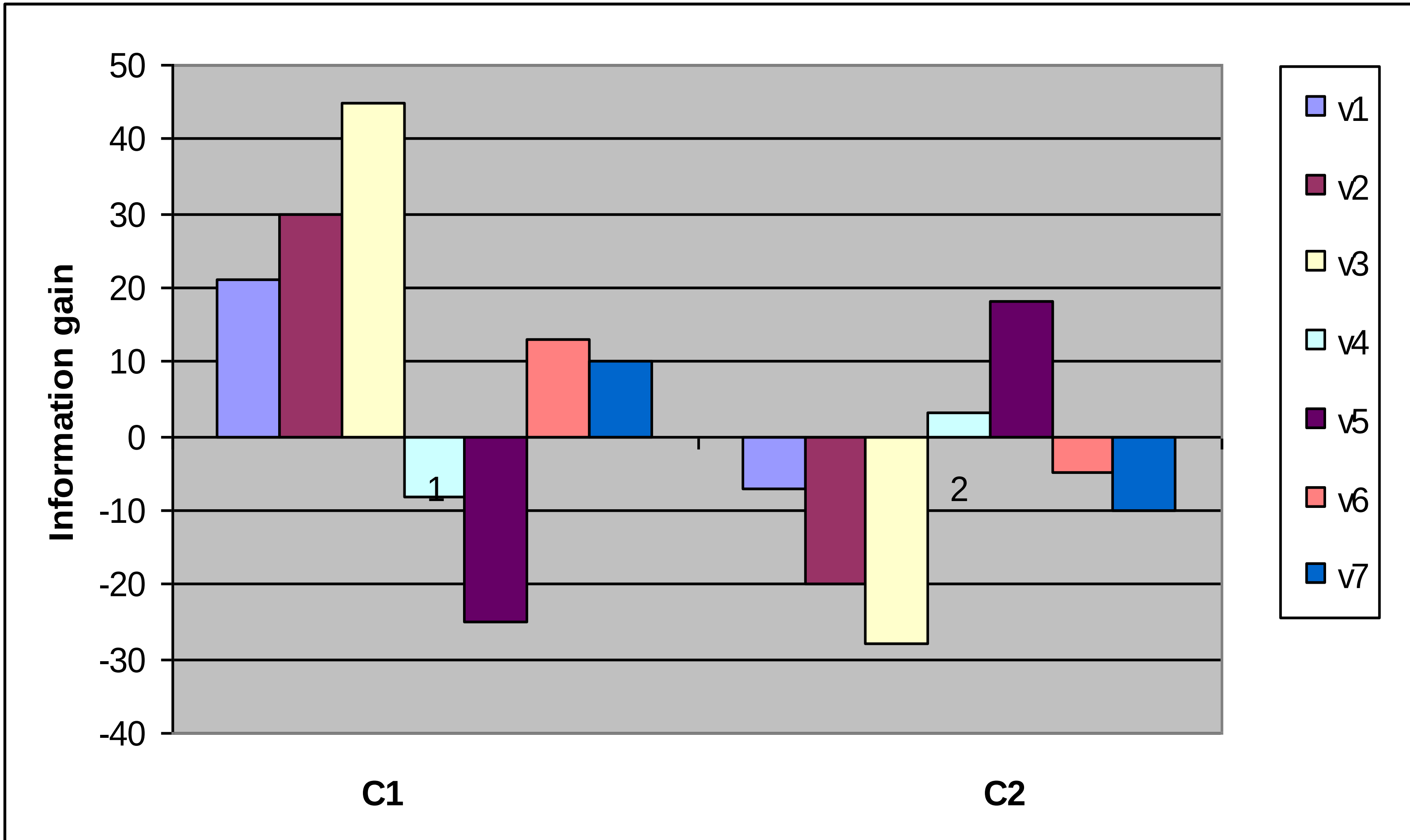
Example of explanation of semi-naïve Bayesian classifier

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

- Decision tree learning
- Bayesian Classifier
- Rule learning
- Evaluation

Rule Learning

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

data

knowledge discovery
from data

Rule learning

Model: a set of rules

Patterns: individual rules

Given: transaction data table, relational database (a set of objects, described by attribute values)

Find: a classification model in the form of a set of rules;
or a set of interesting patterns in the form of individual rules

Rule set representation

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

  IF Condition THEN Class

  Class IF Conditions

  Class $\leftarrow$ Conditions

- Form of CN2 rules:

  IF Conditions THEN MajClass [ClassDistr]

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

Contact lens data: Classification rules

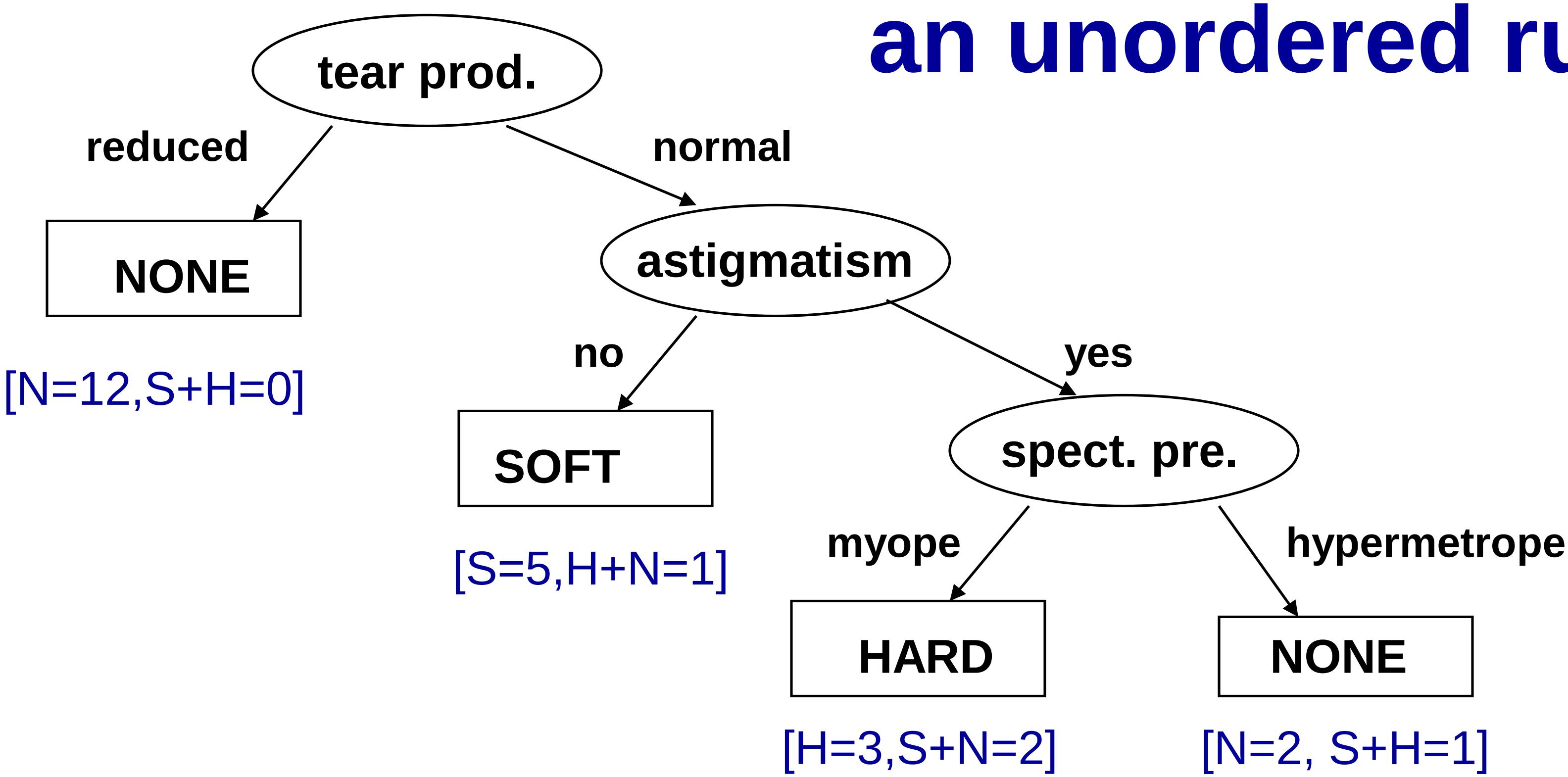
Type of task: prediction and classification
Hypothesis language: rules X ➔ C, if X then C
X conjunction of attribute values, C class

tear production=reduced → lenses=NONE
tear production=normal & astigmatism=yes &
spect. pre.=hypermetrope → lenses=NONE
tear production=normal & astigmatism=no → lenses=SOFT
tear production=normal & astigmatism=yes &
spect. pre.=myope → lenses=HARD
DEFAULT lenses=NONE

Rule learning

- Two rule learning approaches:
  - Learn decision tree, convert to rules
  - Learn set/list of rules
    - Learning an unordered set of rules
    - Learning an ordered list of rules
- Heuristics, overfitting, pruning

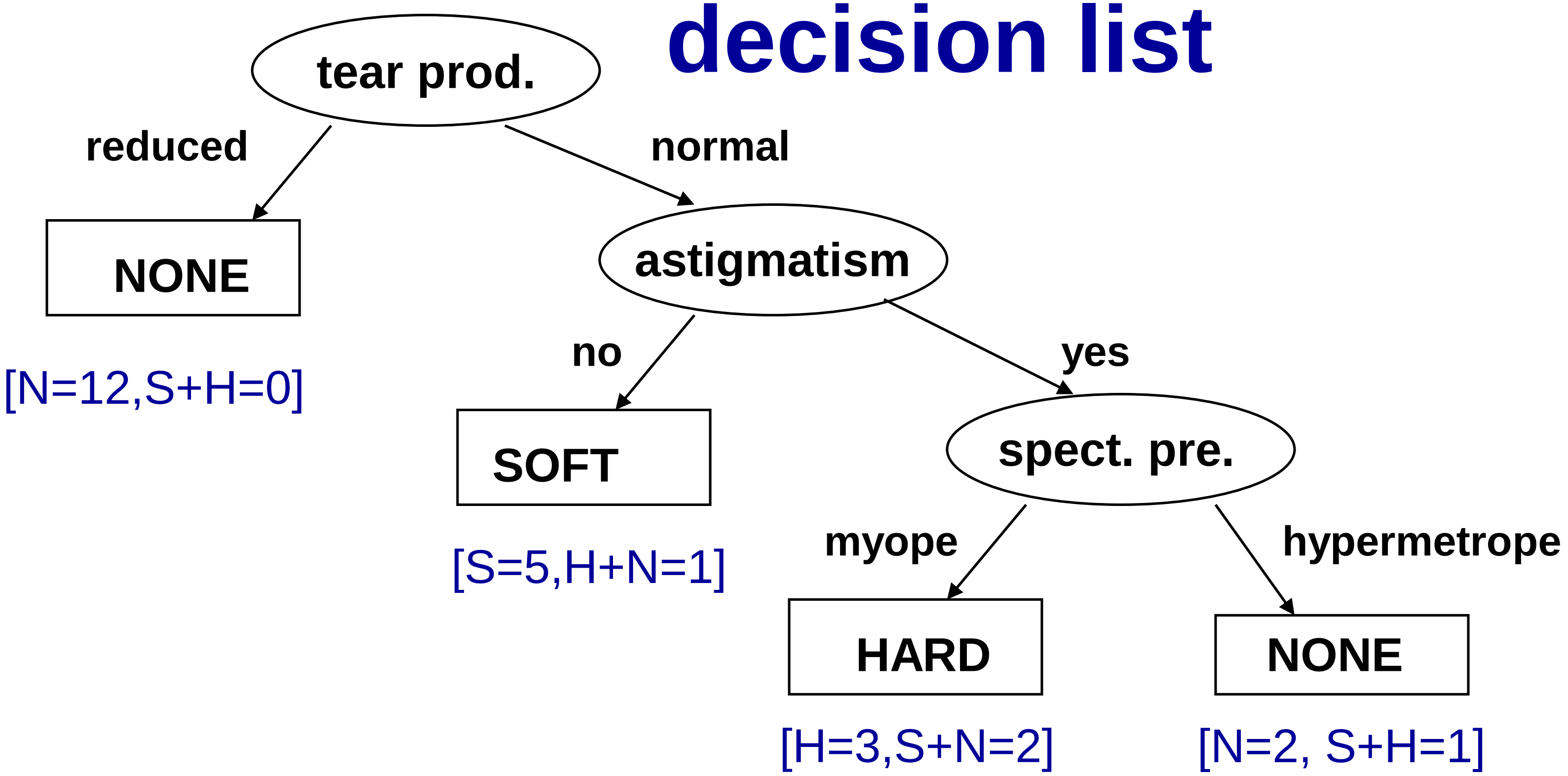
Contact lenses: convert decision tree to an unordered rule set



tear production=reduced **=>** lenses=NONE [S=0,H=0,N=12]
tear production=normal & astigmatism=yes & spect. pre.=hypermetrope **=>** lenses=NONE [S=0,H=1,N=2]
tear production=normal & astigmatism=no **=>** lenses=SOFT [S=5,H=0,N=1]
tear production=normal & astigmatism=yes & spect. pre.=myope **=>** lenses=HARD [S=0,H=3,N=2]
DEFAULT lenses=NONE

Order independent rule set (may overlap)

Contact lenses: convert decision tree to decision list



IF tear production=reduced THEN lenses=NONE
 ELSE /*tear production=normal*/
   IF astigmatism=no THEN lenses=SOFT
   ELSE /*astigmatism=yes*/
     IF spect. pre.=myope THEN lenses=HARD
      ELSE /* spect.pre.=hypermetrope*/
        lenses=NONE

Ordered (order dependent) rule list

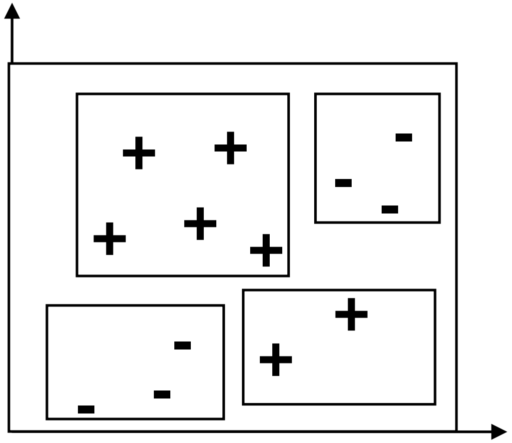
Converting decision tree to rules, and rule post-pruning (Quinlan 1993)

- Very frequently used method, e.g., in C4.5 and J48
- Procedure:
  - grow a full tree (allowing overfitting)
  - convert the tree to an equivalent set of rules
  - prune each rule independently of others
  - sort final rules into a desired sequence for use

Concept learning: Task reformulation for rule learning: (pos. vs. neg. examples of Target class)

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	17	myope	no	reduced	NO
O2	23	myope	no	normal	YES
O3	22	myope	yes	reduced	NO
O4	27	myope	yes	normal	YES
O5	19	hypermetrope	no	reduced	NO
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	YES
O15	43	hypermetrope	yes	reduced	NO
O16	39	hypermetrope	yes	normal	NO
O17	54	myope	no	reduced	NO
O18	62	myope	no	normal	NO
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	NO

Original covering algorithm (AQ, Michalski 1969,86)



Given examples of N classes $C_1, \ldots, C_N$

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

Multi-class learning: One-against-all learning strategy

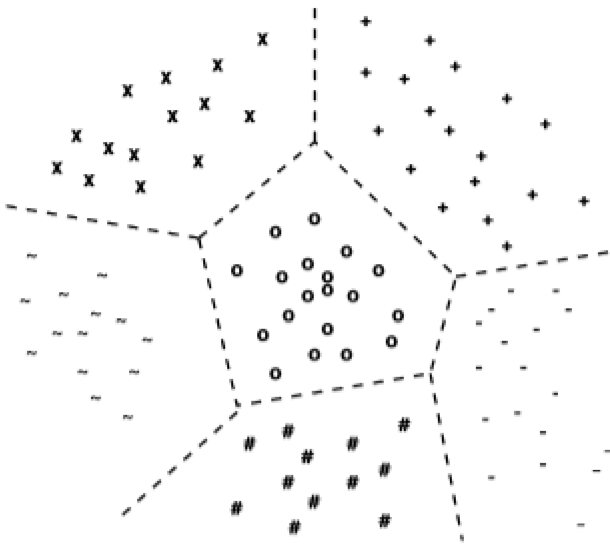


Fig. 10.2: A multiclass classification

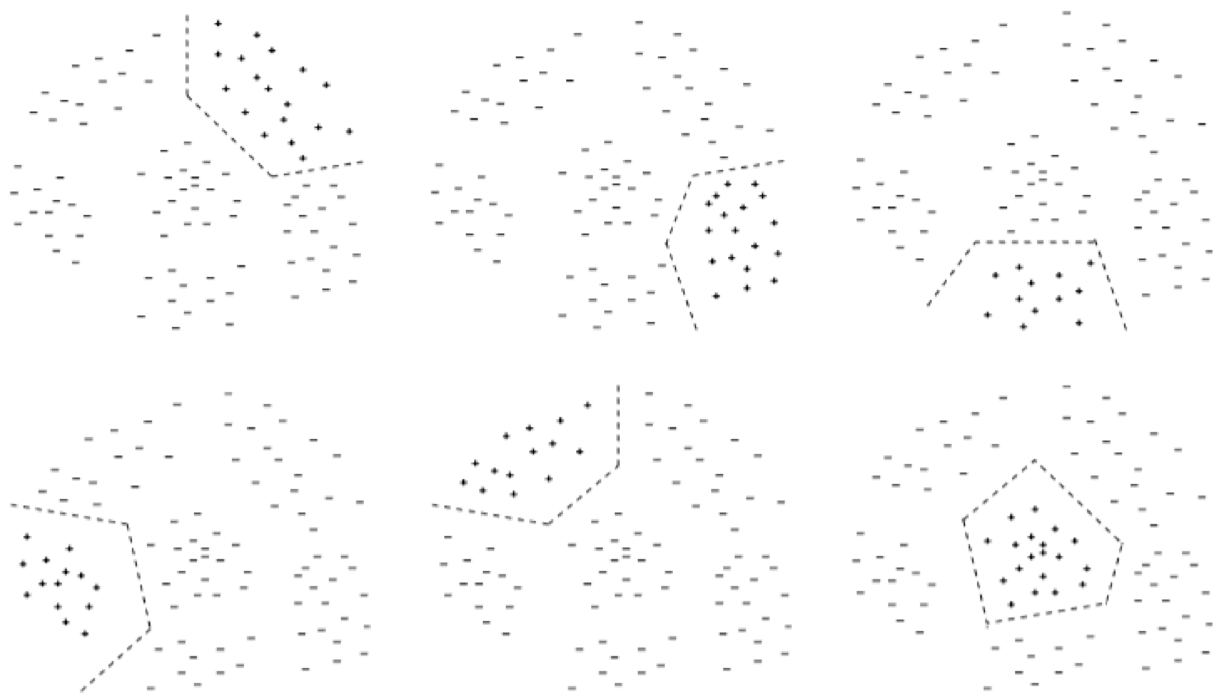
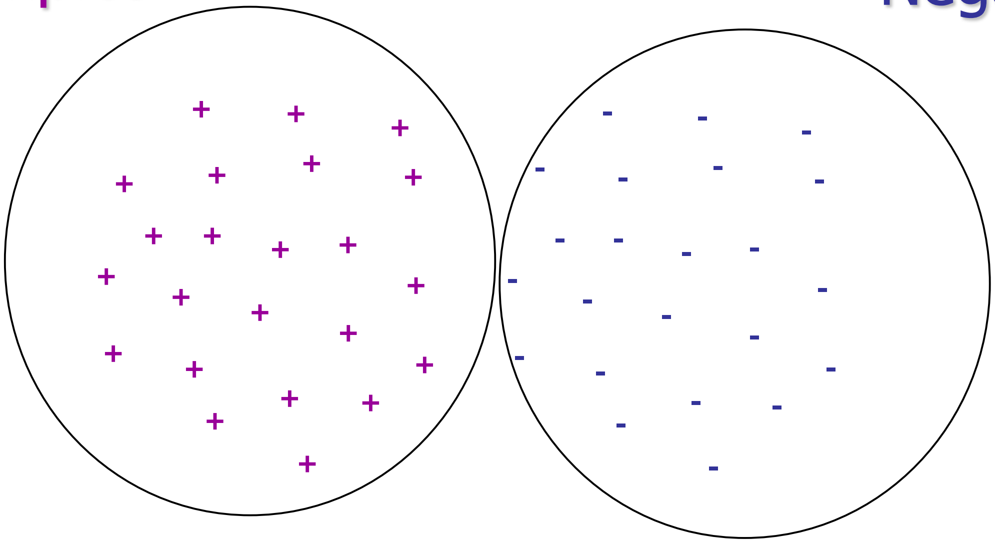


Fig. 10.4: The six binary learning problems that are the result of one-against-all class binarization of the multiclass dataset of Figure 10.2.

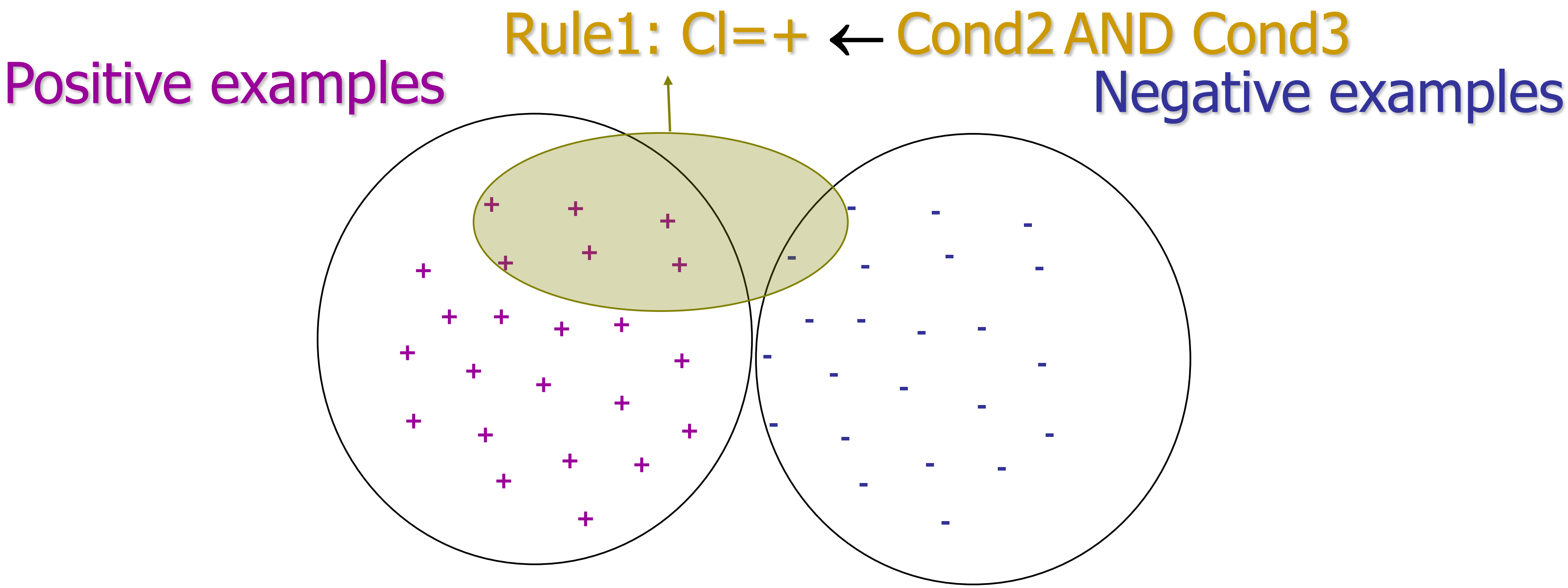
Covering algorithm

Positive examples

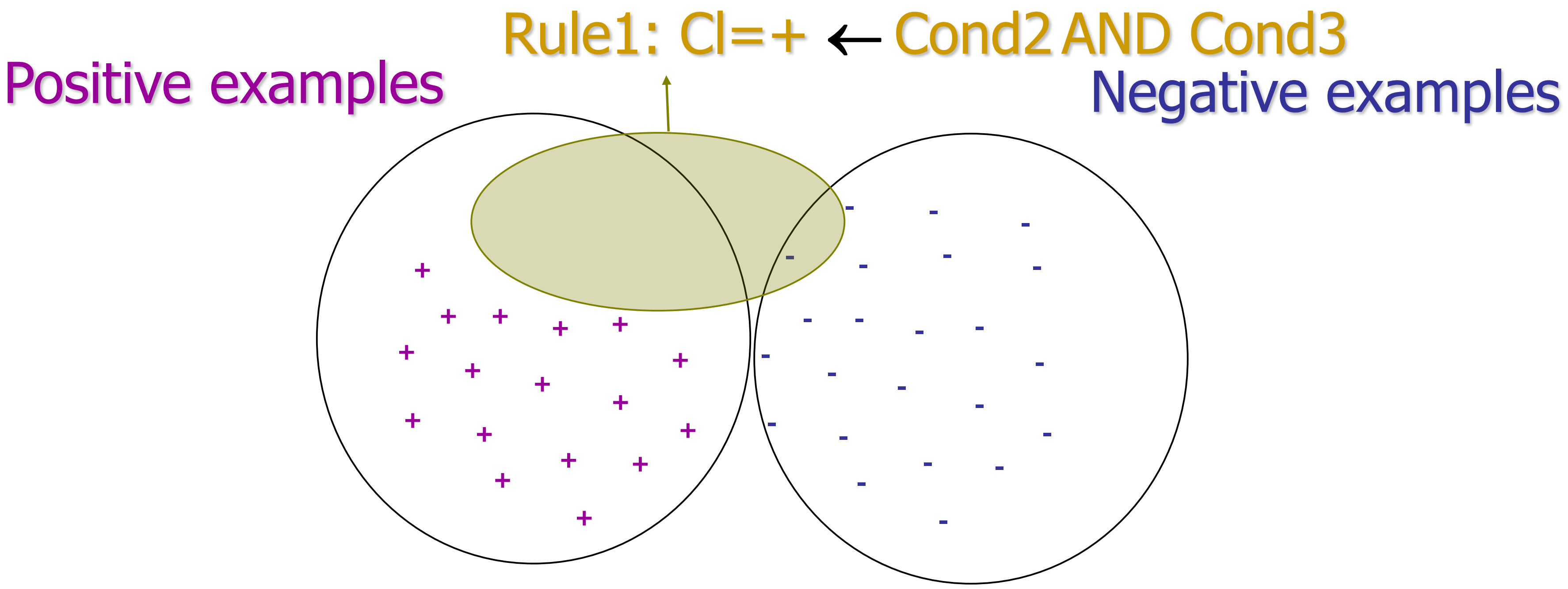
Negative examples



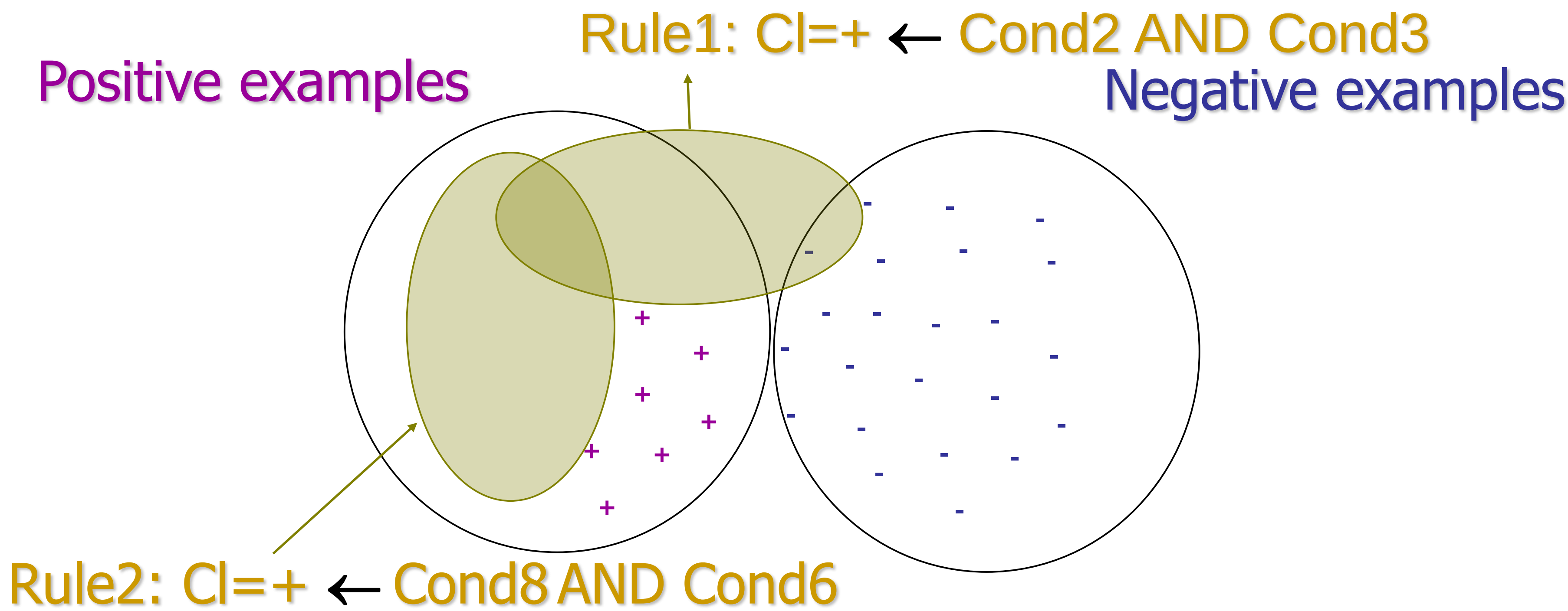
Covering algorithm



Covering algorithm



Covering algorithm



Learn-one-rule: Greedy vs. beam search

- learn-one-rule by greedy general-to-specific search, at each step selecting the `best' descendant, no backtracking
  - e.g., the best descendant of the initial rule

    lenses=NONE ←
  - is rule lenses=NONE ← tear production=reduced

- beam search: maintain a list of k best candidates at each step; descendants (specializations) of each of these k candidates are generated, and the resulting set is again reduced to k best candidates

Learn-one-rule: Greedy vs. beam search

- learn-one-rule by greedy general-to-specific search, at each step selecting the `best' descendant, no backtracking

  e.g., best descendant of initial rule lenses=NONE ←

  is rule lenses=NONE ← tear production=reduced

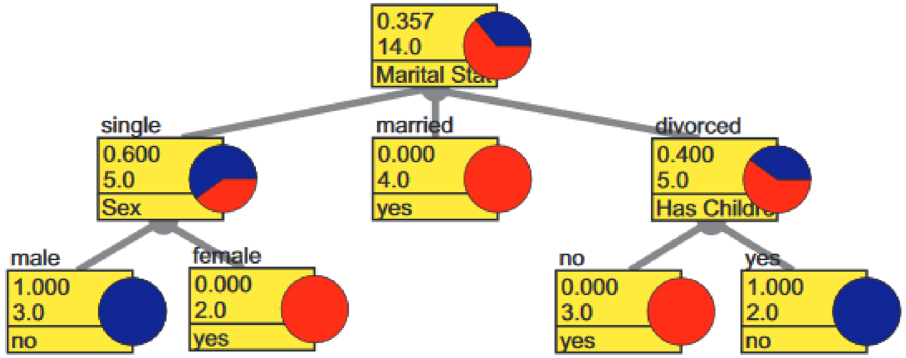
  e.g., best descendant of initial rule Approved=yes ←

  is rule Approved=yes ← Marital status = married

- 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

Recall: Binary classification problem - Survey data

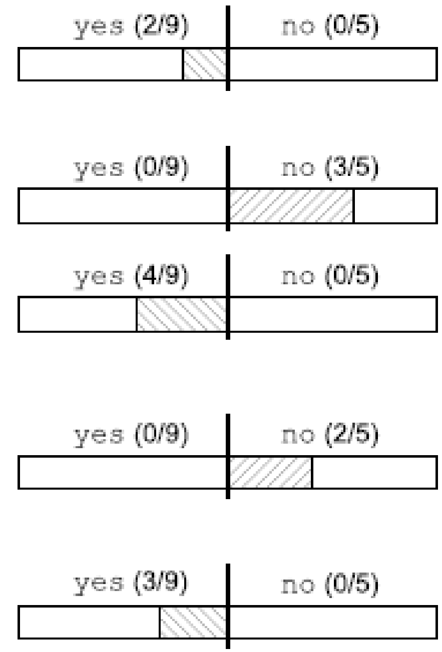
Education	Marital Status	Sex	Has Children	Approved
primary	single	male	no	no
primary	single	male	yes	no
primary	married	male	no	yes
university	divorced	female	no	yes
university	married	female	yes	yes
secondary	single	male	no	no
university	single	female	no	yes
secondary	divorced	female	no	yes
secondary	single	female	yes	yes
secondary	married	male	yes	yes
primary	married	female	no	yes
secondary	divorced	male	yes	no
university	divorced	female	yes	no
secondary	divorced	male	no	yes

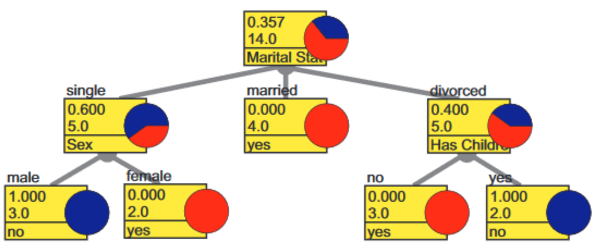


Survey data: Classification rule learning

Education	Marital Status	Sex	Has Children	Approved
primary	single	male	no	no
primary	single	male	yes	no
primary	married	male	no	yes
university	divorced	female	no	yes
university	married	female	yes	yes
secondary	single	male	no	no
university	single	female	no	yes
secondary	divorced	female	no	yes
secondary	single	female	yes	yes
secondary	married	male	yes	yes
primary	married	female	no	yes
secondary	divorced	male	yes	no
university	divorced	female	yes	no
secondary	divorced	male	no	yes

```
IF   MaritalStatus = single
 AND Sex = female
THEN Approved = yes

IF   MaritalStatus = single
 AND Sex = male
THEN Approved = no

IF   MaritalStatus = married
THEN Approved = yes

IF   MaritalStatus = divorced
 AND HasChildren = yes
THEN Approved = no

IF   MaritalStatus = divorced
 AND HasChildren = no
THEN Approved = yes
```

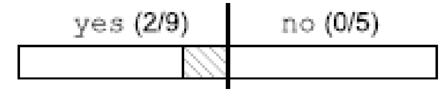




Survey data: Classification rule pruning

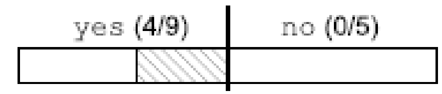
Education	Marital Status	Sex	Has Children	Approved
primary	single	male	no	no
primary	single	male	yes	no
primary	married	male	no	yes
university	divorced	female	no	yes
university	married	female	yes	yes
secondary	single	male	no	no
university	single	female	no	yes
secondary	divorced	female	no	yes
secondary	single	female	yes	yes
secondary	married	male	yes	yes
primary	married	female	no	yes
secondary	divorced	male	yes	no
university	divorced	female	yes	no
secondary	divorced	male	no	yes

```
IF   MaritalStatus = single
 AND Sex = female
THEN Approved = yes
```

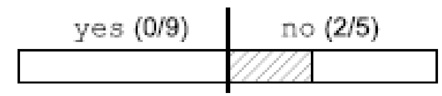



```
IF   MaritalStatus = single
 AND Sex = male
THEN Approved = no
```

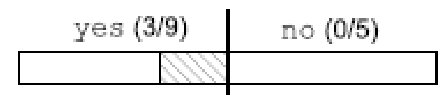

```
IF   MaritalStatus = married
THEN Approved = yes
```



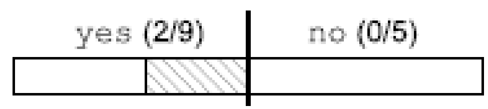

```
IF   MaritalStatus = divorced
 AND HasChildren = yes
THEN Approved = no
```



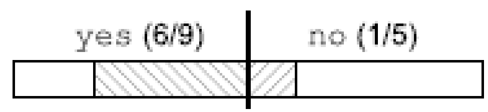

```
IF   MaritalStatus = divorced
 AND HasChildren = no
THEN Approved = yes
```



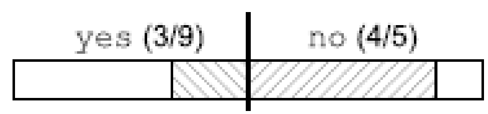

```
IF   MaritalStatus = married
THEN Approved = yes
```




```
IF   Sex = female
THEN Approved = yes
```

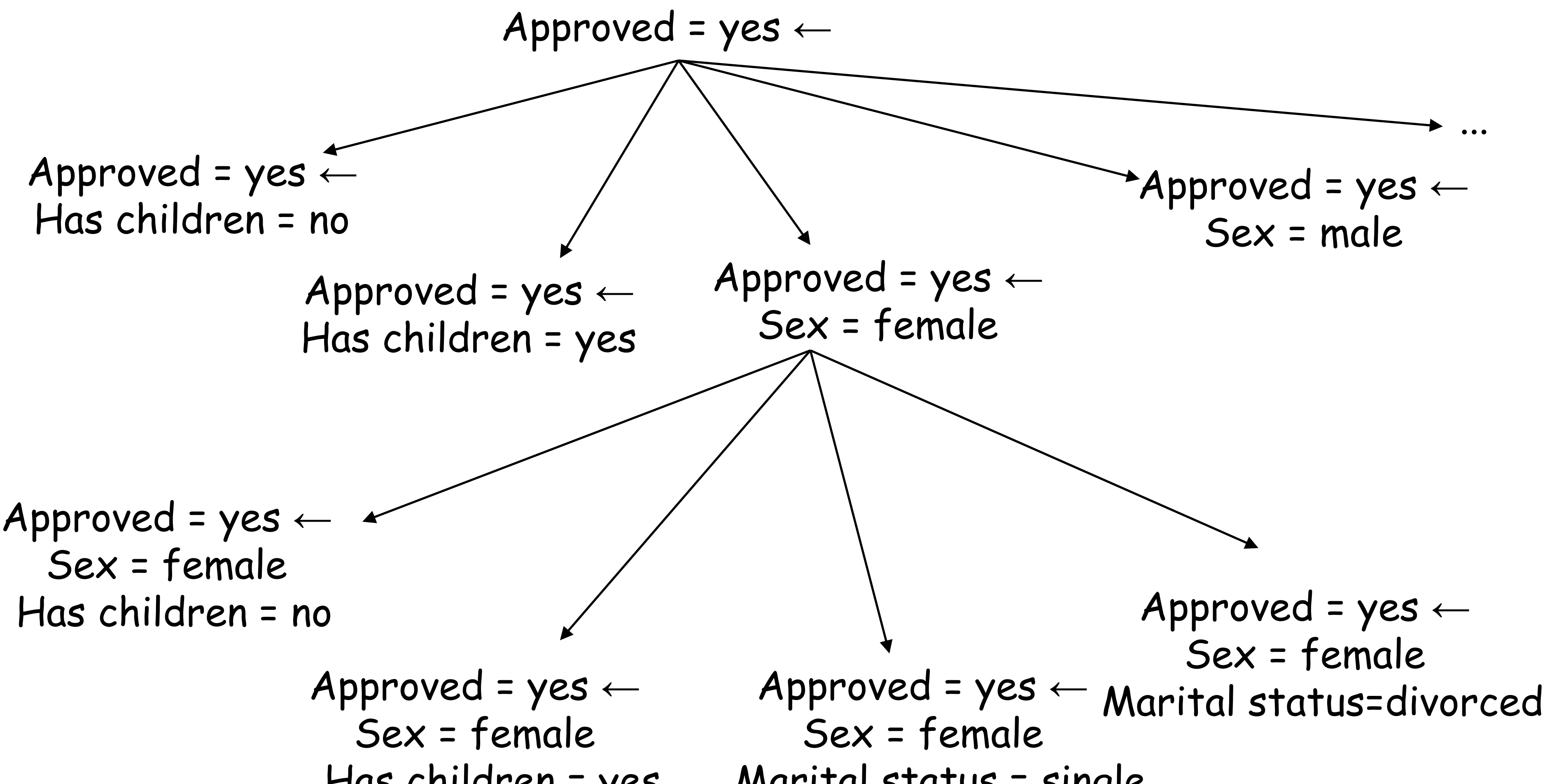



```
IF   Sex = male
THEN Approved = no
```

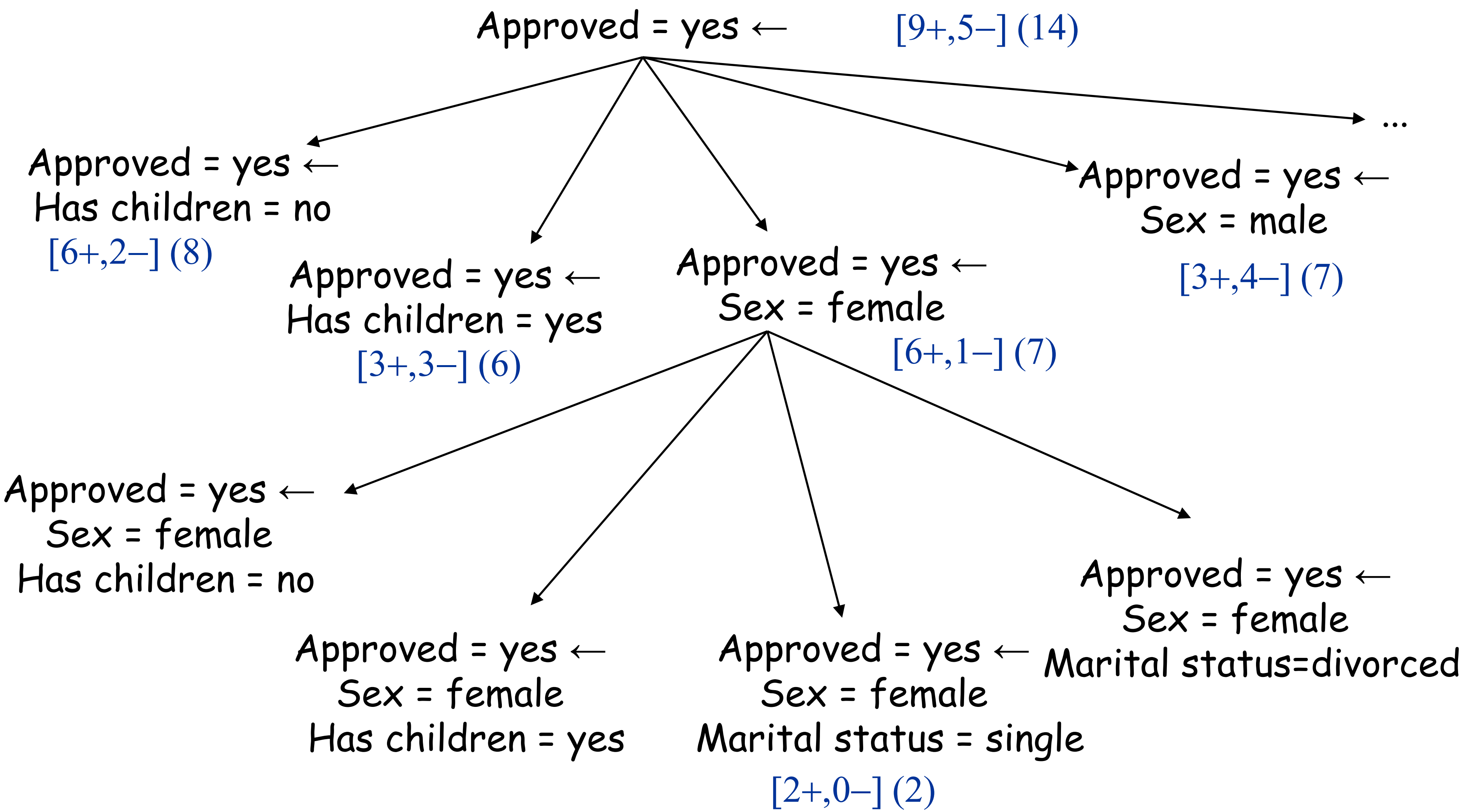



```
DEFAULT  Approved = yes
```

Learn-one-rule as heuristic search: 2nd rule in Survey data example



Learn-one-rule as heuristic search: 2nd rule in Survey data example



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

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

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(Cl))

Ordered set of rules: if-then-else rules

- rule Class IF Conditions is learned by first determining Conditions and then Class
- **Notice:** mixed sequence of classes C1, …, Cn in RuleBase
- **But: ordered** execution when classifying a new instance: rules are sequentially tried and the first rule that `fires' (covers the example) is used for classification
- **Decision list {R1, R2, R3, …, D}:** rules Ri are interpreted as **if-then-else** rules
- If no rule fires, then DefaultClass (majority class in E_{cur})

Sequential covering algorithm

- RuleBase := empty
- E_{cur}:= E
- **repeat**
  - learn-one-rule R
  - RuleBase := RuleBase U R
  - E_{cur} := E_{cur} - {examples covered and correctly classified by R} **(DELETE ONLY POS. EX.!)**
  - **until** performance(R, E_{cur}) < ThresholdR
- RuleBase := sort RuleBase by performance(R,E)
- return RuleBase

Learn ordered set of rules (CN2, Clark and Niblett 1989)

- RuleBase := empty
- E_{cur}:= E
- **repeat**
  - learn-one-rule R
  - RuleBase := RuleBase U R
  - E_{cur} := E_{cur} - {all examples covered by R} **(NOT ONLY POS. EX.!)**
- **until** performance(R, E_{cur}) < ThresholdR
- RuleBase := sort RuleBase by performance(R,E)
- RuleBase := RuleBase U DefaultRule(E_{cur})

Learn-one-rule: Beam search in CN2

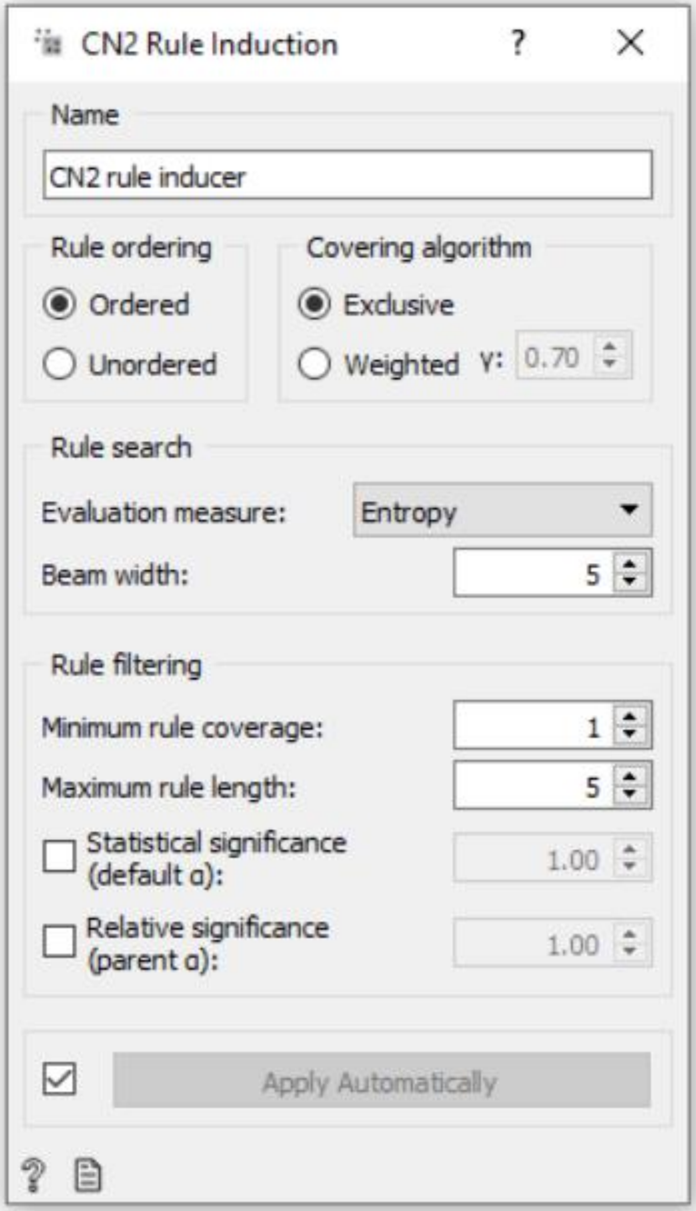
- Beam search in CN2 learn-one-rule algo.:
  - construct BeamSize of best rule bodies (conjunctive conditions) that are statistically significant
  - BestBody - min. entropy of examples covered by Body
  - construct best rule R := Head $\leftarrow$ BestBody by adding majority class of examples covered by BestBody in rule Head
- performance (R, E_{cur}) : - Entropy(E_{cur})
  - performance(R, E_{cur}) < ThresholdR (neg. num.)
  - Why? Ent. > t is bad, Perf. = -Ent < -t is bad

Variations

- Sequential vs. simultaneous covering of data (as in TDIDT): choosing between attribute-values vs. choosing attributes
- Learning rules vs. learning decision trees and converting them to rules
- Pre-pruning vs. post-pruning of rules
- What statistical evaluation functions to use
- Probabilistic classification

- Best performing rule learning algorithm: Ripper
- JRip implementation of Ripper in WEKA, available in ClowdFlows

CN2 rule learner in Orange



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

- Decision tree learning
- Bayesian Classifier
- Rule learning
- Evaluation

Classifier evaluation

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

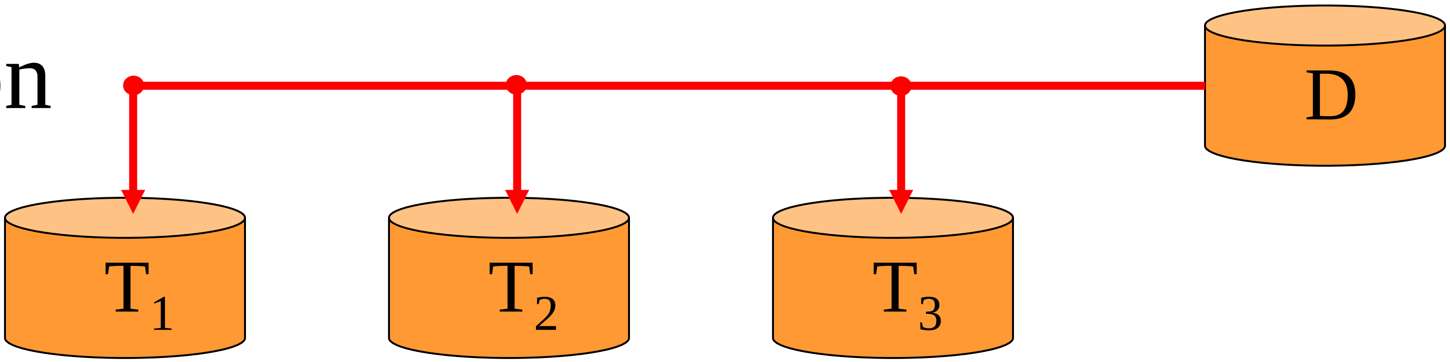
Evaluating hypotheses

- **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

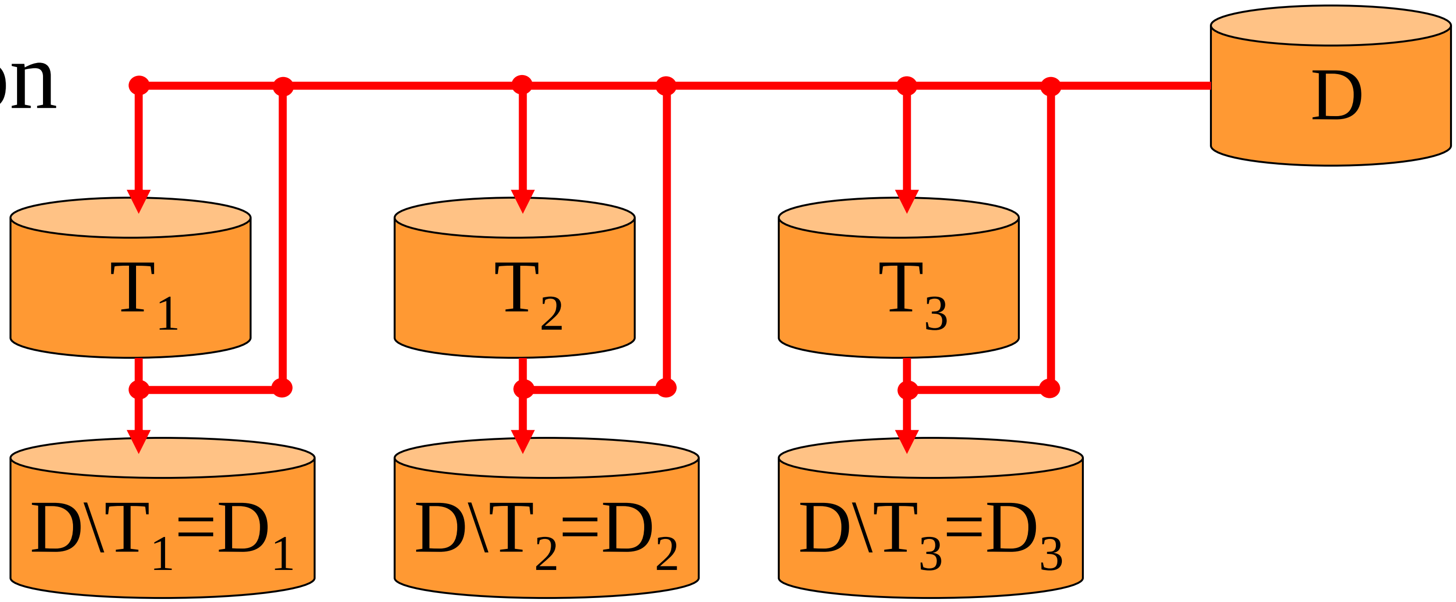
n-fold cross validation

- A method for accuracy estimation of classifiers
- Partition set D into n disjoint, almost equally-sized folds T_i where $U_i\ T_i = D$
- **for** i = 1, ..., n **do**
  - form a training set out of n-1 folds: Di = $D \backslash T_i$
  - 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

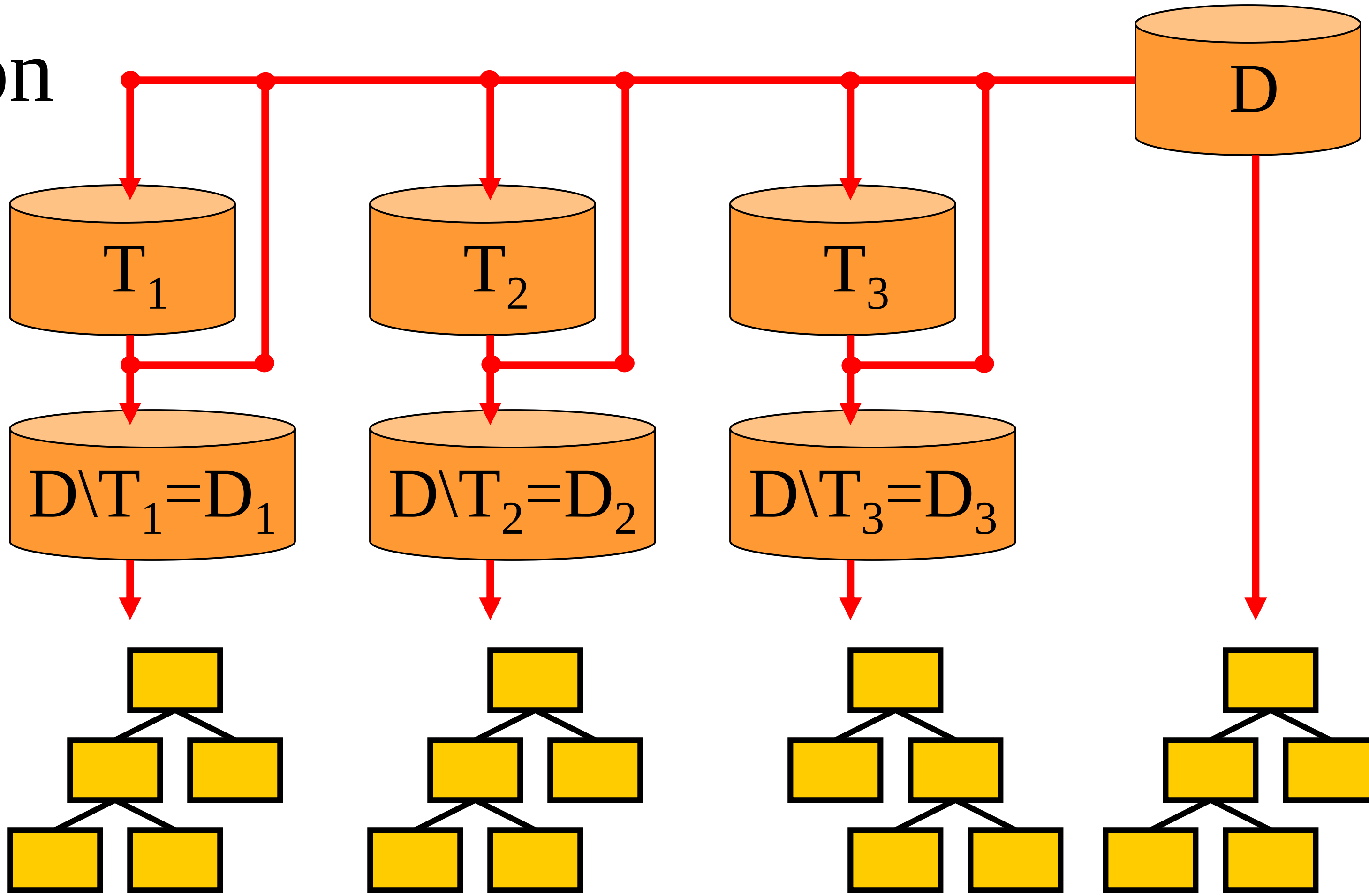
- Partition



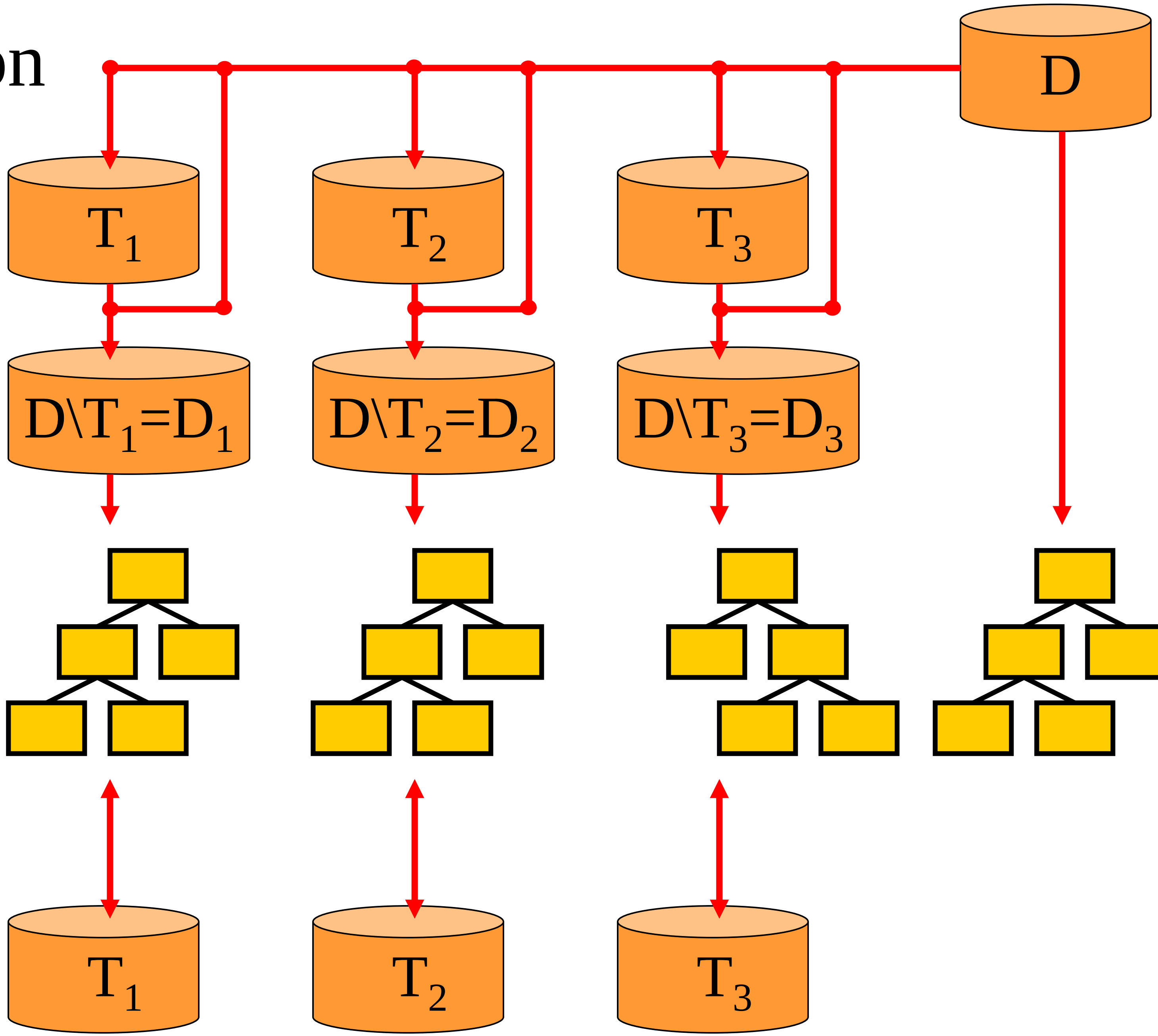
- Partition
- Train



- 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

ROC space

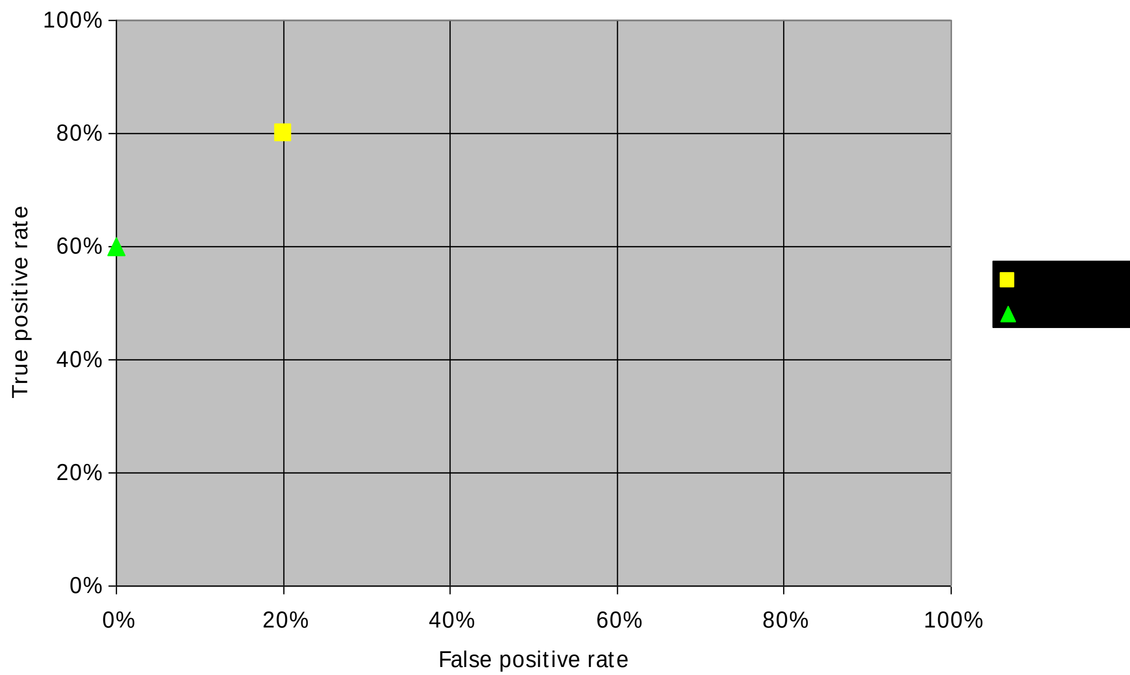
- ***True positive rate*** = #true pos. / #pos.
  - TPr_1 = 40/50 = 80%
  - TPr_2 = 30/50 = 60%
- ***False positive rate*** = #false pos. / #neg.
  - FPr_1 = 10/50 = 20%
  - FPr_2 = 0/50 = 0%
- ***ROC space*** has
  - FPr on X axis
  - TPr on Y axis

Classifier 1

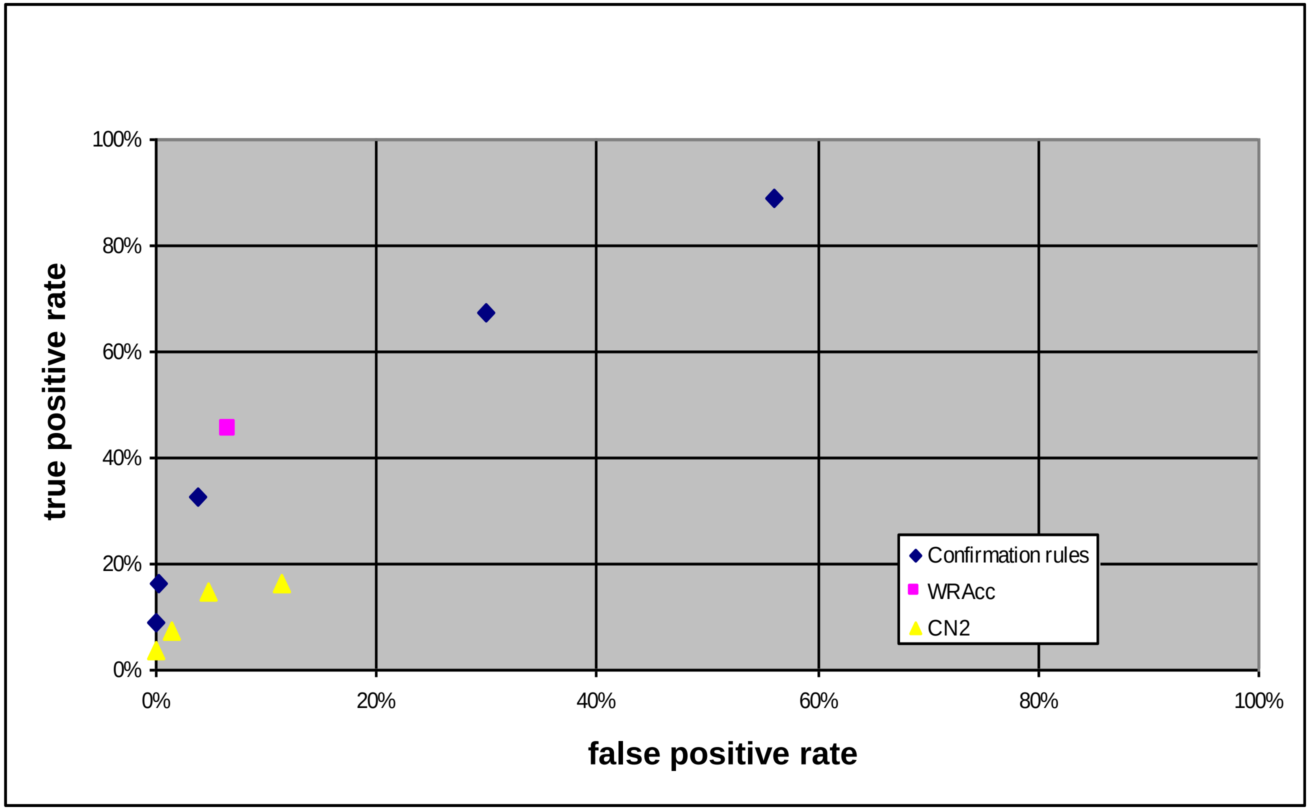
	Predicted positive	Predicted negative	
Positive examples	**40**	**10**	50
Negative examples	**10**	**40**	50
	50	50	100

Classifier 2

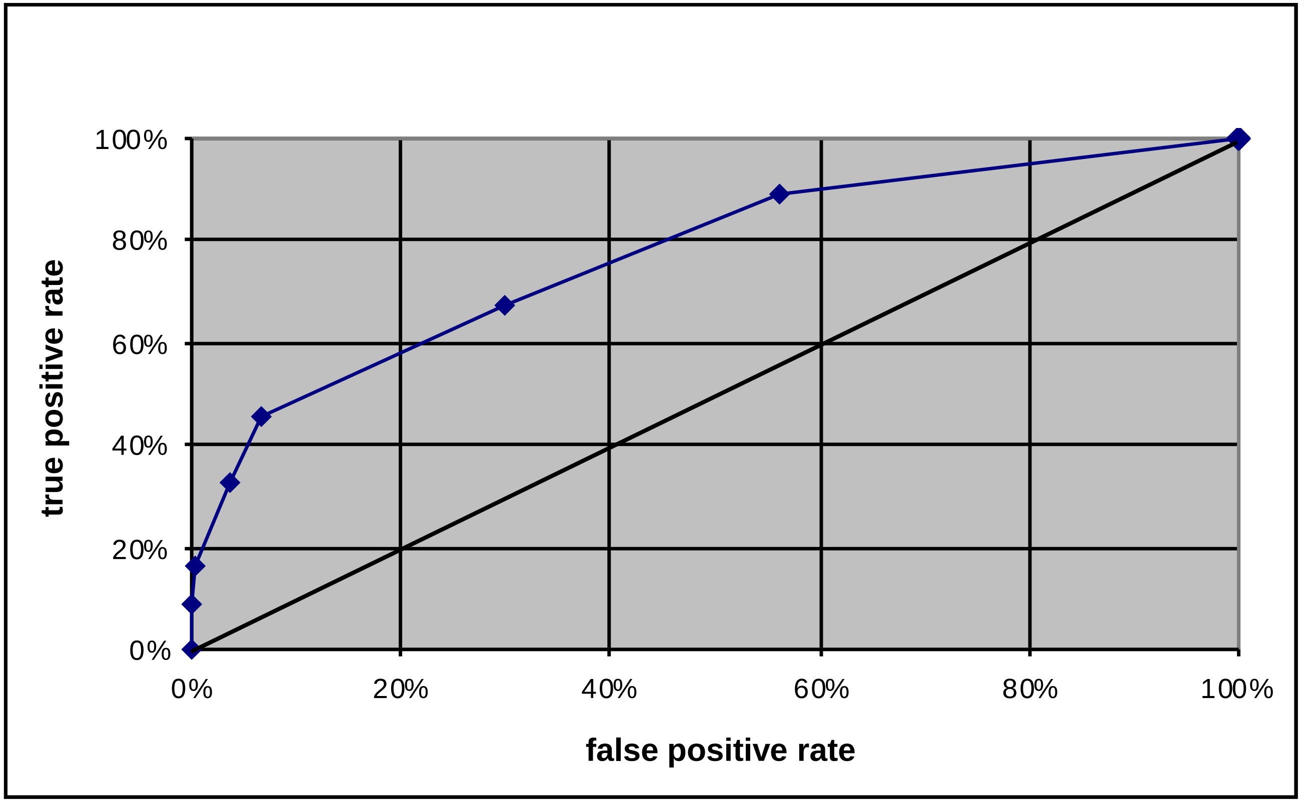
	Predicted positive	Predicted negative	
Positive examples	**30**	**20**	50
Negative examples	**0**	**50**	50
	30	70	100



The ROC space



The ROC convex hull



Course Outline

I. Introduction

- Data Mining and KDD process
- Introduction to Data Mining
- Data Mining platforms

II. Predictive DM

- Decision Tree learning
- Bayesian classifier
- Classification rule learning
- Classifier evaluation

III. Predictive DM

- Regression

IV. Descriptive DM

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

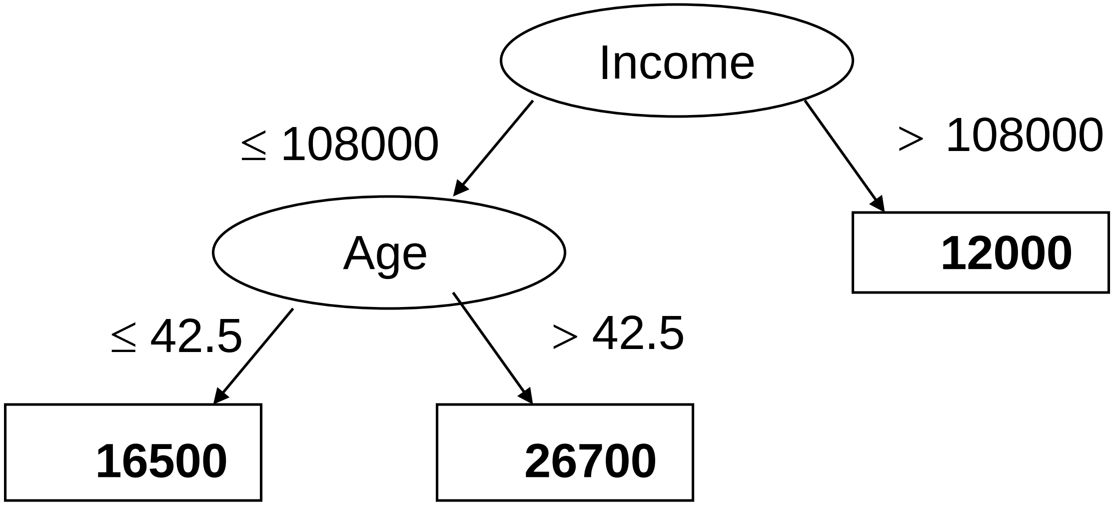
III. Predictive DM – Regression

- Estimation or regression task: given objects described with attribute values, induce a model to predict the numeric class value
- Data are objects, characterized with attributes (discrete or continuous), classes of objects are continuous (numeric)
- Regression trees, linear and logistic regression, ANN, kNN, ...
- Regression tree learners, model tree learners:
  - M5, M5P (implemented in WEKA), Tree (in Orange)

Estimation/regression example: Customer data

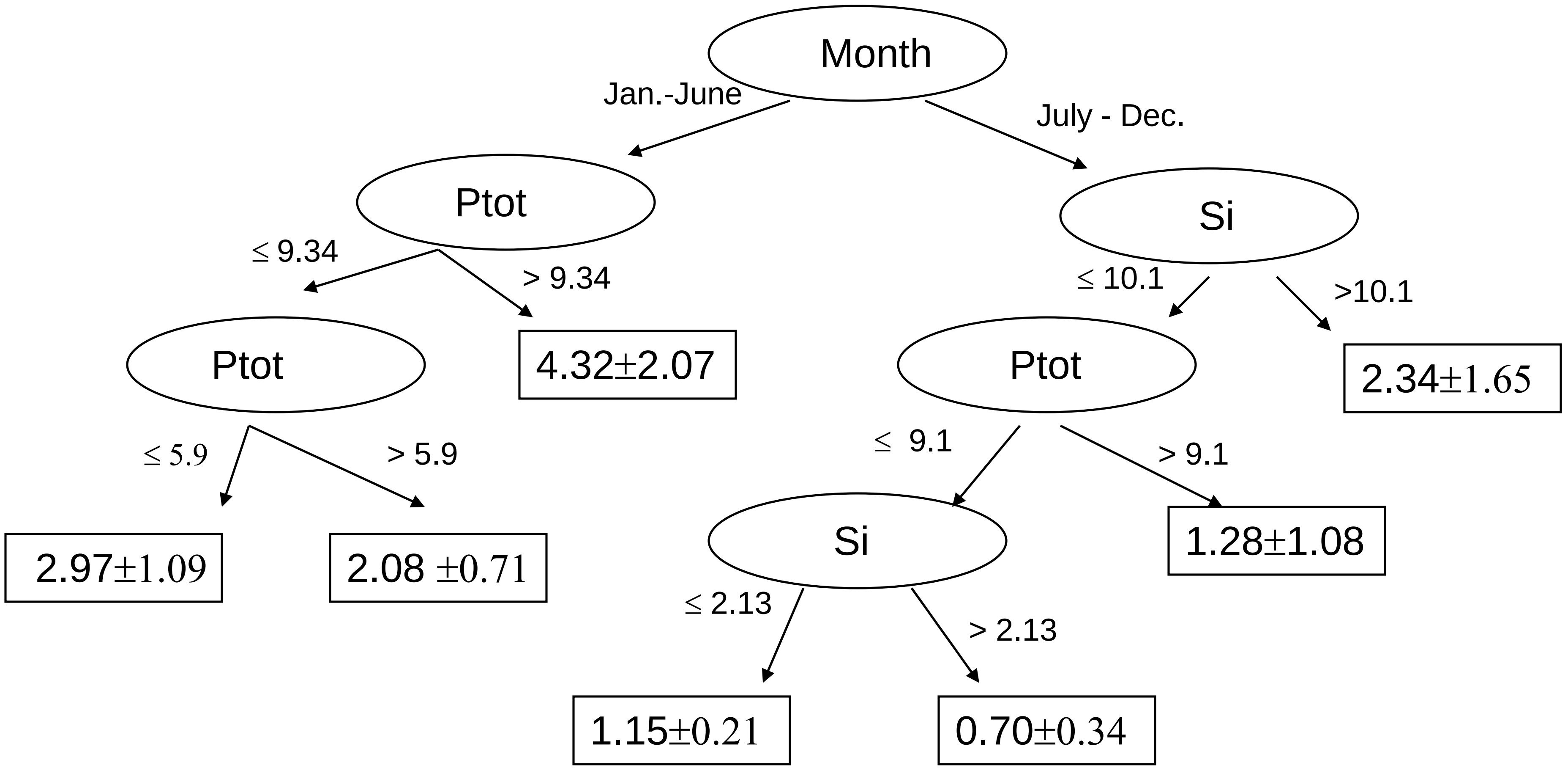
Customer	Gender	Age	Income	Spent	
c1	male	30	214000	18800	
c2	female	19	139000	15100	
c3	male	55	50000	12400	
c4	female	48	26000	8600	
c5	male	63	191000	28100	
O6-O13	...	...	...	...	
c14	female	61	95000	18100	
c15	male	56	44000	12000	
c16	male	36	102000	13800	
c17	female	57	215000	29300	
c18	male	33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

Customer data: regression tree



In the nodes one usually has Predicted value +- st. deviation

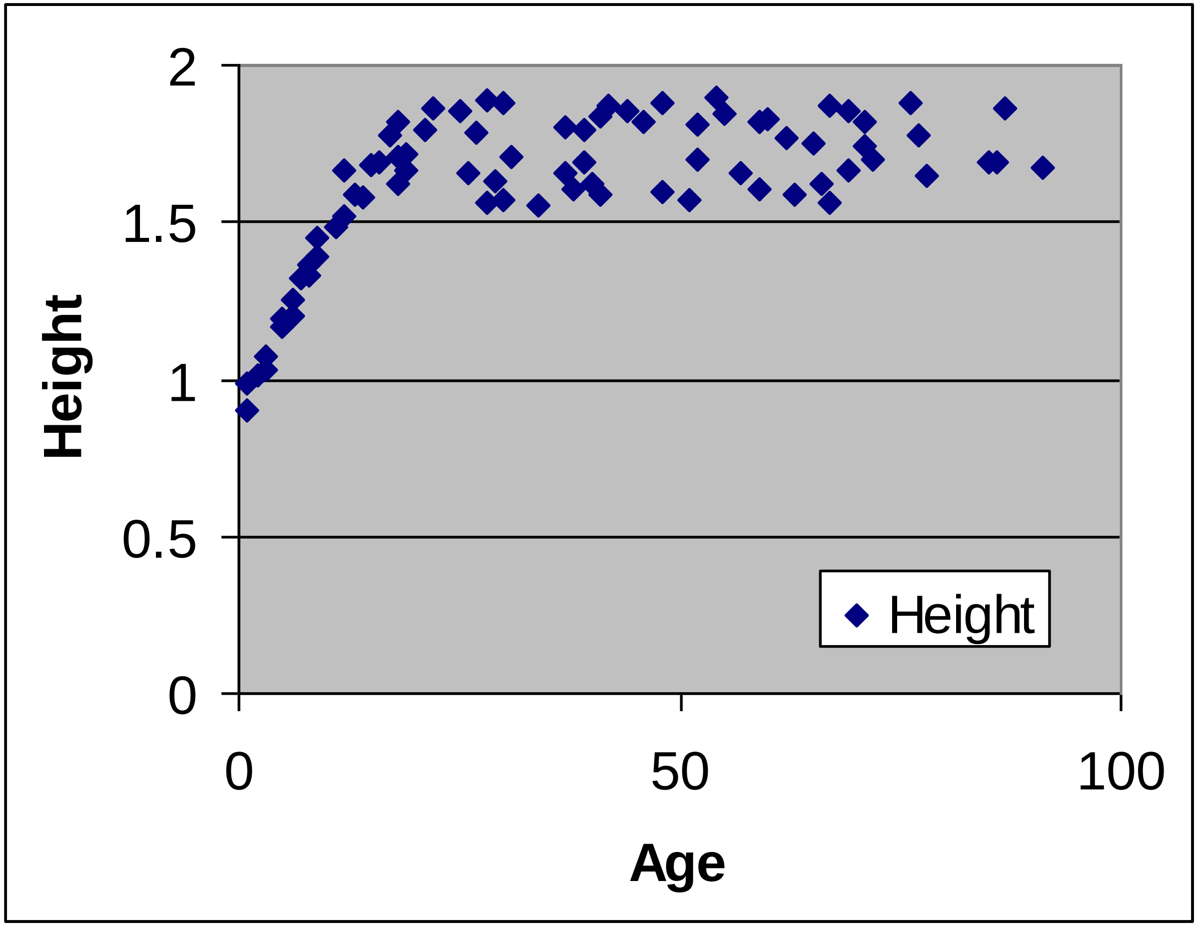
Predicting algal biomass: regression tree



Regression	Classification
Data: attribute-value description	
Target variable: Continuous	**Target variable**: Categorical (nominal)
Evaluation: cross validation, separate test set, …	
Error: MSE, MAE, RMSE, …	**Error**: 1-accuracy
Algorithms: Linear regression, regression trees,…	**Algorithms**: Decision trees, Naïve Bayes, …
Baseline predictor: Mean of the target variable	**Baseline predictor**: Majority class

Example regression problem

- data about 80 people: Age and Height



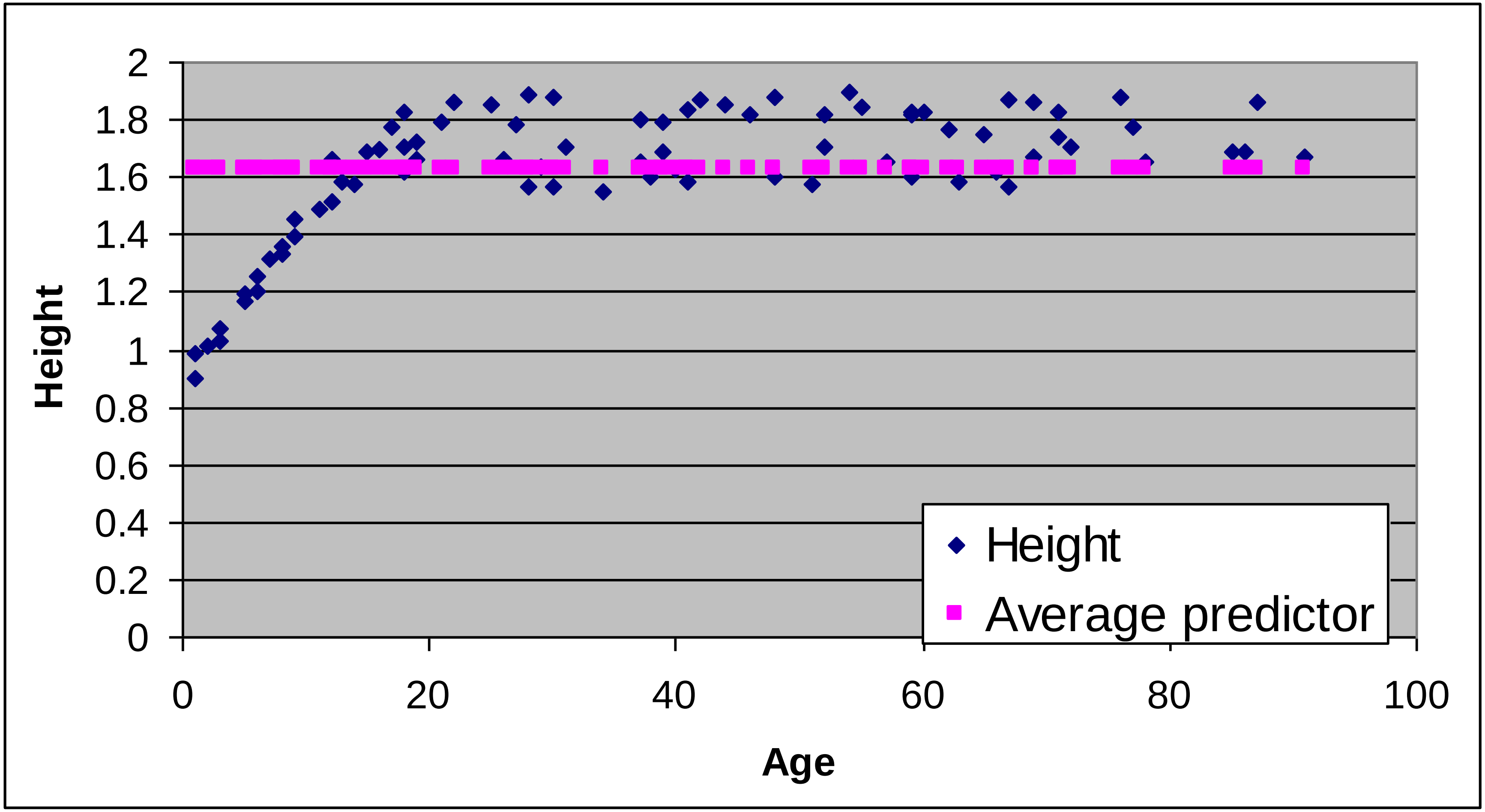
Age	Height
3	1.03
5	1.19
6	1.26
9	1.39
15	1.69
19	1.67
22	1.86
25	1.85
41	1.59
48	1.60
54	1.90
71	1.82
...	...

Test set

Age	Height
2	0.85
10	1.4
35	1.7
70	1.6

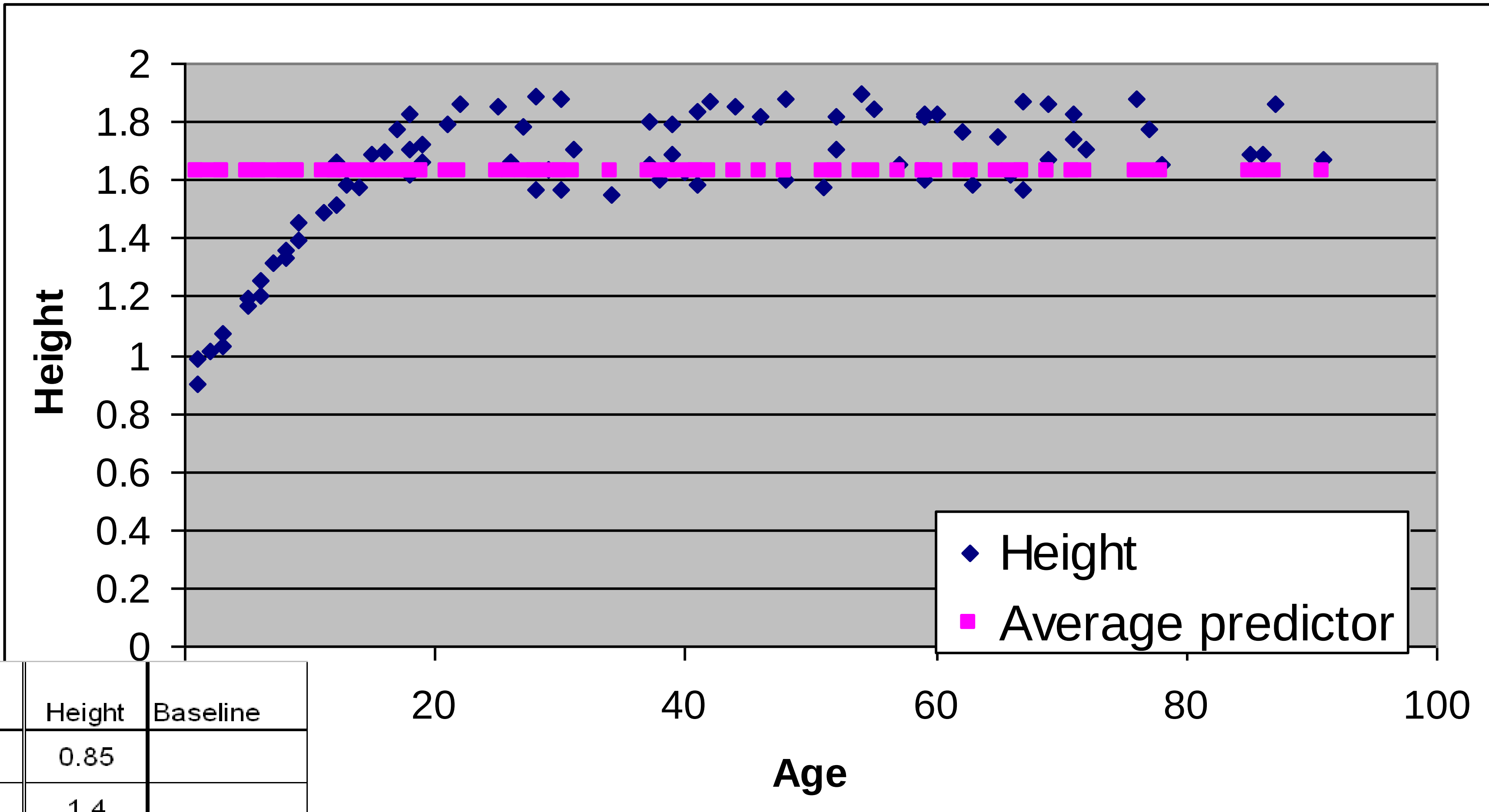
Baseline numeric model

- Average of the target variable



Baseline numeric predictor

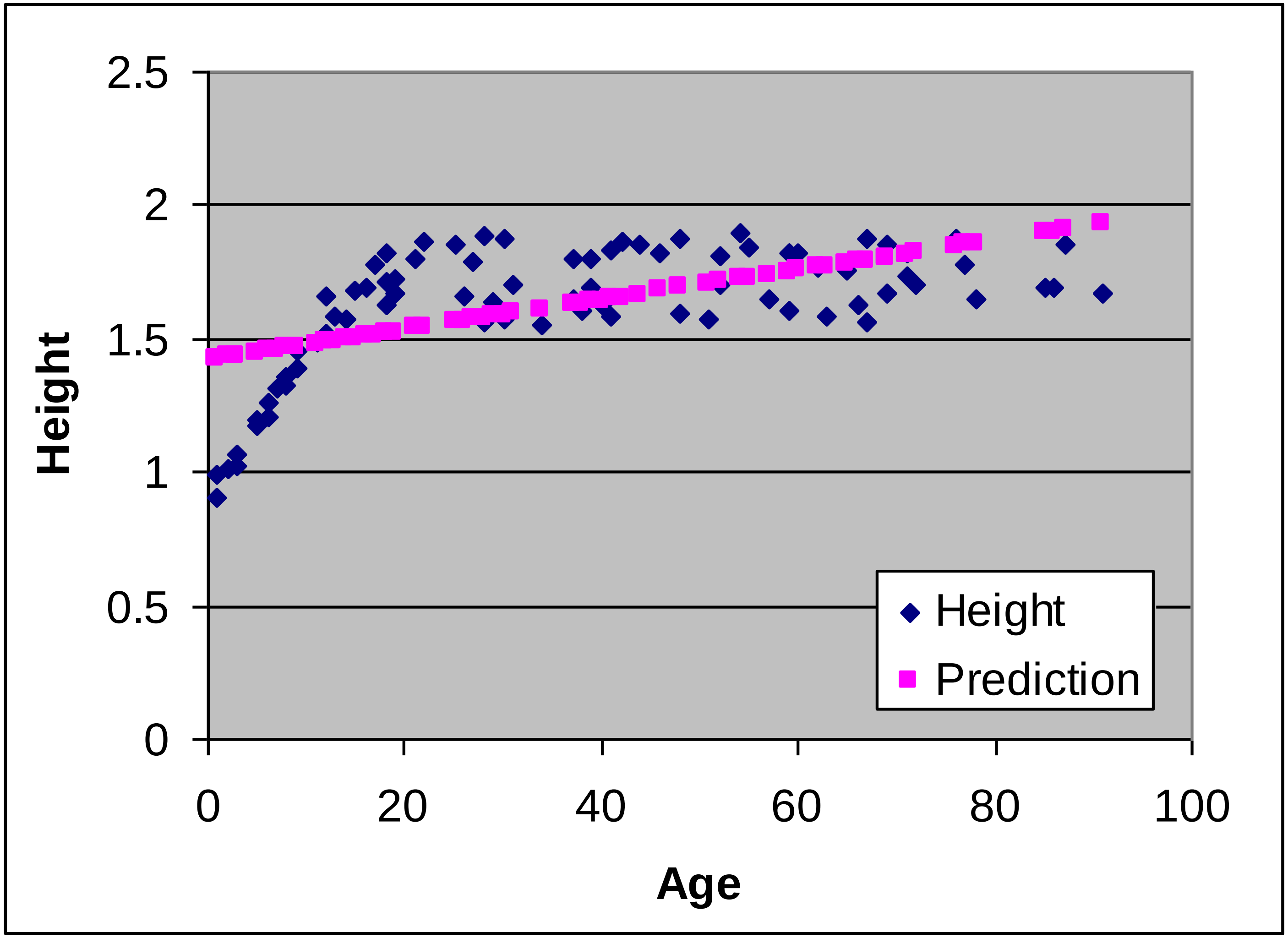
- Average of the target variable is 1.63



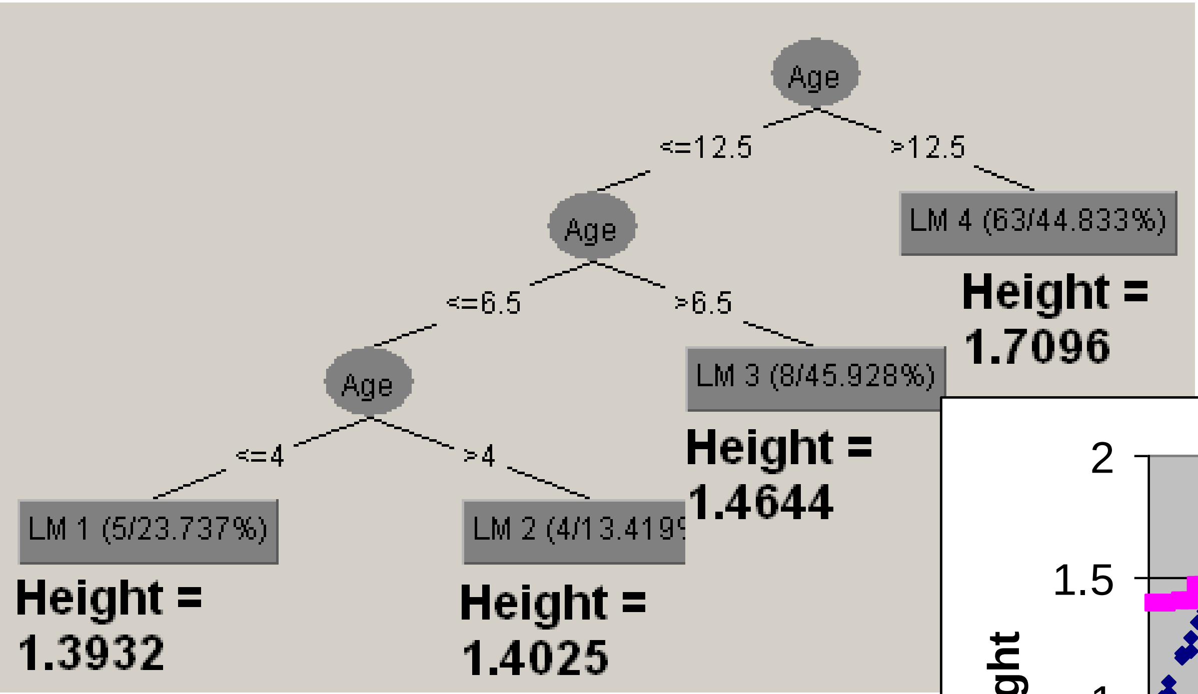
Age	Height	Baseline
2	0.85	
10	1.4	
35	1.7	
70	1.6	

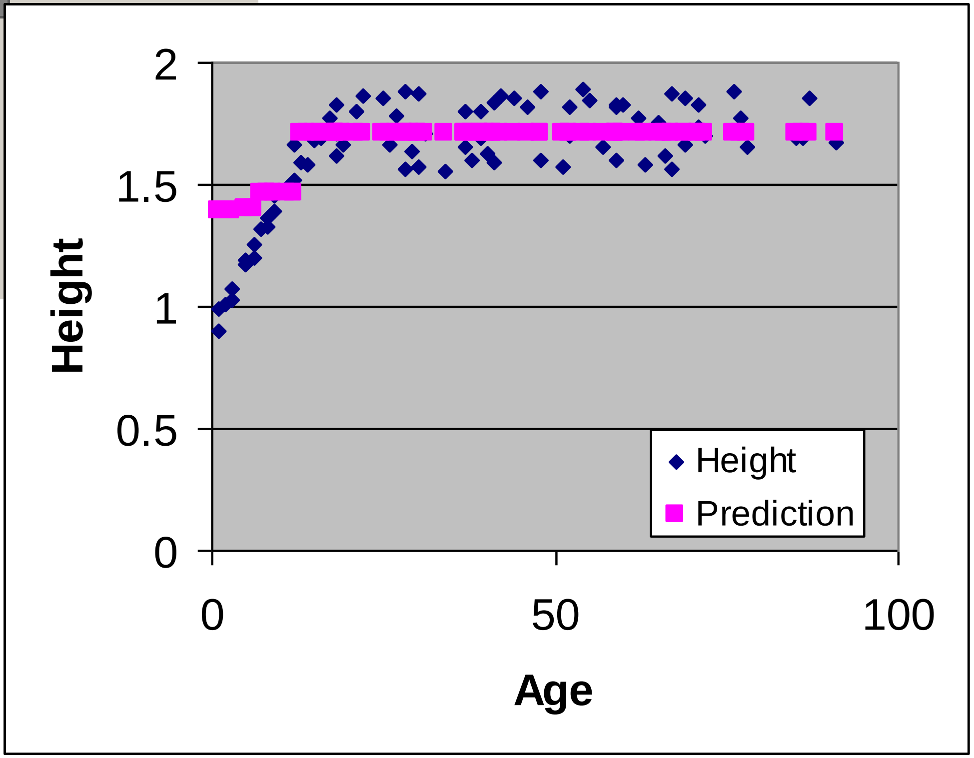
Linear Regression Model

Height = 0.0056 * Age + 1.4181

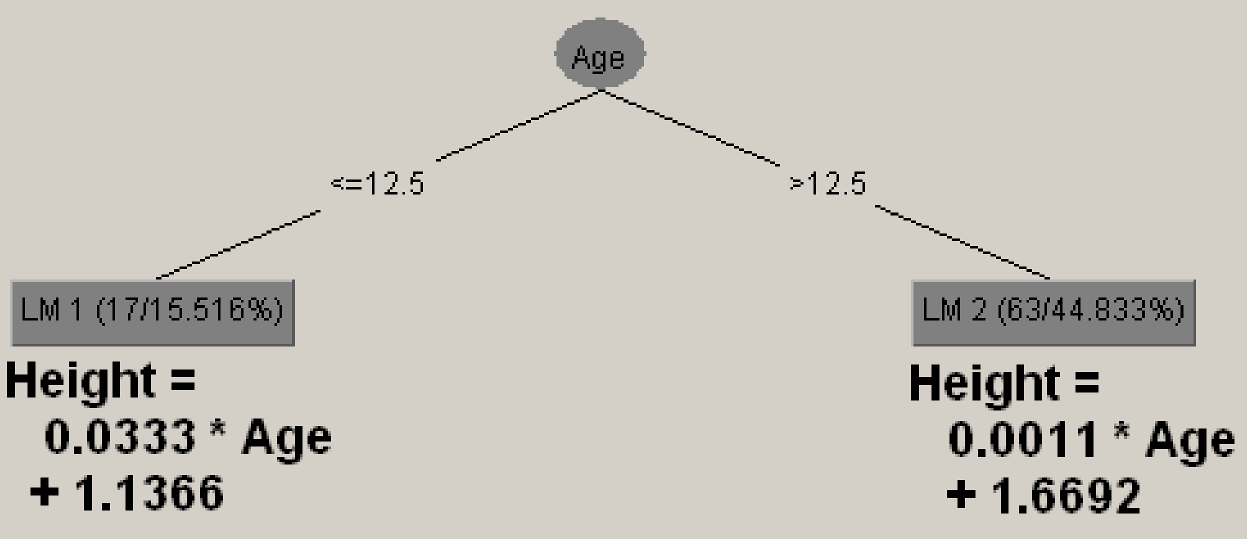


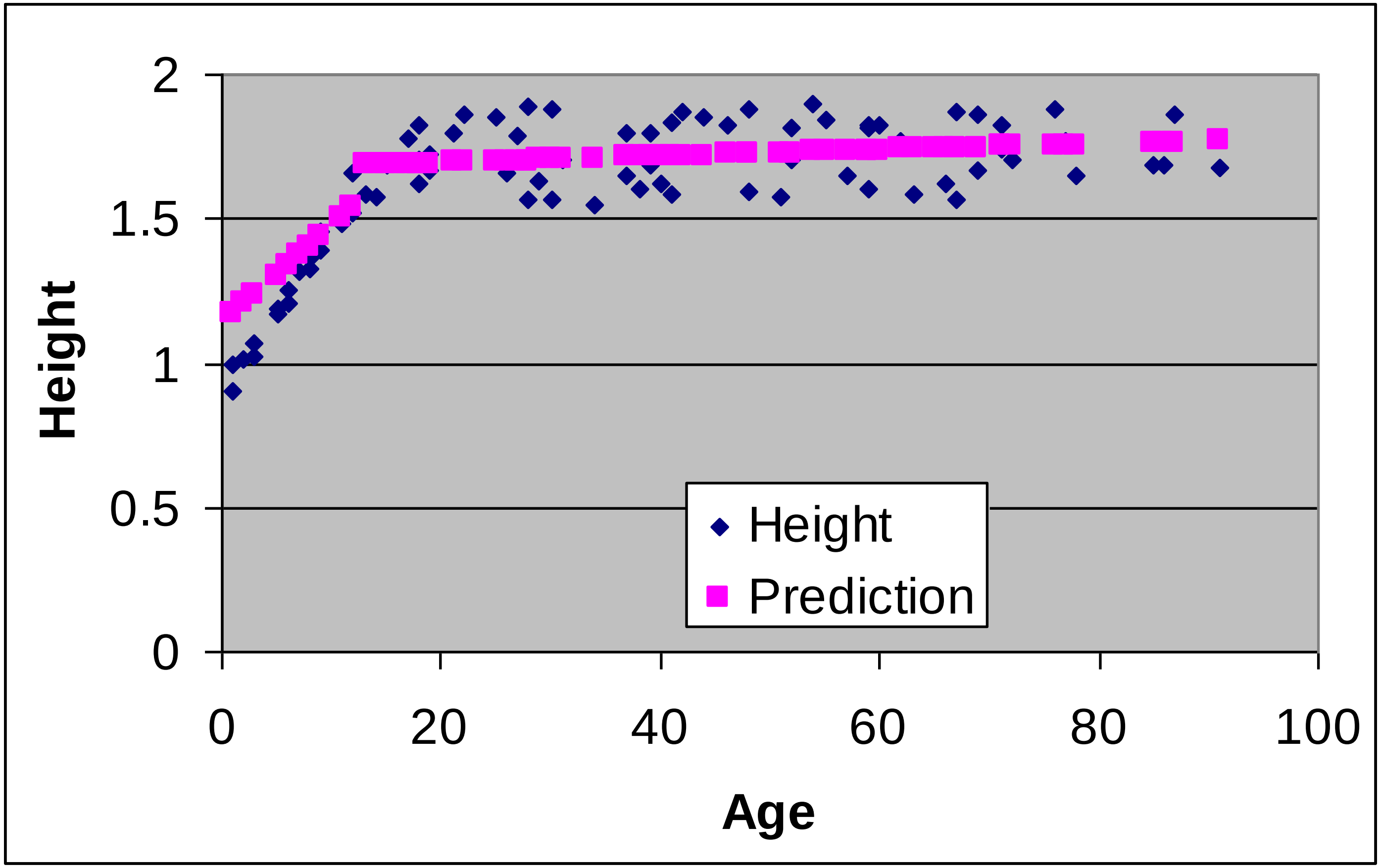
Regression tree





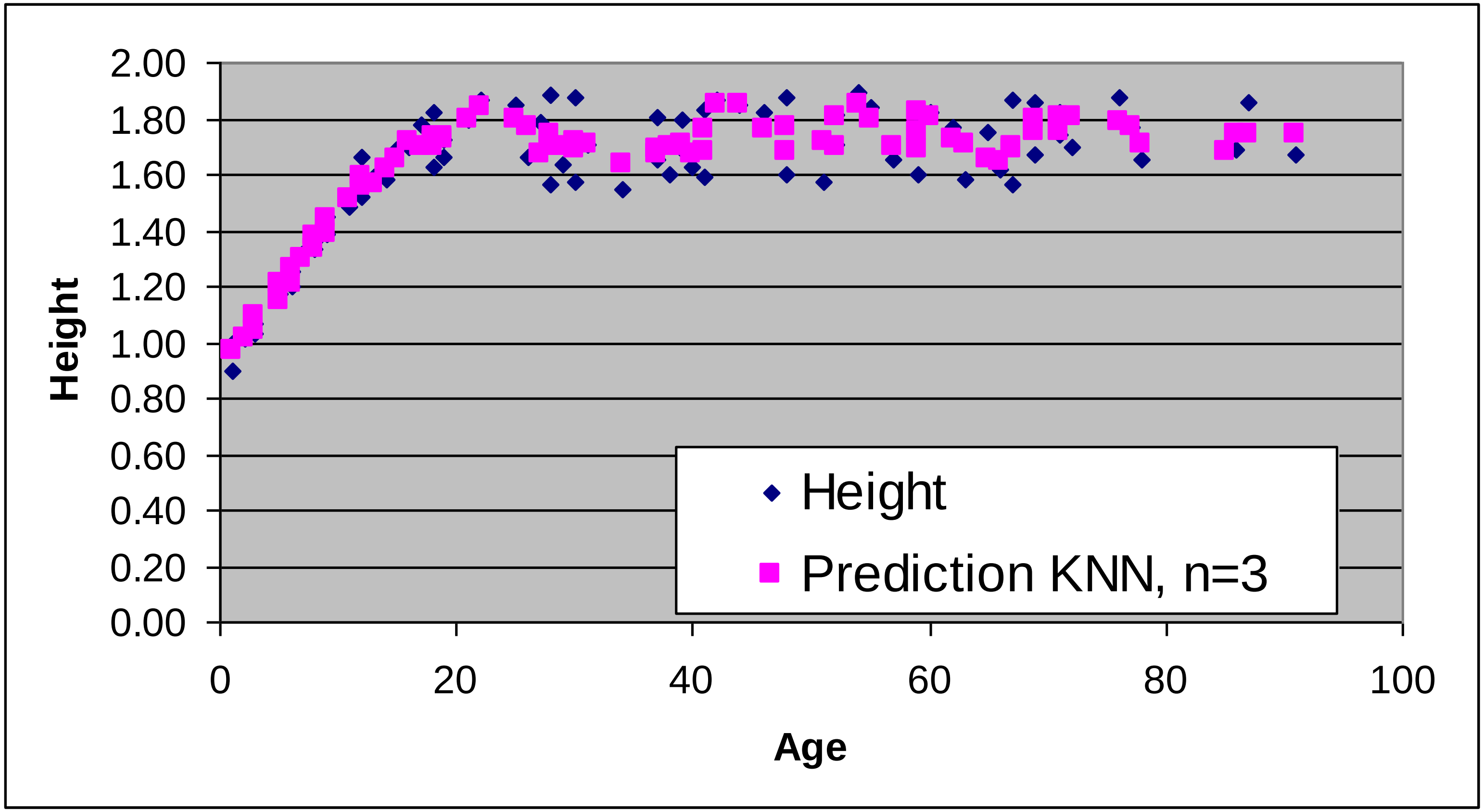
Model tree


Age
<=12.5
>12.5
LM 1 (17/15.516%)
LM 2 (63/44.833%)
Height =
0.0333 * Age
+ 1.1366
Height =
0.0011 * Age
+ 1.6692


2
1.5
1
0.5
0
0
20
40
60
80
100
Height
Age
Height
Prediction

kNN – K nearest neighbors

- Looks at K closest examples (by age) and predicts the average of their target variable
- K=3



Which predictor is the best?

Age	Height	Baseline	Linear regression	Regression tree	Model tree	kNN
2	0.85	1.63	1.43	1.39	1.20	1.01
10	1.4	1.63	1.47	1.46	1.47	1.51
35	1.7	1.63	1.61	1.71	1.71	1.67
70	1.6	1.63	1.81	1.71	1.75	1.81

Course Outline

I. Introduction
- Data Mining and KDD process
- Introduction to Data Mining
- Data Mining platforms

II. Predictive DM
- Decision Tree learning
- Bayesian classifier
- Classification rule learning
- Classifier evaluation

III. Predictive DM
- Regression

IV. Descriptive DM
- Predictive vs. descriptive induction
- Subgroup discovery
- Association rule learning
- Hierarchical clustering

Part IV. Descriptive DM techniques

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

Descriptive DM: Subgroup discovery 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

Customer data: Subgroup discovery

Type of task: description (pattern discovery)

Hypothesis language: rules **X ➔ Y,** if X then Y

X is conjunctions of items, Y is target class

Age > 52 & Sex = male ➔ BigSpender = no

Age > 52 & Sex = male & Income ≤ 73250

➔ BigSpender = no

Descriptive DM: Association rule learning 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

Customer data: Association rules

Type of task: description (pattern discovery)

Hypothesis language: rules **X ➔ Y,** if X then Y

X, Y conjunctions of items

1. Age > 52 & BigSpender = no ➔ Sex = male
2. Age > 52 & BigSpender = no ➔
   Sex = male & Income ≤ 73250
3. Sex = male & Age > 52 & Income ≤ 73250 ➔
   BigSpender = no

Descriptive DM: Clustering and association rule learning 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

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

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

Predictive vs. descriptive DM: Summary from a rule learning perspective

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

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

Descriptive DM

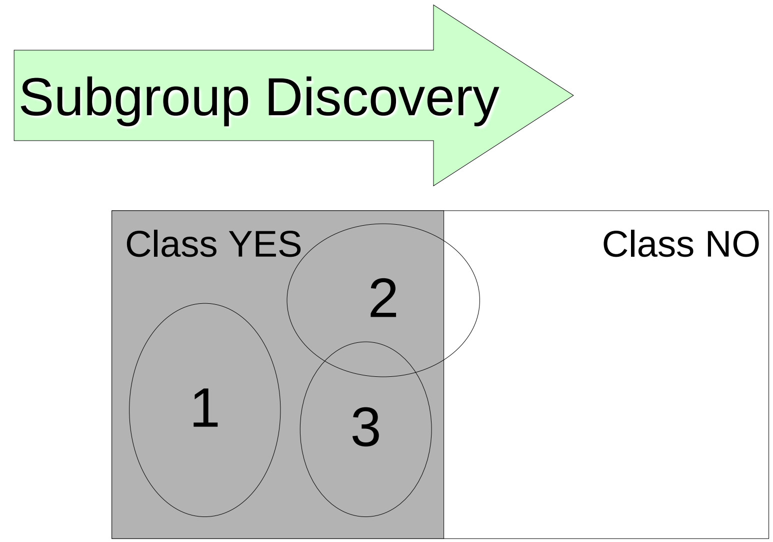
- **Description**
  - Data description and summarization: describe elementary and aggregated data characteristics (statistics, …)
  - Dependency analysis:
    - describe associations, dependencies, …
    - discovery of properties and constraints
- **Segmentation**
  - Clustering: separate objects into 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)

Part IV. Descriptive DM techniques

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

Subgroup Discovery

Person	Age	Spect. presc.	Astigm.	Tear prod.	Lenses
O1	17	myope	no	reduced	NO
O2	23	myope	no	normal	YES
O3	22	myope	yes	reduced	NO
O4	27	myope	yes	normal	YES
O5	19	hypermetrope	no	reduced	NO
O6-O13	...	...	...	...	...
O14	35	hypermetrope	no	normal	YES
O15	43	hypermetrope	yes	reduced	NO
O16	39	hypermetrope	yes	normal	NO
O17	54	myope	no	reduced	NO
O18	62	myope	no	normal	NO
O19-O23	...	...	...	...	...
O24	56	hypermetrope	yes	normal	NO

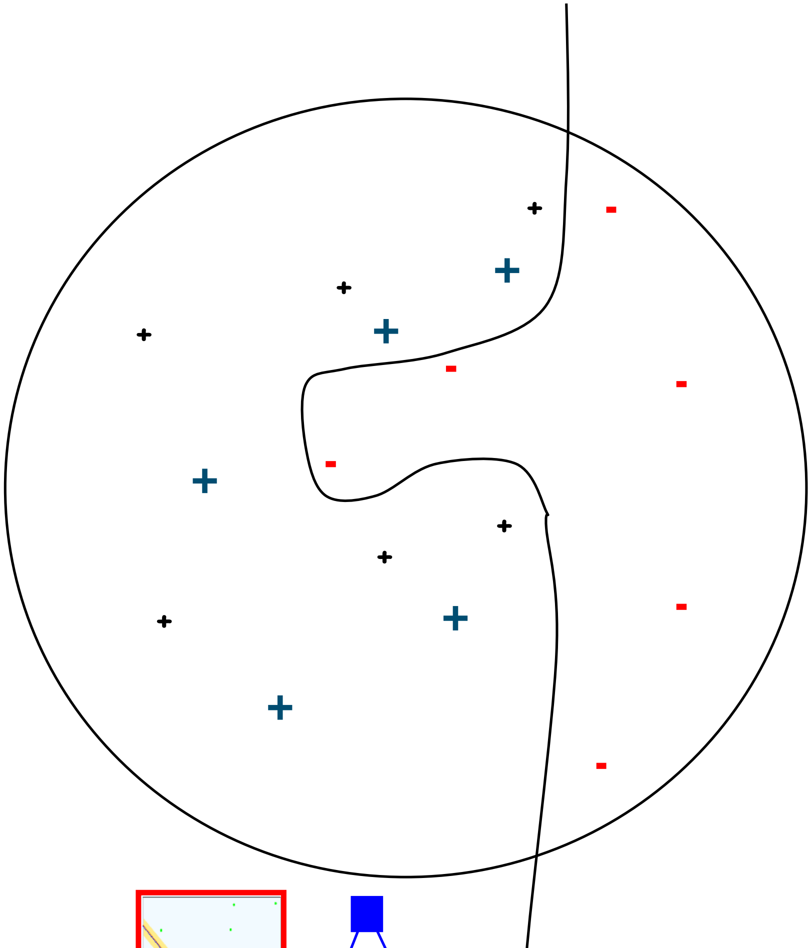


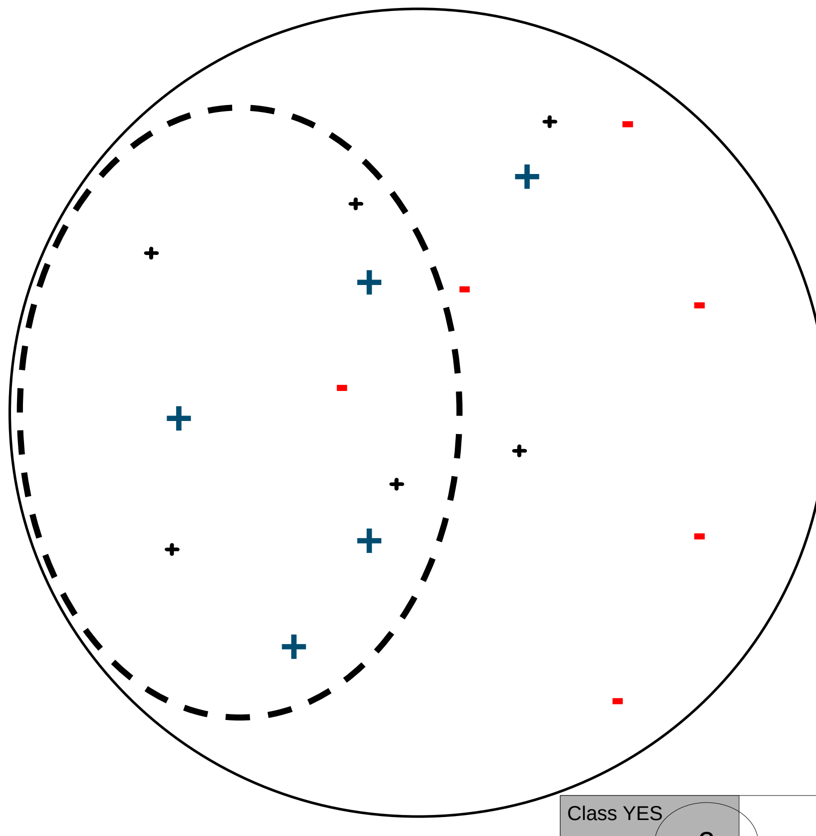
- A task in which individual interpretable patterns in the form of rules are induced from data, labeled by a predefined property of interest.
- SD algorithms learn several independent rules that describe groups of target class examples
  - subgroups must be large and significant

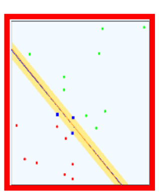
Classification versus Subgroup Discovery

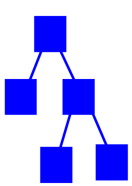
- **Classification (predictive induction) - constructing sets of classification rules**
  - aimed at learning a model for classification or prediction
  - rules are dependent
- **Subgroup discovery (descriptive induction) – constructing individual subgroup describing rules**
  - aimed at finding interesting patterns in target class examples
    - large subgroups (high target class coverage)
    - with significantly different distribution of target class examples (high TP/FP ratio, high significance, high WRAcc
  - each rule (pattern) is an independent chunk of knowledge

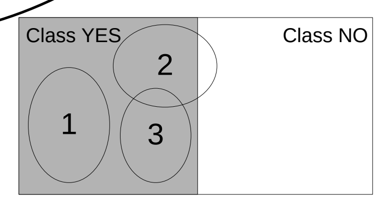
Classification versus Subgroup discovery











Subgroup discovery

Education	Marital Status	Sex	Has Children	Approved
primary	single	male	no	no
primary	single	male	yes	no
primary	married	male	no	yes
university	divorced	female	no	yes
university	married	female	yes	yes
secondary	single	male	no	no
university	single	female	no	yes
secondary	divorced	female	no	yes
secondary	single	female	yes	yes
secondary	married	male	yes	yes
primary	married	female	no	yes
secondary	divorced	male	yes	no
university	divorced	female	yes	no
secondary	divorced	male	no	yes

Approved = yes ← Sex = female
Approved = yes ← Marital status = married
Approved = yes ← Marital status = divorced & Has children = no
Approved = yes ← Education = university

Selected rules discovered by Apriori-SD subgroup discovery algorithm.

Subgroup discovery in High CHD Risk Group Detection

Input: Patient records described by anamnestic, laboratory and ECG attributes

Task: Find and characterize population subgroups with high CHD risk (large enough, distributionaly unusual)

From **best induced descriptions**, five were selected by the expert as **most actionable** for CHD risk screening (by GPs):

high-CHD-risk ← male & pos. fam. history & age > 46

high-CHD-risk ← female & bodymassIndex > 25 & age > 63

high-CHD-risk ← ...

high-CHD-risk ← ...

high-CHD-risk ← ...

(Gamberger & Lavrač, JAIR 2002)

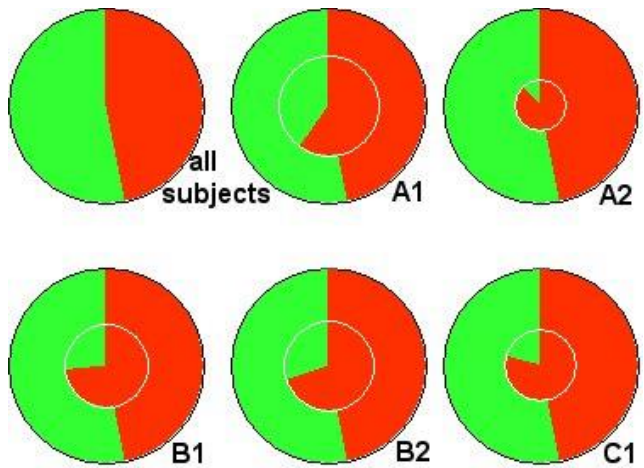
Subgroup Discovery: Medical Use Case

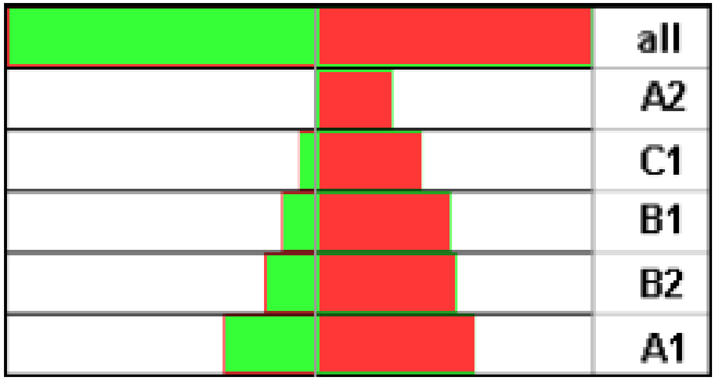
- **Find and characterize population subgroups with high risk for coronary heart disease (CHD)** (Gamberger, Lavrač, Krstačić)
- **A1** for males: **principal risk factors**

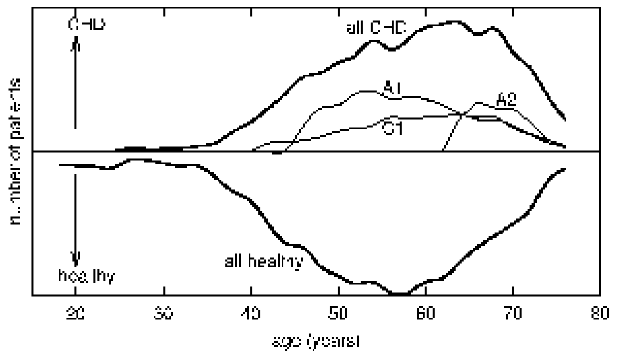
  CHD ← pos. fam. history & age > 46
- **A2** for females: **principal risk factors**

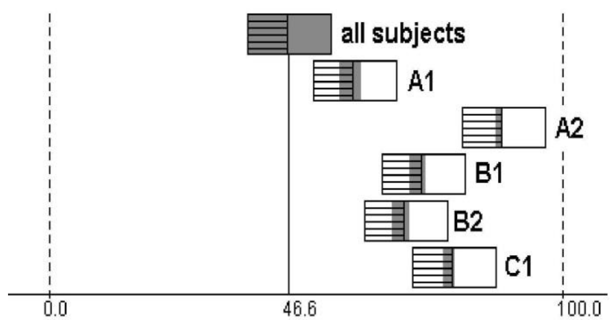
  CHD ← bodyMassIndex > 25 & age >63
- **A1, A2** (anamnestic info only), **B1, B2** (an. and physical examination), **C1** (an., phy. and ECG)
- **A1: supporting factors** (found by statistical analysis): psychosocial stress, as well as cigarette smoking, hypertension and overweight

Subgroup visualization









The CHD task: Find, characterize and visualize population subgroups with high CHD risk (large enough, distributionally unusual, most actionable)

Subgroup discovery in functional genomics

- Functional genomics is a typical scientific discovery domain, studying genes and their functions
- Very large number of attributes (genes)
- Interesting subgroup describing patterns discovered by SD algorithm

CancerType = Leukemia	
IF	KIAA0128 = DIFF. EXPRESSED
AND	prostoglandin d2 synthase = NOT_ DIFF. EXPRESSED

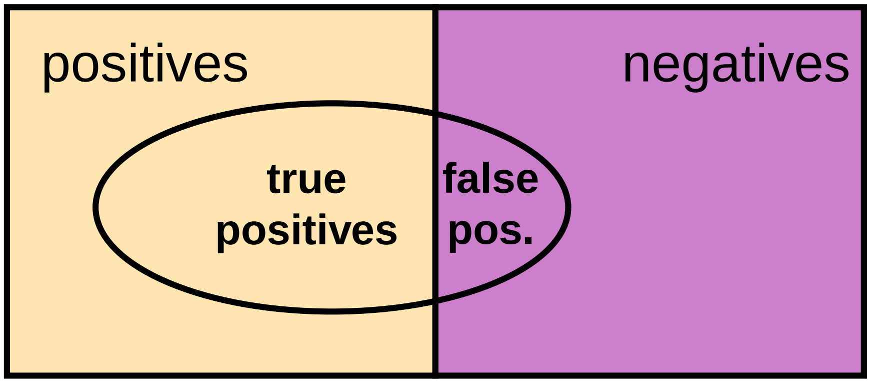
- Interpretable by biologists

D. Gamberger, N. Lavrač, F. Železný, J. Tolar

Journal of Biomedical Informatics 37(5):269-284, 2004

Subgroups vs. classifiers

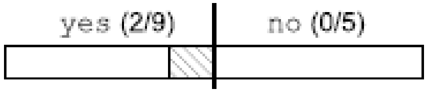
- Classifiers:
  - Classification rules aim at pure subgroups
  - A set of rules forms a domain model
- Subgroups:
  - Rules describing subgroups aim at significantly higher proportion of positives
  - Each rule is an independent chunk of knowledge
- Link
  - SD can be viewed as cost-sensitive classification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



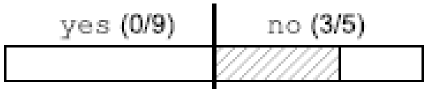
Recall: Survey data Classification rule learning

Education	Marital Status	Sex	Has Children	Approved
primary	single	male	no	no
primary	single	male	yes	no
primary	married	male	no	yes
university	divorced	female	no	yes
university	married	female	yes	yes
secondary	single	male	no	no
university	single	female	no	yes
secondary	divorced	female	no	yes
secondary	single	female	yes	yes
secondary	married	male	yes	yes
primary	married	female	no	yes
secondary	divorced	male	yes	no
university	divorced	female	yes	no
secondary	divorced	male	no	yes

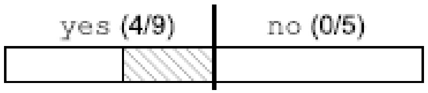
```
IF   MaritalStatus = single
 AND Sex = female
THEN Approved = yes
```



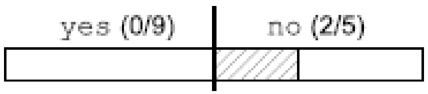

```
IF   MaritalStatus = single
 AND Sex = male
THEN Approved = no
```



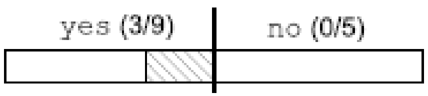

```
IF   MaritalStatus = married
THEN Approved = yes
```




```
IF   MaritalStatus = divorced
 AND HasChildren = yes
THEN Approved = no
```



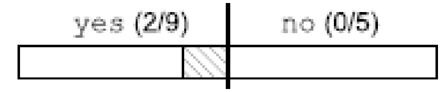

```
IF   MaritalStatus = divorced
 AND HasChildren = no
THEN Approved = yes
```



Survey data
Subgroup discovery

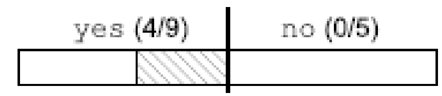
Education	Marital Status	Sex	Has Children	Approved
primary	single	male	no	no
primary	single	male	yes	no
primary	married	male	no	yes
university	divorced	female	no	yes
university	married	female	yes	yes
secondary	single	male	no	no
university	single	female	no	yes
secondary	divorced	female	no	yes
secondary	single	female	yes	yes
secondary	married	male	yes	yes
primary	married	female	no	yes
secondary	divorced	male	yes	no
university	divorced	female	yes	no
secondary	divorced	male	no	yes

```
IF   MaritalStatus = single
 AND Sex = female
THEN Approved = yes
```

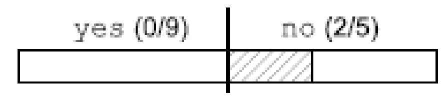



```
IF   MaritalStatus = single
 AND Sex = male
THEN Approved = no
```

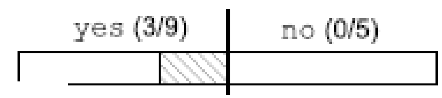

```
IF   MaritalStatus = married
THEN Approved = yes
```



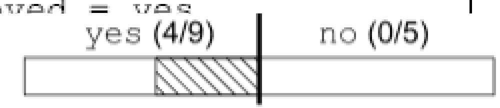

```
IF   MaritalStatus = divorced
 AND HasChildren = yes
THEN Approved = no
```



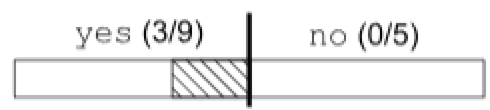

```
IF   MaritalStatus = divorced
 AND HasChildren = no
THEN Approved = yes
```



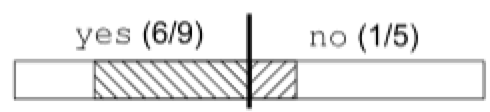
IF MaritalStatus = married
THEN Approved = yes



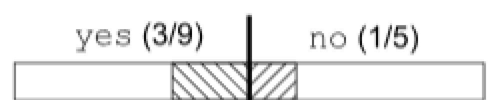
IF MaritalStatus = divorced
AND HasChildren = no
THEN Approved = yes




```
IF   Sex = female
THEN Approved = yes
```




```
IF   Education = university
THEN Approved = yes
```



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

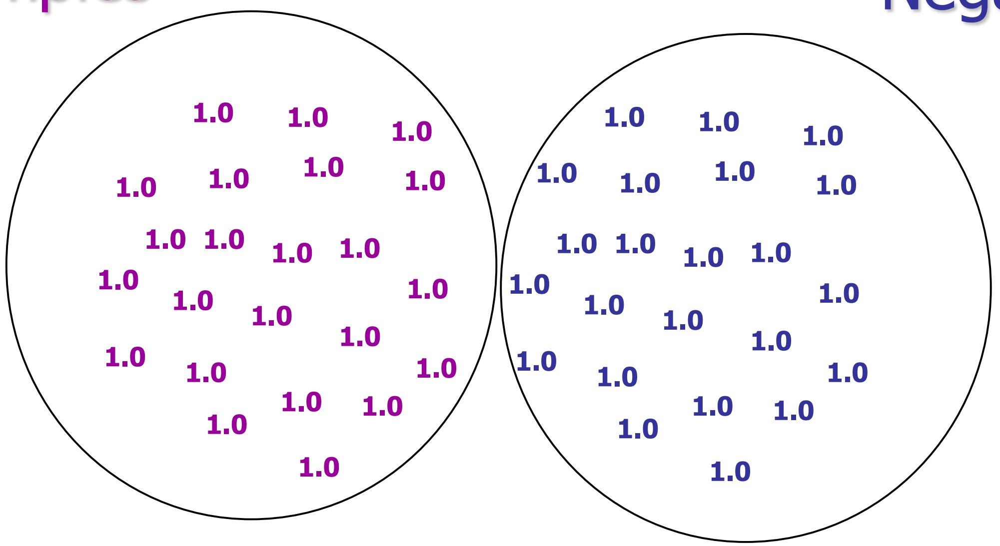
- Standard covering approach:
  covered examples are deleted from current training set

- 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

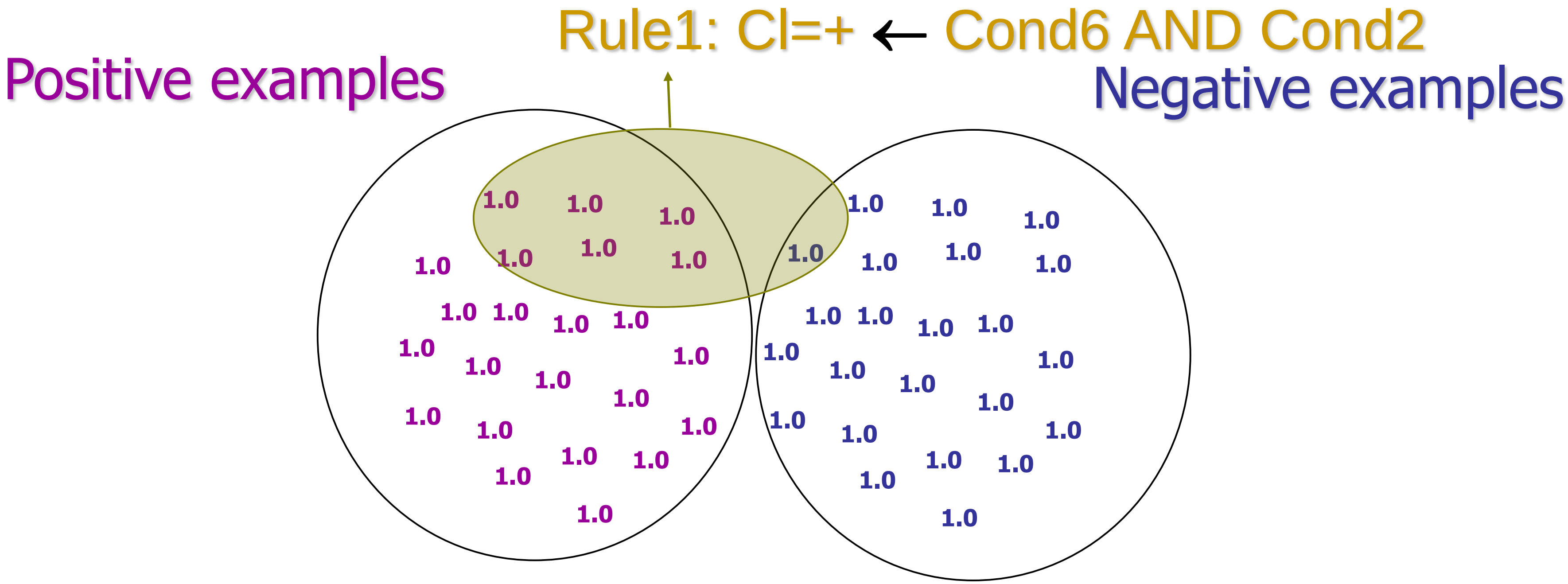
Subgroup Discovery

Positive examples

Negative examples



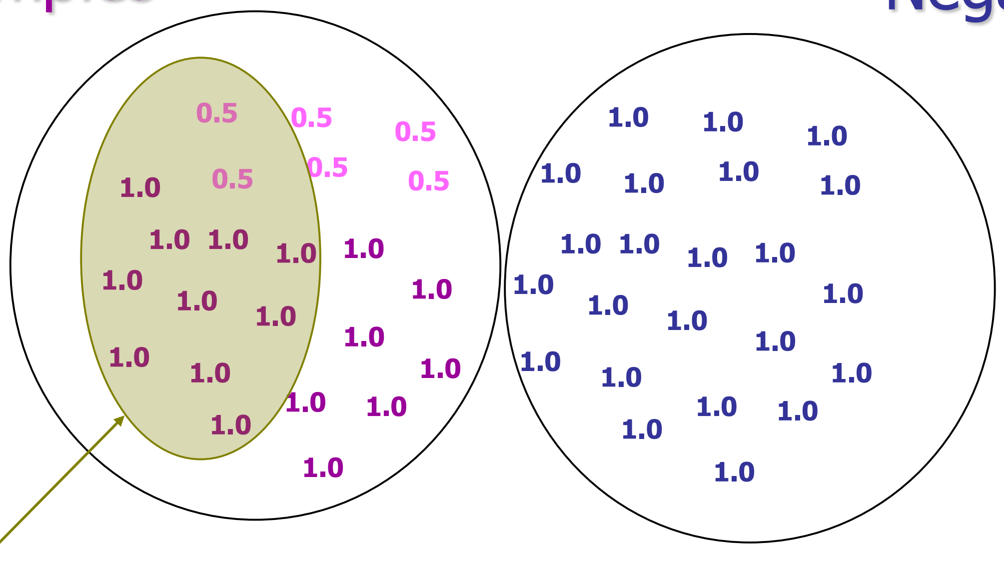
Subgroup Discovery



Subgroup Discovery

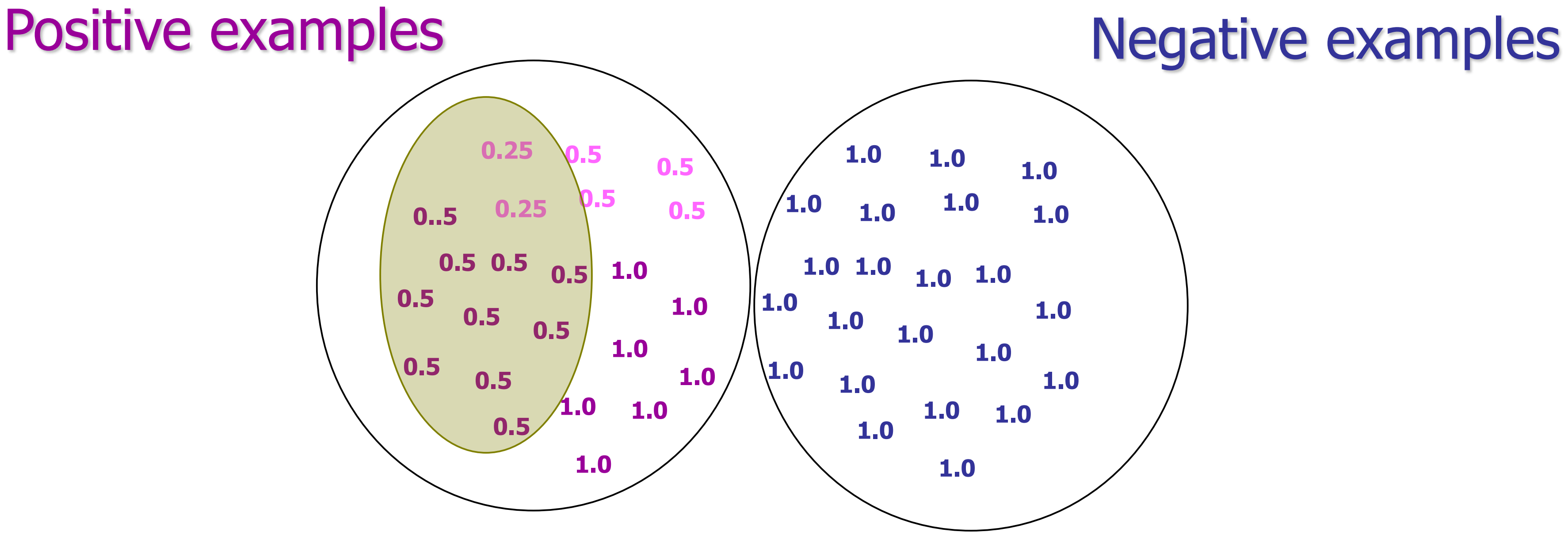
Positive examples

Negative examples



Rule2: Cl=+ ← Cond3 AND Cond4

Subgroup Discovery



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:

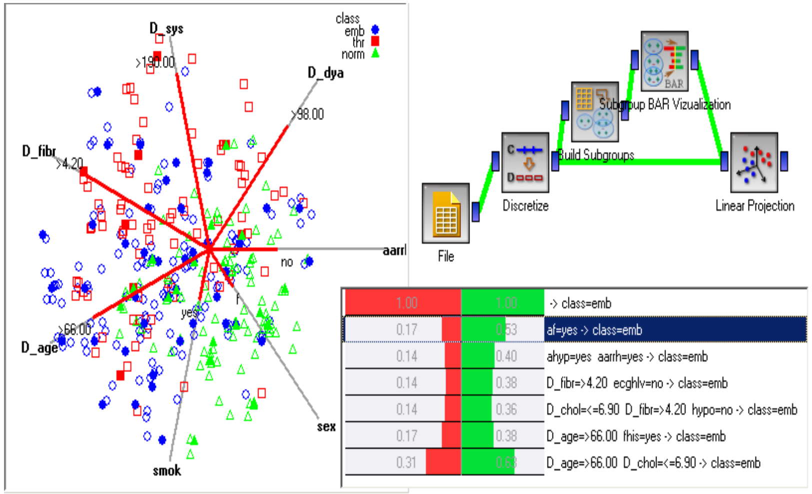
  WRAcc(Cl $\leftarrow$ Cond) = p(Cond) (p(Cl|Cond) - p(Cl)) =

  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

SD algorithms in the Orange DM Platform

- **Orange** data mining toolkit
  - classification and subgroup discovery algorithms
  - data mining workflows
  - visualization



- **SD Algorithms in Orange**
  - SD (Gamberger & Lavrač, JAIR 2002)
  - Apriori-SD (Kavšek & Lavrač, AAI 2006)
  - CN2-SD (Lavrač et al., JMLR 2004): Adapting CN2 classification rule learner to Subgroup Discovery

Part IV. Descriptive DM techniques

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

Association Rule Learning

Rules: X =>Y, if X then Y

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

Given: Transactions

itemsets (records)

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

Find: A set of association rules in the form X =>Y

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)

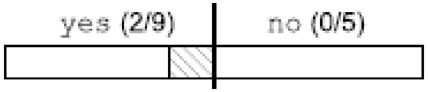
Association Rule Learning: Examples

- Market basket analysis
  - beer & coke ⇒ peanuts & chips (5%, 65%)
    (IF beer AND coke THEN peanuts AND chips)
  - Support 5%: 5% of all customers buy all four items
  - Confidence 65%: 65% of customers that buy beer and coke also buy peanuts and chips
- Insurance
  - mortgage & loans & savings ⇒ insurance (2%, 62%)
  - Support 2%: 2% of all customers have all four
  - Confidence 62%: 62% of all customers that have mortgage, loan and savings also have insurance

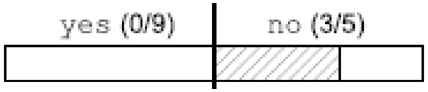
Recall: Survey data
Classification rule learning

Education	Marital Status	Sex	Has Children	Approved
primary	single	male	no	no
primary	single	male	yes	no
primary	married	male	no	yes
university	divorced	female	no	yes
university	married	female	yes	yes
secondary	single	male	no	no
university	single	female	no	yes
secondary	divorced	female	no	yes
secondary	single	female	yes	yes
secondary	married	male	yes	yes
primary	married	female	no	yes
secondary	divorced	male	yes	no
university	divorced	female	yes	no
secondary	divorced	male	no	yes

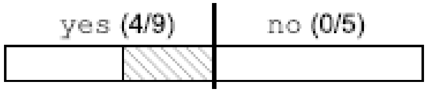
```
IF   MaritalStatus = single
 AND Sex = female
THEN Approved = yes
```



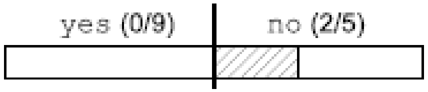

```
IF   MaritalStatus = single
 AND Sex = male
THEN Approved = no
```



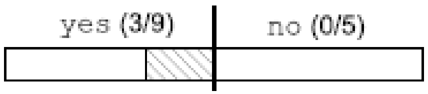

```
IF   MaritalStatus = married
THEN Approved = yes
```




```
IF   MaritalStatus = divorced
 AND HasChildren = yes
THEN Approved = no
```



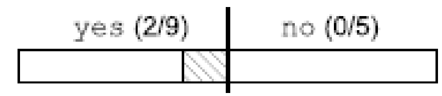

```
IF   MaritalStatus = divorced
 AND HasChildren = no
THEN Approved = yes
```



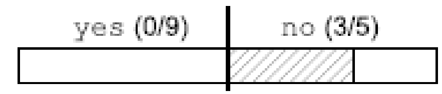
Survey data
association rule learning

Education	Marital Status	Sex	Has Children	Approved
primary	single	male	no	no
primary	single	male	yes	no
primary	married	male	no	yes
university	divorced	female	no	yes
university	married	female	yes	yes
secondary	single	male	no	no
university	single	female	no	yes
secondary	divorced	female	no	yes
secondary	single	female	yes	yes
secondary	married	male	yes	yes
primary	married	female	no	yes
secondary	divorced	male	yes	no
university	divorced	female	yes	no
secondary	divorced	male	no	yes

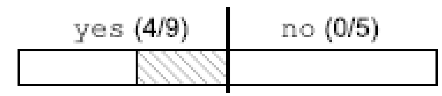
```
IF   MaritalStatus = single
 AND Sex = female
THEN Approved = yes
```



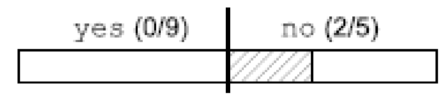

```
IF   MaritalStatus = single
 AND Sex = male
THEN Approved = no
```



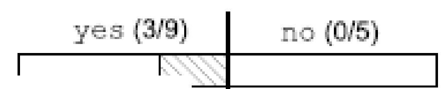

```
IF   MaritalStatus = married
THEN Approved = yes
```



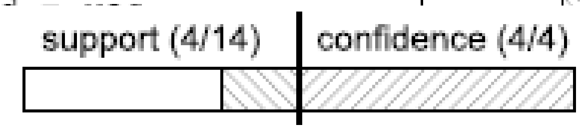

```
IF   MaritalStatus = divorced
 AND HasChildren = yes
THEN Approved = no
```



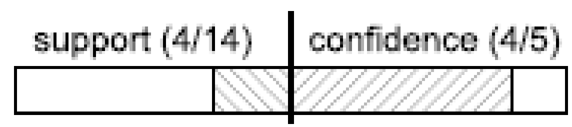

```
IF   MaritalStatus = divorced
 AND HasChildren = no
THEN Approved = yes
```



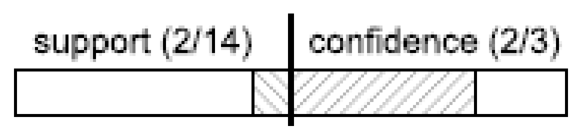

```
IF   Education = university
THEN Sex = female
```




```
IF   Approved = no
THEN Sex = male
```




```
IF   Education = secondary
 AND MaritalStatus = divorced
THEN HasChildren = no
 AND Approved = yes
```



Association Rule Learning

Given: a set of transactions D

Find: all association rules that hold on the set of transactions that have

- user defined minimum support, i.e., support > MinSup, and
- user defined minimum confidence, i.e., confidence > MinConf

It is a form of exploratory data analysis, rather than hypothesis verification

Searching for the associations

- Find all large itemsets
- Use the large itemsets to generate association rules
- If XY is a large itemset, compute

  r =support(XY) / support(X)
- If r > MinConf, then X $\Rightarrow$ Y holds

  (support > MinSup, as XY is large)

Large itemsets

- Large itemsets are itemsets that appear in at least MinSup transaction
- All subsets of a large itemset are large itemsets (e.g., if A,B appears in at least MinSup transactions, so do A and B)
- This observation is the basis for very efficient algorithms for association rules discovery (linear in the number of transactions)

Association rules vs. Classification rules

- Exploration of dependencies
- Different combinations of dependent and independent attributes
- Complete search (all rules found)

- Focused prediction
- Predict one attribute (class) from the others
- Heuristic search (subset of rules found)

Part IV. Descriptive DM techniques

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

Hierarchical clustering

- Algorithm (agglomerative hierarchical clustering):

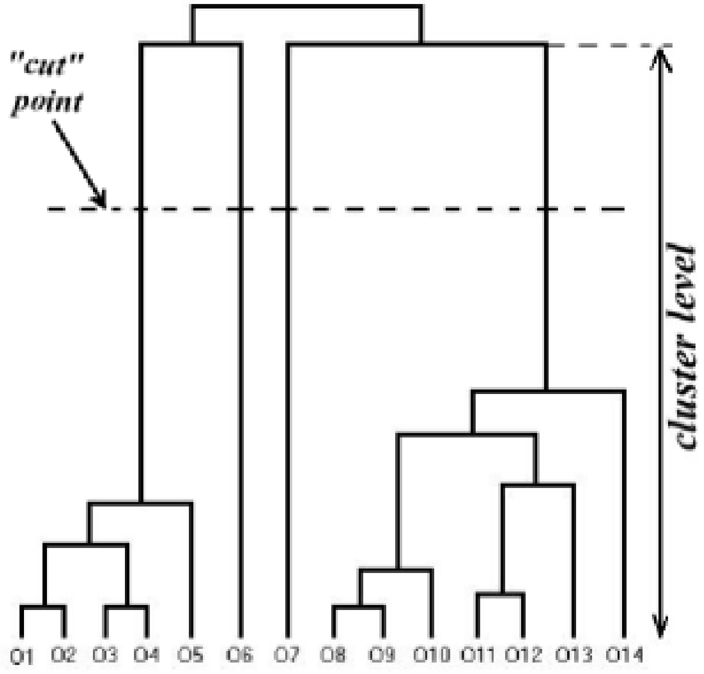
Each instance is a cluster;

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;

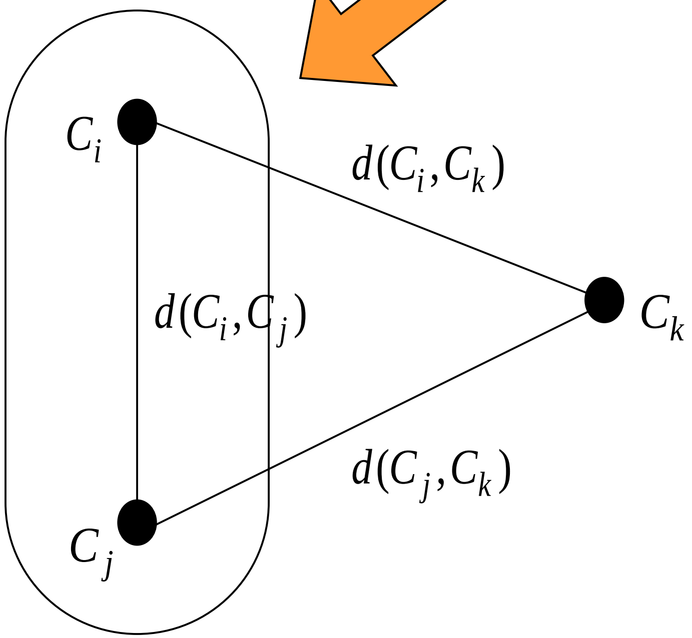
until one cluster left;

- Dendogram:



Hierarchical clustering

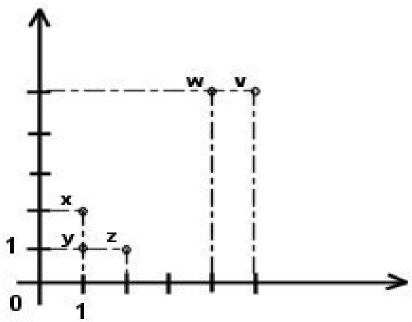
- Fusing the nearest pair of clusters



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

- Computing the dissimilarities from the “new” cluster

Hierarchical clustering: example



a) sample problem

	x	y	z	w	v
x	0	1	1	5	5.66
y		0	1.41	4.24	5
z			0	4.47	5
w				0	1
v					0

b) dissimilarity matrix

	(x,y)	z	w	v
(x,y)	0	1.41	5	5.66
z		0	4.47	5
w			0	1
v				0

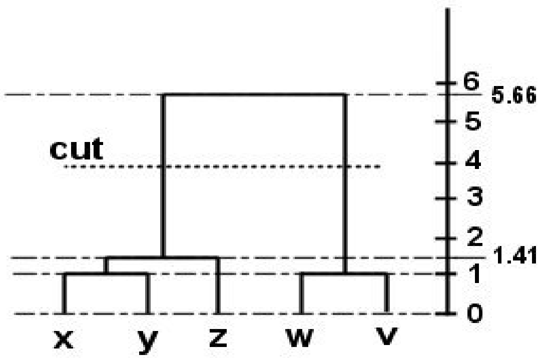
c) dissimilarity matrix after 'fusing' elements **x** and **y**

	(x,y)	z	(w,v)
(x,y)	0	1.41	5.66
z		0	5
(w,v)			0

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

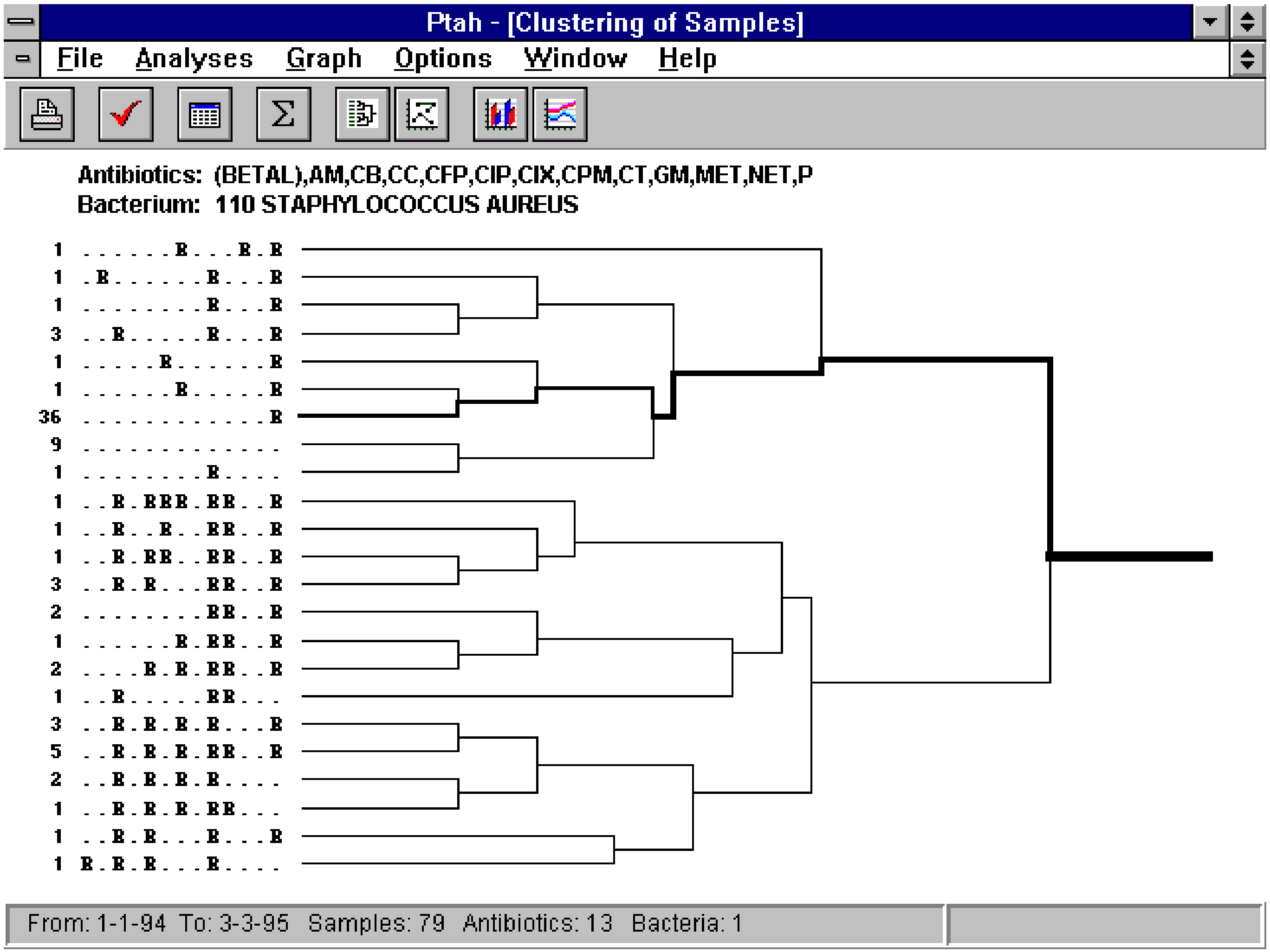
	(x,y,z)	(w,v)
(x,y,z)	0	5.66
(w,v)		0

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]

Course Outline

I. Introduction
- Data Mining and KDD process
- Introduction to Data Mining
- Data Mining platforms

II. Predictive DM
- Decision Tree learning
- Bayesian classifier
- Classification rule learning
- Classifier evaluation

III. Predictive DM
- Regression

IV. Descriptive DM
- Predictive vs. descriptive induction
- Subgroup discovery
- Association rule learning
- Hierarchical clustering