Intelligent Decision Support Systems

(Part VI - INTELLIGENT MODELS FOR DECISION SUPPORT: DISCRIMINANT MODELS)

Miquel Sànchez i Marrè
miquel@cs.upc.edu
http://cs.upc.edu/~miquel

Course 2016/2017

https://kemlg.upc.edu
PART 6 – INTELLIGENT MODELS FOR DECISION SUPPORT: DISCRIMINANT MODELS

https://kemlg.upc.edu
Discriminant Models
Discriminant Models

- Rule-Based Reasoning
  - (AI) Decision Trees
  - (AI) Classification Rules
  - (Stats) Discriminant Analysis
  - (AI & Stats) Box-Plot Based Induction Rules, Random Forests

- Case-Based Reasoning
  - (AI) Instance-Based Learning (IBL): K-NN classifier

- Bayesian Reasoning
  - (AI & Stats) Naive Bayes Classifier

- Statistical Learning Classifiers
  - (AI) Support Vector Machines

- Discriminant/ Classifier Ensemble Methods
  - (AI) Bagging
  - (AI) Boosting: Adaboost
  - (AI) Random Forest
Discriminant Models
Rule-Based Reasoning (RBR)

https://kemlg.upc.edu
Expert Systems

- Definition:

- They are Knowledge-based Systems which solve complex and specialized problems usually solved by human experts, in a concret domain.
Why appeared Expert Systems?

- Economic needs
  - High qualified experts are not always available and is expensive
  - Cheap Learning tools for other experts or non-experts
  - Preserve the experts’ knowledge

- Computational efficiency needs
  - General problema solving methods are inefficient (*soft methods*)
Expert Knowledge

DATA + ALGORITHM = PROGRAM

KNOWLEDGE + INFERENCE = ES
ES Classification according to the tasks [Hayes-Roth et al., 1983] (1)

- **Interpretation** Systems
  - Infer descriptions of situations, from data and observations

- **Predictive** Systems
  - Infer likely consequences from situations or events

- **Diagnostic** Systems
  - Infer the faults of a System from the available symptoms

- **Design** Systems
  - Develop object configurations satisfying some constraints.
ES Classification according to the tasks [Hayes-Roth et al., 1983] (2)

- **Planning Systems**
  - Generate sequences of actions to achieve some issues

- **Monitoring Systems**
  - Study of the system behaviour along time

- **Corrective / Repairing Systems**
  - Generate solutions for System faults

- **Control Systems**
  - Study and manage the behaviour of a dynamic system
Knowledge Engineering Steps (1)

[ Buchanan et al., 1983 ]

- Identification
- Conceptualization
- Formalization
- Implementation
- Refinement
- Reformulation
- Redesign
- Requirements
- Concepts
- Structure
- Rules
- Test

© Miquel Sànchez i Marrè, KEMLG, 2014
Knowledge Engineering Steps (2)

- IDENTIFICATION
  - Viability of the ES building
  - Search knowledge sources (experts, books, etc.)
  - Determine the necessary data to solve the problem
  - Determine the issues (solutions) and the criteria which determine the solution
Knowledge Engineering Steps (3)

- **CONCEPTUALIZATION**
  - Detail the basic elements to characterize the domain (*relevant facts*) and their own *relations*.
  - Distinguish the *evidences*, the *hypotheses* and the *actions* to be done.
  - Enumerate the different *hypotheses/objectives*.
  - *Decompose* the problem *into* subproblems.
  - Characterize the reasoning blocks and the reasoning flow.
Knowledge Engineering Steps (4)

- **FORMALIZATION**

  - Determine the *reasoning strategies* required:
    - Classification / diagnosis / temporal planning / causal structures / spatial design / configuration

  - Identify *search space* and *search type*

  - Identify the *problem solving methodology*:
    - Heuristic classification / Constructive resolution / Hypothesis and hierarchical test

- Analyze the *inexactness* (uncertainty, imprecision or incompleteness) and the *completeness*
Knowledge Engineering Steps (5)

- **IMPLEMENTATION**
  - Knowledge Representation and implementation
    - Fact Base Definition
    - Modular structure of the Knowledge Base
    - Definition of Inference Rules for each Module
  - Decisions over the problem solving control (meta-knowledge)
    - Definition of Meta-Rules associated to each module

- **TESTING**
  - Determine with expert/s a set of representative test cases
  - Assess the System operation (prototype)
Expert System Architecture
Knowledge Base (1)

- Domain Knowledge + Heuristic Knowledge
- Knowledge Types:
  - Factual Knowledge
    - Objects and features
  - Conditional Knowledge
    - Conditions and deductions
  - Relational Knowledge
    - Temporal, Causal and Conceptual relations
Knowledge Base (2)

- Knowledge Representation techniques

Inference Rule System / Production System:
  *First used and most commonly used*

Structured Representations

- To model objects and relations
  - Semantic Networks/Frames
  - To describe the domain

Mixed Representations: *Rules + Structured representations*
Knowledge Base (3)

- Organization of the knowledge domain and of the problem solving process

Inference Rules

IF <Conditions> THEN <Actions>
Knowledge Base (4)

- Each rule is normally formed off:
  - <Rule-Identifier>
  - <Conditions or Premisses>
    - Propositions
    - First-order Predicates
  - <Rule-Certainty>
  - <Actions or Conclusions>
    - New deductions
    - Actions
    - Computations

© Miquel Sànchez i Marrè, KEMLG, 2014
Knowledge Base (5)

- Examples

(RDECP03
  High-Sludge-Conc-PrimDec
  No Sludge-Waste-PrimDec
  0.8
  Clean-sewage tube
  . . . )

(R08007
  No Neutropènia
  Associated-dermatology ectima-gangrenosum
  very-possibly
  Pseudomones
  . . . )
Inference Engine

Inference Engine ≡ Reasoning Module

To deduce new facts, to execute actions to solve the current problem, from a set of initial facts, and a given knowledge base, with the user interaction.

Inference Engine

Rule Interpret + Control Strategy/ies
General Cycle of an Inference Engine

- **Detection**: Obtaining the set of applicable rules

  - Conflict Set Formation

- **Selection**: Selection of the rule to be applied / fired

  - Conflict Set Resolution

- **Application**: Application / Firing of the selected rule

  - Inference
General Cycle: Detection

- Construction of the candidate rules to be applied

- Rules are candidate or not depending on control strategy used

- Rule interpreter make the computations and needed instantiations which are possible in each problem solving state.

- One rule can be used with different instantiations (First-order predicates)
General Cycle: Selection

- Selection of the best rule among the obtained ones in the previous step.
- Selection depends on the conflict resolution strategy used by the inference engine.
- Most commonly used criteria (usually more than one criterium is used):
  - The more/less used rule
  - The most specific/most general rule
  - The most informative → it gives the highest number of unknown facts.
  - The rule with a higher certainty degree
  - The first rule in order

© Miquel Sànchez i Marrè, KEMLG, 2014
General Cycle: Application

- The *rule interpreter* executes the selected rule updating the State of the Fact Base with *new deductions, computations, actions and/or new subgoals*

- *Instance Propagation* (in First-order Predicate Calculus)

- If t is needed, *certainty propagation* from premises to conclusions, by means of different logic connective
End of the Cycle

- The cycle ends when **no more applicable rules are found** or when **the desired conclusion/s are reached** or when **some unexpected exceptions happen**

- Depending on **the problem** and on the **control strategy** the reasoning chain could be cut

Some back steps must be reconsidered => backtracking must be used
Inference Engines: Strategies (I)

- Deductive Engine / Progressive Chaining
  - *forward chaining*
  - *data driven*
  - *Evidences, symptoms, data → conclusions and/or hypothesis*

\[ A \land B \land C \rightarrow H \]
**Inference Engines: Strategies (II)**

- Inductive Engines / Regressive Chaining
  - *backward chaining*
  - *goal driven*
  - Conclusions and/or hypothesis \( \rightarrow \) *Evidences, symptoms, data*

\[
A \land B \land C \rightarrow H
\]
Forward Chaining

- Based in Modus Ponens:
  \[ A, A \rightarrow B \mid -B \]

- Is a deductive method according to classic logics

- **Problem Resolution**: Search from initial state to the final state (goal), through the intermediate states generated by the chain inferences derived from rule application

- Starting from evidences/symptoms/data, tries to deduce all it is possible

© Miquel Sànchez i Marrè, KEMLG, 2014
Forward Chaining: Operation

- Fact Base is initialized with the known facts
- Obtain the derivable consequences of the Fact Base:
  - Selection of the aplicable rules, which are those with known antecedents (i.e., they are in the Fact Base) or they are askable facts.
  - Add the new conclusions/values to the Fact Base
Forward Chaining. Pros & Cons

- **Drawbacks**
  - **Does not focus on a goal**: The conflict resolution strategy/ies is very critical
  - **Combinatorial Explosion**: Due to possible instantiations of the premise predicates (First-order Predicate Calculus)

- **Advantages**
  - Makes easier the Knowledge Formalization
  - Modus ponens is very intuitive
# Forward Chaining: Example

<table>
<thead>
<tr>
<th>Knowledge Base</th>
<th>Fact Base</th>
<th>Goal/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: $A \land B \land C \rightarrow D$</td>
<td>$A$</td>
<td>$G??$</td>
</tr>
<tr>
<td>R2: $A \land E \land F \rightarrow G$</td>
<td>$E$</td>
<td></td>
</tr>
<tr>
<td>R3: $B \land C \land D \rightarrow H$</td>
<td>$B$</td>
<td></td>
</tr>
<tr>
<td>R4: $E \rightarrow C$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R5: $A \land H \rightarrow F$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6: $A \land C \rightarrow H$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Miquel Sànchez i Marrè, KEMLG, 2014
Forward Chaining: Example

1: A, E, B
2: A, E, B, C(R4)

A, E, B, C, D

R1

R3

R6

3: R1

R6

A, E, B, C, H

R5

A, E, B, C, D, H

A, E, B, C, D, H

A, E, B, C, D, H, F

A, E, B, C, D, H

A, E, B, C, D, H, F, G!

A, E, B, C, D, H, F ....
Backward Chaining

- It is an inductive method. The deductive direction is broken:

  \[ B, A \rightarrow B \quad \neg \quad | \quad -A \]

- Ledded by a goal: Hypothesis to be validated. The reasoning chain must be reconstructed in reverse order.

- Each step implies new subgoals or subhypothesis required to be validated.
Backward Chaining: Operation

- Initialize the Fact Base with an initial set of facts
- Initialize the list of hypothesis /goals to be verified
- **While** there are more hypothesis to be verified do
  - Validate the first hypothesis of the list

**EndWhile**

- To validate the hypothesis means to:
  - **if** it is already validated \(\rightarrow\) remove it from the list
    - Check whether it is already verified in the Fact Base
  - **else** use the Knowledge Base and the Fact Base to validate it

- Select one rule

- Add the premisses of the rule as new subgoals to be validated instead of the hypothesis
Backward Chaining: Advantages

- The problem resolution is better directed. Only the necessary knowledge and facts are considered to solve the problem.

- The problem solving is really the exploration of an and/or graph.
# Backward Chaining: Example (I)

<table>
<thead>
<tr>
<th>Knowledge Base</th>
<th>Fact Base</th>
<th>Goal/ s</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: A ∧ B → C</td>
<td>A</td>
<td>H??</td>
</tr>
<tr>
<td>R2: C → D</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>R3: E ∧ F → G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4: A → E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R5: D → G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6: A ∧ G → H</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Miquel Sànchez i Marrè, KEMLG, 2014
Backward Chaining: Example (II)

H ➔ G

A ➔ H

R6

E ➔ F

A ➔ G

R6

D ➔ G

R5

F ➔ E

R3

G ➔ H

R3

A ➔ F

R4

G ➔ A

R5

© Miquel Sánchez i Marrè, KEMLG, 2014
Backward Chaining: Example (III)

Let us suppose that $F$ can be asked

We must go back and reconsider other options
Backward Chaining: Example (IV)

Backward Chaining: Example (IV)
Backward Chaining: Example (V)
Advantages of Rule-Based Systems

- Adequated in ill-structured domains
- Efficient in diagnosis and classification tasks
- Autoexplanation ability
- Easiness for user communication
- Extensions are easily constructed (approximate reasoning)
Limitations of Rule-Based Systems

- Fragility
- Difficulty with controlling the reasoning
- Low reusability of Knowledge Bases
- Unable to learn
- Knowledge Acquisition problem
- Validation problem
History of Expert Systems (1)

- Oldest ones ≈ 1965
- DENDRAL (1965-1970)
  - Mass spectography and magnetic resonance of organic molecules
- META-DENDRAL (1970)
  - Heuristic rules construction from data
- MACSYMA
  - Manipulation of algebraic formula
History of Expert Systems (2)

- MYCIN (1972-1976)
  - Diagnosis of infectious illness in blood
  - 400 rules
  - Reasoning with uncertainty

- EMYCIN (1980)
  - Exports MYCIN’s control system
  - First Expert System Environment (shell)

- HEARSAY-II (1975)
  - Natural Language Interpretation (hear + say)
  - 1000 words
History of Expert Systems (3)

- PROSPECTOR (1977)
  - Mining Exploration
  - Another uncertainty reasoning model

- R1/XCON (1980)
  - Computer System’s Configuration
  - DEC, ≈ 200,000 rules

- INTERNIST (1982)
  - Internal Medicine Diagnosis
  - 500,000 - 1,000,000 rules
History of Expert Systems (4)

- CENTAUR (1983)
  - Diagnosi of pulmonar diseases
  - Rules and prototypes

- MOLE (1986)
  - Expert System classification shell

- TEST (1987)
  - Troubleshooting Expert System Tool
  - Diagnosis / classification

- VT (1988)
  - Vertical Transportation
  - Elevator System design
Meta-Knowledge / Meta-Reasoning

- **Meta-reasoning** ≡ reasoning over the own reasoning
  - Controlling **how** and **when** to apply the knowledge
  - Implicit Meta-knowledge
    - Conflict resolution strategy (criteria)
    - In first-generation Expert Systems: Artificial Premisses to control rule applicability (i.e., repeat or !)
  - Explicit Meta-knowledge
    - Introduction of **meta-rules** (Davis, 1980): Rules acting over rules
    - Separation between control and knowledge
    - **Unified reasoning mechanism**: Inference engine used both by rules and by meta-rules
    - **Strategy** concept: Necessary elements ordered for the problem solving process
Meta-Rules

- **Meta-rule**: Control unit over the knowledge
- Kind of meta-rules.
  - Meta-rules over rules
    - Activate / deactivate rules
  - Meta-rules over **modules**
    - Kind of search in the modules (forward, backward)
    - Cut level in the minimum certainty of the rules
    - Rule subsumption
  - Meta-rules over **strategies**
    - Strategy: ordered set of modules to be visited
    - Exceptions
  - Meta-rules over actuation plans
    - Which strategy should be applied first when more than one are available

© Miquel Sànchez i Marrè, KEMLG, 2014
Meta-rules: examples

(MR-DECP01
  OK-BOMB
  1.0
  (DEACTIVATE-RULES RDECP005
   RDECP006 RDECP007
   RDECP008 RDECP009
   RDECP019 RDECP020))

(MR-PRI NC
  FEVER
  1.0
  (BACKWARD-ENGINE FLUE))

(MR-ESTR01
  CLASS1
  POSSIBLE
  (VISIT-MODULES C1))

(MR-03024
  SI DA
  POSSIBLE
  (VISIT-MODULES
   BACTERIANA-ATÍPICA
   PNEUMOCISTIS-CARINI TBC
   CITOMEGALOVIRUS CRIPTOCOC
   NOCARDIA ASPERGILLUS
   PNEUMOCOC ENTEROBACTÈRIES))

(MR-02012
  AGE < 14
  SURE
  (STOP-SYSTEM))
Discriminant Models
Rule-Based Reasoning (RBR)

Decision trees

https://kemlg.upc.edu
Decision Trees (1)

- **Goal:** to induce a *decision tree* from a supervised database, which represents the *discrimination/classification process* to be followed to get the *label class* for an instance, through the attribute values it shows.

- **Applicability criteria:** a *supervised database*, with one qualitative attribute being the *class attribute*, with a representative number of examples of the different possible *class labels*.

- **Most common methods:**
  - ID3 [Quinlan, 1979; 1983; 1986] Information Gain
  - CART [Breiman *et al.*, 1984] Impurity measure
  - C4.5 [Quinlan, 1993] Gain ratio → J4.8 [WEKA] → C5 [RuleQuest, Quinlan]

- **Input:** original supervised data matrix

- **Output:** a decision tree which is able to discriminate/classify the qualitative attribute of interest (class attribute)

- **Evaluation Parameters:** compactness, predictive accuracy, scalability, robustness, interpretability
Decision trees

- The nodes are qualitative attributes
- The branches are the possible values of a qualitative attribute
- The leaves of the tree have the qualitative prediction of the attribute that acts as a class label
- Model the process of deciding to which class belongs a new example of the domain
Decision Tree: example 1

Decision tree for the contact lens data
Decision Tree: example 2

Decision tree for the weather data

outlook

- sunny
- overcast
- rainy

humidity
- high
- normal

windy
- false
- true

no
yes

yes
no
ID3 Algorithm

https://kemlg.upc.edu
**ID3 algorithm (1)**

- **ID3** ≡ Induction Decision Tree [Quinlan, 1979], [Quinlan, 1986]
- Machine Learning Technique
- Decision Tree Induction
- Top-Down strategy
- From a set of *examples/instances* and the class to which they belong, it builds up the *best* decision tree which explains the instances

Criteria:
- **Compactness** → the more compact better complexity
- **Goodness** → predictive/discriminant ability
It is a greedy algorithm selecting at each step
the best attribute
→
the best one
is the most discriminant one (potentially more useful)
according to
Gain Information function: $G(X,A)$
ID3 (3)

- The construction process is **iterative**:
  - (1) It selects a subset (window) of the examples *(training set)*.
  - (2) It is constructed a decision tree that allows to discriminate the set of examples of the window.
  - (3) **If** the decision tree induced explains the rest of examples of the training set
    - **Then**
      - the decision tree is the definitive
    - **Else**
      - the badly classified examples (exceptions) are added to the window and go again to (2)
    - **End If**
ID3 (4)

\[
\begin{align*}
E_1 &= ((A_1 V_{11}) (A_2 V_{12}) \ldots (A_n V_{1n}) C_i) \\
E_2 &= ((A_1 V_{21}) (A_2 V_{22}) \ldots (A_n V_{2n}) C_j) \\
E_3 &= ((A_1 V_{31}) (A_2 V_{32}) \ldots (A_n V_{3n}) C_k) \\
& \ldots \\
E_m &= ((A_1 V_{m1}) (A_2 V_{m2}) \ldots (A_n V_{mn}) C_k)
\end{align*}
\]

Training set

Window

© Miquel Sànchez i Marrè, KEMLG, 2016
ID3: basic idea (5)

- It is a **greedy algorithm** selecting at each step the **best** attribute.

- Select at each step the attribute which **can discriminate more**.

- The selection is done through maximizing a certain function $G(X, A)$. 

© Miquel Sànchez i Marrè, KEMLG, 2016
ID3: formalization (6)

- $X \equiv \text{set of examples} \equiv \{x_j\}_{j=1,m}$
- $A \equiv \text{set of attributes} \equiv \{A_k\}_{k=1,n}$
- $C \equiv \text{set of classes} \equiv \{C_i\}_{i=1,p}$
- $\# \equiv \text{Cardinality}$

$A_i \in A$, an attribute
$x \in X$, an example
$v$, a value

$V(A_i) \equiv \text{set of possible values for attribute } A_i$
$A_i(X) \equiv \text{value of } X \text{ for } A_i$
$A^{-1}_i(v) = \{x \in X : A_i(x) = v\}$

$V(A_i) \equiv \{v_l\}_{l=1,q}$
**ID3: selection criteria (7)**

- Select attribute $A_k$ which maximizes the gain of information

$$G(X, A_k) = I(X,C) - E(X, A_k) \iff E(X, A_k) \approx 0$$

where

$$I(X,C) = -\sum_{C_i \in C} p(X,c_i) \cdot \log_2 p(X,c_i)$$

$$E(X, A_k) = \sum_{v_l \in V(A_k)} p(X,v_l) \cdot I(A^{-1}_k(v_l),C)$$

$$p(X,c_i) = \frac{\# C_i}{\# X}$$

$$p(X,v_l) = \frac{\# A^{-1}_k(v_l)}{\# X}$$

Entropy of information

Probability that one example belongs to class $C_i$

Probability that one example has the value $v_l$ for the attribute $A_k$
**ID3: algorithm (8)**

Function ID3 (in X, A are sets) returns decision tree is

```plaintext
var tree1, tree2 are decision tree endvar

option
  case All examples belong to the same class C_i do
    tree1 ← buildTree (C_i)
  case Not all examples belong to the same class C_i do
    option
      case A ≠ ∅ do
        A_max ← max \( A_k \in A \) \( \{G(X,A_k)\} \);
        tree1 ← buildTree(A_max);
        for each \( v \in V(A_{max}) \) do
          tree2 ← ID3(A_{max}^{-1} (v), A-{A_{max}});
          tree1 ← addBranch(arbre1, arbre2, v)
        endfor each
      case A = ∅ do
        tree1 ← buildTree(majorityClass(X))
      endcase
    endoption
  endcase
endoption
returns tree1
endfunction
```
**ID3: example (9)**

<table>
<thead>
<tr>
<th></th>
<th>Eye Colour</th>
<th>Hair Colour</th>
<th>Height</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Blue</td>
<td>Blonde</td>
<td>Tall</td>
<td>C+</td>
</tr>
<tr>
<td>E2</td>
<td>Blue</td>
<td>Brown</td>
<td>Medium</td>
<td>C+</td>
</tr>
<tr>
<td>E3</td>
<td>Brown</td>
<td>Brown</td>
<td>Medium</td>
<td>C-</td>
</tr>
<tr>
<td>E4</td>
<td>Green</td>
<td>Brown</td>
<td>Medium</td>
<td>C-</td>
</tr>
<tr>
<td>E5</td>
<td>Green</td>
<td>Brown</td>
<td>Tall</td>
<td>C+</td>
</tr>
<tr>
<td>E6</td>
<td>Brown</td>
<td>Brown</td>
<td>Low</td>
<td>C-</td>
</tr>
<tr>
<td>E7</td>
<td>Green</td>
<td>Blonde</td>
<td>Low</td>
<td>C-</td>
</tr>
<tr>
<td>E8</td>
<td>Blue</td>
<td>Brown</td>
<td>Medium</td>
<td>C+</td>
</tr>
</tbody>
</table>
ID3: example (10)

\[ I(X, C) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \]

\[ (1,2,5,8) \quad (3,4,5,7) \]

\[ C^+ \quad C^- \]

\[ E(X, \text{Eye-colour}) = \frac{3}{8} (-1 \log_2 1 - 0 \log_2 0) + \frac{2}{8} (-0 \log_2 0 - 1 \log_2 1) + \frac{3}{8} (-\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3}) = 0.344 \]

\[ E(X, \text{Hair-colour}) = \frac{2}{8} (-\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2}) + \frac{6}{8} (-3/6 \log_2 \frac{3}{6} - 3/6 \log_2 \frac{3}{6}) = 1 \]

\[ E(X, \text{Height}) = \frac{2}{8} (-1 \log_2 1 - 0 \log_2 0) + \frac{4}{8} (-\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2}) + \frac{2}{8} (-0 \log_2 0 - 1 \log_2 1) = 0.5 \]
ID3: example (11)

\[
G(X, \text{Eye-colour}) = 1 - 0.366 = 0.656 \\
G(X, \text{Hair-colour}) = 1 - 1 = 0 \\
G(X, \text{Height}) = 1 - 0.5 = 0.5
\]

<table>
<thead>
<tr>
<th>Eye-colour</th>
<th>Hair-colour</th>
<th>Height</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td>Brown</td>
<td>Medium</td>
<td>C-</td>
</tr>
<tr>
<td>1, 2, 8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brown</td>
<td></td>
<td>Tall</td>
<td>C+</td>
</tr>
<tr>
<td>3, 6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td></td>
<td>Low</td>
<td>C-</td>
</tr>
<tr>
<td>4, 7, 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\( I(X,C) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{1}{3} = 0.918 \)

\(1\)

\( E(X, \text{Hair-colour}) = \frac{1}{3} (- \log_2 1 - \log_2 1) + \frac{2}{3} (-\frac{1}{2} \log_2 1/2 - \frac{1}{2} \log_2 1/2) = \frac{2}{3} \)

\( E(X, \text{Height}) = \frac{1}{3} (-0 \log_2 0 - 1 \log_2 1) + \frac{1}{3} (-1 \log_2 1 - 0 \log_2 0) + \frac{1}{3} (-0 \log_2 0 - 1 \log_2 1) = 0 \)

\( G(X, \text{Hair-colour}) = 0.918 - 0.666 = 0.252 \)

\( G(X, \text{Height}) = 0.918 - 0 = 0.918 \)
ID3: example (13)

Eye-colour

Blue
1,2,8 +
3,6 -
Brown

Green
4,7, 5 - +

Height
Tall
5 +
Medium
4 -
Low
7 -
ID3: example (14)

Eye-colour = Blue → Class = C+

Eye-colour = Brown → Class = C-

Eye-colour = Green ∧ Height = Tall → Class = C+

Eye-colour = Green ∧ Height = Medium → Class = C-

Eye-colour = Green ∧ Height = Low → Class = C-
Discriminant Models
Rule-Based Reasoning (RBR)

Classification Rules

https://kemlg.upc.edu
Classification Rules

- Let express knowledge over data though expressions like:
  - \textit{if} condition/s \textit{then} conclusion
  - Example:
    - \textit{if} salary > 300000 \textit{and} number-of-children \(\epsilon[1,2]\) \textit{then} potential customer

- Conditions Format:
  - conjunctive \(c_1 \& c_2 \& ... \& c_n\)
  - \(k\)-term-DNF \(conj_1 o ... o conj_{k'}, k'<k\)
  - \(k\)-DNF \(conj_1 o ... o conj_k\)
  - \(k\)-CNF \(disj_1 \& ... \& disj_k\)

- Selector-Based
  - A selector is the pair: \text{Attribute}_{ij} = \text{Value}_{ij}
Classification Rules and Trees

- Classifications: Rules vs. Trees

Rules:

1. IF Attribute$_1$ = Value$_{11}$ AND Attribute$_2$ = Value$_{12}$ then Class = Class$_1$
2. IF Attribute$_1$ = Value$_{11}$ AND Attribute$_2$ = Value$_{22}$ then Class = Class$_2$

Decision Tree:
Rule Induction Process

- Classification Rule Induction:
## Classification Rule induction Algorithms

- RULES (Rules Extraction System) [Pham & Askoy, 1995]
- PRISM [Cendrowska, 1987]
- CN2 [Clark & Niblett, 1989]
- Rise [Domingos, 1996]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Order</th>
<th>Learning Style</th>
<th>Algorithmic Basis</th>
<th>Construction</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RULES</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PRISM</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CN2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RISE</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
RULES

**algorithm** RULES
NumberCombinations $\leftarrow 1$
Rules $\leftarrow \emptyset$

**while** (NumberCombinations $\leq$ NumberAttributes) **and**

not All instances are classified **do**
Find all *selectors* (pairs Attribute-Value) from NON classified instances
Form *conditions* as a combination of NumberCombinations selectors

**for each** condition $Condition_i$ **do**

If instances satisfying the condition belong to the same class $C_i$
then
Create the rule R: $Condition_i \rightarrow C_i$
Check irrelevant conditions
Rules $\leftarrow$ Rules + R

endif
**endForeach**

NumberCombinations $\leftarrow$ NumberCombinations + 1

**endwhile**

**for each** unclassified instance **do**
form a rule with the instance: $R_{\text{un.ex}}$
Rules $\leftarrow$ Rules + $R_{\text{un.ex}}$

**endforeach**

**return** (Rules)
## Example: Contact Lenses

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>Reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>
Example: Contact Lenses with RULES (1)

- Selector Computation:
  - $edat = jove$
  - $edat = pre-prebiòpic$
  - $edat = prebiòpic$
  - $deficiència = miope$
  - $deficiència = hipermètrope$
  - $astigmatisme = sí$
  - $astigmatisme = no$
  - $producció llacrimal = normal$
  - $producció llacrimal = reduïda$

- Rule Generation with 1 selector in the antecedent:
  
  R1: $producció llacrimal = reduïda \rightarrow Lents = cap$
### Example: Contact Lenses with RULES (2)

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>
Example: Contact Lenses with RULES (3)

- Instances not yet classified

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebíòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebíòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebíòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebíòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebíòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebíòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>prebíòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>prebíòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>
Example: Contact Lenses with RULES (4)

- **Selector Computation:**
  - edat = jove
  - edat = pre-prebiòpic
  - edat = prebiòpic
  - deficiència = miope
  - deficiència = hipermètrope
  - astigmatisme = sí
  - astigmatisme = no
  - producció llacrimial = normal

- **Rule Generation with 2 selectors in the antecedent:**
  
  \[ R2: \text{Astigmatisme} = \text{no} \ \text{AND} \ \text{Deficiència visual} = \text{hipermètrope} \rightarrow \text{Lents} = \text{toves} \]

- If there are already unclassified instances, start rule generation with 3 selectors:
  
  ...

- If there are already unclassified instances, start rule generation with 4 selectors:
  
  ...
Example: Contact Lenses with RULES (5)

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>
algorithm PRI SM
Prism ← ∅
For each class $C_i$ do
   $E$ ← set of instances of class $C_i$
   while $E ≠ ∅$ do
      Create rule $R$: $∅ \rightarrow C_i$
      while not $R$ is perfect and attributes are available do
         For each pair attribute-value (A-V) not appearing in $R$ do
            Form $R'_{a-v}$ extending the rule $R$ adding the condition $A=V$ in the antecedent
            Compute the precision $p/t$ (positive/total) de $R'_{a-v}$
         endForEach
         Select $R'_{op}$ as the $R'_{a-v}$ maximizing the precision $p/t$ in all instances
         {if there is a tie, select the one with higher $p$}
         $R$ ← $R'_{op}$
      endwhile
      $E$ ← $E -$ instances covered by the rule $R$
      Prism ← Prism + $R$
   endwhile
endForEach
return (Prism)
### Example: Contact Lenses with PRISM (1)

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>
Example: Contact Lenses with PRISM (2)

- For class $C_i$:
  - Let set $C_i = \text{dures}$

**R:** If $\text{Astigmatisme} = \text{sí}$ $\rightarrow$ Lents = dures

- $\text{Edat} = \text{jove}$ $\rightarrow$ 2/8
- $\text{Edat} = \text{pre-presbiòpic}$ $\rightarrow$ 1/8
- $\text{Edat} = \text{presbiòpic}$ $\rightarrow$ 1/8
- $\text{Deficiència visual} = \text{miope}$ $\rightarrow$ 3/12
- $\text{Deficiència visual} = \text{hipermetrope}$ $\rightarrow$ 1/12
- $\text{Astigmatisme} = \text{no}$ $\rightarrow$ 0/12
- $\text{Astigmatisme} = \text{sí}$ $\rightarrow$ 4/12
- $\text{Producció llacrima} = \text{reduïda}$ $\rightarrow$ 0/12
- $\text{Producció llacrima} = \text{normal}$ $\rightarrow$ 4/12

If $\text{Astigmatisme} = \text{sí}$ $\rightarrow$ Lents = dures

© Miquel Sànchez i Marrè, KEMLG, 2014
## Example: Contact Lenses with PRI SM (3)

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïida</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>

© Miquel Sànchez i Marrè, KEMLG, 2014
Example: Contact Lenses with PRI SM (4)

- For class $C_i$:
  - The algorithm continues because R is not perfect

R: If Astigmatisme = sí AND ? $\rightarrow$ Lents = dures

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edat = jove</td>
<td>2/4</td>
</tr>
<tr>
<td>Edat = pre-presbiòpic</td>
<td>1/4</td>
</tr>
<tr>
<td>Edat = presbiòpic</td>
<td>1/4</td>
</tr>
<tr>
<td>Deficiència visual = miope</td>
<td>3/6</td>
</tr>
<tr>
<td>Deficiència visual = hipermetrope</td>
<td>1/6</td>
</tr>
<tr>
<td>Producció llacrimal = reduïda</td>
<td>0/6</td>
</tr>
<tr>
<td>Producció llacrimal = normal</td>
<td>4/6</td>
</tr>
</tbody>
</table>

If Astigmatisme = sí AND Producció Llacral = normal $\rightarrow$ Lents = dures
Example: Contact Lenses with PRI SM (5)

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>
Example: Contact Lenses with PRI SM (6)

- For class C_i:
  - The algorithm continues because R is not perfect

  \[ R: \text{If } \text{Astigmatisme} = \text{sí} \text{ AND Producció Llacrimal} = \text{normal} \text{ AND ?} \rightarrow \text{Lents} = \text{dures} \]

\begin{align*}
Edat & = \text{jove} & 2/2 \\
Edat & = \text{pre-presbiòpic} & 1/2 \\
Edat & = \text{presbiòpic} & 1/2 \\
Deficiència visual & = \text{miope} & 3/3 \\
Deficiència visual & = \text{hipermetrope} & 1/3
\end{align*}

\[ \text{R1: If } \text{Astigmatisme} = \text{sí AND Producció Llacrimal} = \text{normal AND deficiència visual} = \text{miope} \rightarrow \text{Lents} = \text{dures} \]

\textit{R1 is already 100\% accurate !!}
### Example: Contact Lenses with PRI SM (7)

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
</tbody>
</table>
Example: Contact Lenses with PRI SM (8)

- For class $C_i = dures$:
  - 3 instances of class $C_i = dures$ are already classified by R1
  - The algorithm continues because 1 instance of $C_i$ is still unclassified
  - ...
CN2: previous concepts

- A complex is a combination of selectors.
- CN2 create rules from continually searching the best complex (the best combination of selectors).
- The procedure for finding the best complex is an heuristic search algorithm, which keeps the best K complex found (k-beam), where K is defined by the user.
- It uses 2 heuristic values in the learning process to determine which are the best:
  - Determining if a new complex should replace the 'best complex' found so far
  - Which complexes to discard if the maximum size is exceeded
  - CN2 uses the entropy measure to evaluate the complex quality
- Once the best complex is found, a rule is formed with the best complex as the antecedent, and the consequent with the mode class of all instances covering the complex \rightarrow rules are not 100% accurate!
- At the end, when no more complex are found, if there are already instances not covered by any rule, add at the end of the rule list one default rule, which classifies the instances according to the mode class of them.
**Algorithm CN2 (k)**

Let $E$ be the set of instances of the DB. Compute SELECTORS.

$\text{CN2\_LIST} \leftarrow \emptyset$

$\text{BEST\_CPX} = \text{Find\_Best\_Complex} (E, k)$

**while** $\text{BEST\_CPX} \neq \text{NUL}$ **and** $E \neq \emptyset$ **do**

- $E' \leftarrow$ instances covered by the BEST\_CPX
- $C \leftarrow$ the mode class of the set of instances $E'$
- Create the rule $R$: “if $\text{BEST\_CPX} \rightarrow \text{class} C$”
- $E \leftarrow E - E'$
- $\text{CN2\_LIST} \leftarrow \text{CN2\_LIST} + R$ \hspace{1em} \{it is added at the end of the list\}
- $\text{BEST\_CPX} = \text{Find\_Best\_Complex} (E, k)$

**endwhile**

**if** $E \neq \emptyset$ **then**

- $C \leftarrow$ the mode class of the set of instances $E$
- Create the rule $\text{DefRule}$: “if $\emptyset \rightarrow \text{class} C$”
- $\text{CN2\_LIST} \leftarrow \text{CN2\_LIST} + \text{DefRule}$ \hspace{1em} \{it is added at the end of the list\}

**endif**

**return** $(\text{CN2\_LIST})$

---

**Procedure** Find\_Best\_Complex $(E, k)$
Example: Contact Lenses with CN2 (1)

- Selectors:
  - \( Edat = jove \)
  - \( edat = pre-prebiòpic \)
  - \( edat = prebiòpic \)
  - \( deficiència = miope \)
  - \( deficiència = hipermètrope \)
  - \( astigmatisme = sí \)
  - \( astigmatisme = no \)
  - \( producció llacírmal = normal \)
  - \( producció llacírmal = reduïda \)

- Best_complex:
  - \( Edat = jove \ AND \ deficiencia = miope \ AND \ producció llacírmal = reduïda \)

- Search the mode class of the instances covered by the best complex
  - \( Lents = cap \)

\[ R: Edat = jove \ AND \ deficiencia = miope \ AND \ producció llacírmal = reduïda \to Lents = \text{CAP} \]
Example: Contact Lenses with CN2 (2)

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>
Rise

algorithm RISE
RuleSet ← E {E is the set of instances of the DB. Each instance is a rule}
Precision_final ← Precision (RuleSet)
repeat
    Precision_initial ← Precision_final
    for each rule R ∈ RuleSet
        I ← instance nearest to the rule R not covered by the rule and being of the same class
        R’ ← Most_Specific_Generalization (R,I)
        RuleSet’ ← RuleSet - R + R’ {R’ instead of R}
        if Precision (RuleSet’) ≥ Precision (RuleSet) then
            RuleSet ← RuleSet’
            if R’ ∈ RuleSet then
                RuleSet ← RuleSet - R’ {is equal to another rule}
        endif
    endif
endfor each
Precision_final ← Precision (RuleSet)
until Precision_final ≤ Precision_initial
return (RuleSet)

Computation of distances by means of distance functions: Clark, Canberra, Manhattan, Euclidean, L’Eixample, etc.

Most_Specific_Generalization: amplifying ranges for continuous attributes.
Example: Contact Lenses with Rise (1)

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>sí</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>sí</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>
Example: Contact Lenses with Rise (2)

- Set of initial rules = set of instances:

  R1: \( Edat = \text{jove} \ AND \ deficiencia = \text{miope} \ AND \ astigmatisme = \text{no} \ AND \ producció \ llacrimal = \text{reduïda} \rightarrow Lents = \text{cap} \)

  R2: \( Edat = \text{jove} \ AND \ deficiencia = \text{miope} \ AND \ astigmatisme = \text{no} \ AND \ producció \ llacrimal = \text{normal} \rightarrow Lents = \text{toves} \)

  ...

  R24: \( Edat = \text{prebiòpic} \ AND \ deficiència = \text{hipermètrope} \ AND \ Astigmatisme = \text{sí} \ AND \ producció \ llacrimal = \text{normal} \rightarrow Lents = \text{cap} \)
### Example: Contact Lenses with Rise (3)

<table>
<thead>
<tr>
<th>Age</th>
<th>Visual Deficiency</th>
<th>Astigmatism</th>
<th>Tear Production</th>
<th>Recommended Lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>jove</td>
<td>miope</td>
<td>si</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>si</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>jove</td>
<td>hipermetrope</td>
<td>si</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>si</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>miope</td>
<td>si</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>si</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>pre-prebiòpic</td>
<td>hipermetrope</td>
<td>si</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>no</td>
<td>normal</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>si</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>miope</td>
<td>si</td>
<td>normal</td>
<td>dures</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>no</td>
<td>normal</td>
<td>toves</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>si</td>
<td>reduïda</td>
<td>cap</td>
</tr>
<tr>
<td>prebiòpic</td>
<td>hipermetrope</td>
<td>si</td>
<td>normal</td>
<td>cap</td>
</tr>
</tbody>
</table>
Example: Contact Lenses with Rise (4)

- For each rule, the nearest instance is found:

  R1: Edat = jove AND deficiencia = miope AND astigmatisme = no
      AND producció llacrimal = reduïda \(\rightarrow\) Lents = cap

  I3: Edat = jove AND deficiencia = miope AND astigmatisme = si AND
      producció llacrimal = reduïda \(\rightarrow\) Lents = cap

  . . .

- Generalization:

  New Rule
  
  Edat = jove AND deficiencia = miope AND producció llacrimal = reduïda
  \(\rightarrow\) Lents = cap
Results and Evaluation

- Testing of all algorithms with:
  - 5 different databases
  - Discretization of continuous attributes in different ranges
  - Defining as a class attribute both discrete and continuous attributes

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Execution time (s)</th>
<th>Number of rules</th>
<th>Number of rules</th>
<th>Global Accuracy</th>
<th>Accuracy of Classified instances</th>
<th>Correctly Classified instances</th>
<th>Wrong Classified instances</th>
<th>Non Classified instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULES</td>
<td>0.041</td>
<td>19</td>
<td>general</td>
<td>77.78%</td>
<td>77.78%</td>
<td>35</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>PRISM</td>
<td>0.401</td>
<td>22</td>
<td>specific</td>
<td>62.22%</td>
<td>93.34%</td>
<td>28</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>CN2</td>
<td>1.663</td>
<td>19</td>
<td>variety</td>
<td>71.11%</td>
<td>71.11%</td>
<td>32</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>RISE</td>
<td>9.053</td>
<td>90</td>
<td>specific</td>
<td>73.33%</td>
<td>97.06%</td>
<td>33</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>
Conclusions

- Algorithm performance:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Nombre de regles</th>
<th>Tipus de regles</th>
<th>Precisió de les regla</th>
<th>Precisió del model</th>
<th>% NO classificats</th>
<th>Temps execució</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rules</td>
<td>Poques</td>
<td>Generals</td>
<td>Alta</td>
<td>Alt</td>
<td>Baix</td>
<td>Baix</td>
</tr>
<tr>
<td>Prism</td>
<td>Poques</td>
<td>Regular</td>
<td>Molt alta (100%)</td>
<td>Baix</td>
<td>Alt</td>
<td>Baix</td>
</tr>
<tr>
<td>CN2</td>
<td>Mitjà</td>
<td>Específiques</td>
<td>Regular</td>
<td>Mitjà</td>
<td>Baix</td>
<td>Mitjà</td>
</tr>
<tr>
<td>Rise</td>
<td>Moltes</td>
<td>Específiques</td>
<td>Alta</td>
<td>Alt</td>
<td>Mitjà</td>
<td>Alt</td>
</tr>
</tbody>
</table>

- **Rules**: it can has a high computational cost due to combinatorial explosion. (DB not necessatly larges, but with many selectors). It is useful to discretise continuous attributes in 3 or 5 ranges at most.

- **Prism**: low computational cost. Just a few rules, all 100% accurate.

- **CN2**: computational cost is proportional to the number of attributes, the number of pairs attribute-value, and the value k.

- **Rise**: it generates a lot of rules. Rules from específic to general ones. Execution cost proportional to the number of instances.
Discriminant Models
Case-Based Reasoning (CBR)

Instance-Based Learning (IBL)

https://kemlg.upc.edu
\(k\)-NEAREST NEIGHBOUR CLASSIFIER

https://kemlg.upc.edu
**K-NN algorithm (1)**

- **Goal**: this discriminant method is a *lazy machine learning technique*, which does not produce any explicit model. It is an *instance-based learning* method, and a particular *Case-Based Reasoning classifier* technique.

- **Applicability criteria**: a *supervised database*, with one qualitative attribute being the *class attribute*, with a representative number of examples of the different possible *class labels*.

- **Input**: original supervised data matrix

- **Output**: *none output model* is produced.

- **Discrimination/ classification process**: when a new instance must be discriminated/classified, it is compared against all the instances in the dataset, using a defined similarity measure \((\text{sim})\). The *class label assigned* to the new instance is *the result of a (weighted) voting among the class labels of the k nearest neighbours*

- **Parameters**: number \(k\) of neighbours, the similarity measure \(\text{sim}()\)

- **Evaluation Parameters**: predictive accuracy, computational time
$k$-NN Algorithm (2)

- All instances correspond to points in the $n$-dimensional space.
- The nearest neighbour are defined in terms of any distance or similarity function.
- The $k$-NN returns the most common value among the $k$ training examples nearest to $d_i$.
\( k\)-NN Algorithm (3)

- Robust to noisy data by averaging \( k\)-nearest neighbours
- Similarity computation between neighbours could be biased by irrelevant attributes.
  - Removal of the least relevant attributes
  - Computing the degree of relevance of all the attributes through \textit{feature weighting} techniques
**k-NN algorithm (4)**

Input: $T = \{t_1, \ldots, t_n\}$ // Training Data points available

$D = \{d_1, \ldots, d_m\}$ // Data points to be classified

$k$ // Number of neighbours

Output: neighbours // the $k$ nearest neighbours

---

**Function K-NN**

Foreach data point $d$

neighbours = $\emptyset$

Foreach training data point $t$

$\text{dist} = \text{distance}(d, t)$

If $|\text{neighbours}| < k$ then

insert($t$, neighbours)

else

fartn = $\text{argmax}_i \text{distance}(t, \text{neighbours}_i)$

if $\text{distance}(\text{dist} < \text{fartn})$

Insert ($t$, neighbours)

Remove ($\text{fartn}$, neighbours)

endif

endif

End function
CBR vs RBR? (1)

- Why CBR?

  - **Problem**: Most of the time the trouble in building Expert Systems comes from trying to fit *experience* into *rules*.

  - Usually it is hard for an expert in a domain the *abstraction* needed to create generic rules from specific past episodes.
CBR vs RBR (2)

- The CBR Solution:
  - to directly use experience (past episodes) in the reasoning (*reason by analogy*).
  - No translation is needed.
What is CBR? (3)

- A definition:
  ".... transferring knowledge from past problem solving episodes to new problems that share significant aspects with corresponding past experience and using the transferred knowledge to construct solutions to new problems."

(Carbonell, 1986)
What is CBR? (2)

- **CBR**: A methodology of solving new problems by adapting the solutions of previous similar problems

- It uses *cases* as an episodic memory (*Case Library*).
CBR Antecedents

● Foundations of CBR
  ■ Our general knowledge about situations is recorded as scripts [Schank & Abelson, 1977]
  ■ Cognitive model is the *Theory of Dynamic Memory* [Schank, 1982]:
    ◆ Indexing is the key to use experience in understanding
    ◆ Remembering, understanding, experiencing, and learning cannot be separated from each other
    ◆ Human memory is dynamic, and change as a result of its experiences

● CBR derives from a view of understanding problem-solving as an explanation process [Riesbeck & Schank, 1989]
Case-Based Reasoning (VI)

- CBR Cycle:

1. **Retrieve**
   - new case
   - retrieved cases

2. **Adapt**
   - best case
   - adapted solution

3. **Eval**
   - evaluated solution (fail/success)

4. **Learn**
   - case to store

5. **CBR Cycle**:
   - © Miquel Sànchez i Marrè, KEMLG, 2014
Examples
Extracted from [Kolodner, 1993]

- CHEF [Hammond, 1986, 1989], a case-based planner for recipe creation
- CASEY [Koton 1988, 1989], a case-based diagnosis program to diagnose a causal explanation of the patient disorders
- JULIA [Hinrichs 1988-1992], a case-based designer in the domain of meal planning
- HYPO [Ashley, 1990], a case-based interpretive program that works in the domain of law
- PROTOS [Bareiss, 1989], a case-based classification program for audiological disorders
- CLAVIER [Hennessy & Hinkle, 1992], a case-based program for configuration of the layout of composite airplane parts for curing in autoclave
- BATTLE [Goodman, 1989], a case-based program for battle planning
- ARCHIE [Pearce et al., 1992], a case-based design program for architecture
- MEDIATOR [Simpson, 1985], a case-based arbitration program for dispute resolution
Components of a CBR System

- Cases
  - Flat or structured

- Case Library/Case Base
  - Flat memory or hierarchical/network memory

- Retrieval Methods
  - Search or indexing in the Case Library
  - Similarity Assessment

- Adaptation Methods

- Evaluation Methods

- Kind of Learning
Case Representation

- Attribute-value representation: a case is a set of features
  - case identifier
  - derivation of the case
  - description of the problem
  - diagnostic of the problem
  - solution to the problem
  - evaluation of the solution (success/failure)
  - utility measure
  - other relevant information

- Structured representation: a case is a structure relating features and other elements
  - tree or network

© Miquel Sànchez i Marrè, KEMLG, 2014
Case Retrieval (1)

- Case Retrieval is more difficult than retrieval/query in Databases.
  - DB Recuperation = exact “matching”
  - CBR Retrieval = partial “matching” (similarity)

- Similarity assessment:
  - Computed between case descriptions,
  - Usually, it is an heuristic function or distance,
  - It can be dependent on the domain.

- An example:
  - Case structure = Feature-Value vector
  - Similarity measure = \( \text{dist}(C_i, C_j) = \sum_{k=1}^{n} w_k \times \text{atr}_k \text{dist}(C_{ik}, C_{jk}) \)

- Retrieval tries to maximize the similarity between the current case and the case(s) and the retrieved cases.
Case Retrieval (2)

- The efficiency of the retrieval process hardly depends on the **Organization of the Case Library**
- Two main approaches:

  - **Flat memories**
    - Easy to manage
    - Slow for retrieval
    - Always finds the best

  - **Hierarchical memories**
    - Hard to manage
    - Fast for retrieval
    - Heuristic search

- The Case Library structure and the Case representation makes easier the relevant case retrieval and its comparison against the current problem.
Adaptation

- When the retrieved case does not perfectly match the new case, then the old solution must be adapted to obtain the new one.

- Strategies:
  - Nul adaptation
  - Structural Adaptation
    - Substitution methods
    - Transformation methods
    - Adaptation ad-hoc (special-purpose)
  - Derivational Adaptation

- Adaptation is a highly domain-dependent process.
Evaluation

- Qualify the quality of a solution
- Three basic ways:
  - Testing the proposed solution in the real world
  - Asking to a human expert
  - Executing a simulation model (laboratory, computerized simulation, etc.)
Learning

- Learning by observation (set of initial cases)
- Learning by experience
  - Learning from successful experiences
  - Learning from failed experiences
CBR Applicability

- When a large historical data repository is available
- When experts describe their domain through examples
- When experience is so valuable as the knowledge from textbooks
- When problems are not completely understood (weak domain models, poor domain knowledge)
- There are too many exceptions to general knowledge
- *When cases with similar solutions have similar problem descriptions*
CBR Applications

- Failure machine diagnosis
- Computer Network diagnosis
- Medical diagnosis
- Bank Credit analysis
- Geological source prediction
- Battle planners
- Message Classification
- Speech recognition
CBR Advantages

- Fast solution proposal, as it does not start from scratch, using previous experiences
- Easiness to extract expert or domain knowledge to create the case library
- Past failed experiences can be used to prevent making the same mistakes in the future
- Integration of learning skills is simple
  - CBR system improves its performance along time
- Exceptional cases could be easily managed
CBR Shortcomings

- The whole Case Base is not always explored, and thus non-optimal solutions could be found
- A large size of memory could be required
- Global consistency of all the cases could be difficult to maintain
- Adaptation functions must be defined for each domain.
- A CBR system cannot reason about what has never happened
Comparison against other methods

Rule-Based Reasoning
- Rules express generic knowledge (*patterns*)
- Rules used in the inferential process *exactly* match with the input problem
- It is difficult to learn new rules and maintain the consistency
  - Static knowledge
  - No learning skills
- It is difficult to acquire the expert knowledge to build the Rule Base
- Performance is constant

Case-Based Reasoning
- Cases express specific or episodic knowledge (*constants*)
- Cases used in the inferential process *partially* match with the input problem
- It is easy to learn new cases, storing them in the Case Base
  - Dynamic knowledge
  - Learning skills
- It is relatively easy to acquire the expert knowledge to build the Case Base
- Performance improves along time
Combination of CBR and RBR?

- Run both Systems in parallel
  - Rules explain what to do usually
  - The cases remember past experiences (positives or negatives)
  - Problem: What do we do when they are contradictory?

- Extraction of general knowledge from specific knowledge
  - When you have a set of cases with similar solutions, specific knowledge can be generalised in form of a rule of inference, and add it to the Knowledge Base
    - Safe Rule Learning

© Miquel Sánchez i Marrè, KEMLG, 2014
DISCRIMINANT/CLASSIFIER ENSEMBLE METHODS

https://kemlg.upc.edu
Classifier Ensemble methods

- **Goal**: to induce a *set of discriminant models* (*classifiers*), which can be of the same type or not, from different samples or weighting the samples or randomly selecting the attributes at each split node of a tree, of a supervised database, aiming at *reducing the discrimination error* of each one of the (weak) classifiers.

- **Applicability criteria**: a *supervised database*, with one qualitative attribute being the *class attribute*, with a representative number of examples of the different possible *class labels*.

- **Most common methods**:
  - Bagging [Breiman *et al.*, 1984]
  - Boosting [Shapire, 1990], AdaBoost [Freund & Shapire, 1996]
  - Random Forests [Breiman, 2001]

- **Input**: original supervised data matrix

- **Output**: a *set of classifiers* which are able to discriminate/classify the qualitative attribute of interest (*class attribute*)

- **Evaluation Parameters**: predictive accuracy, scalability, robustness

- **Discrimination/classification process**: when a new instance must be discriminated, *all the classifiers* are used to get their class label prediction. The *class label assigned* to the new instance is the *result of a (weighted) voting among the class labels of the set of classifiers*.
Learning Ensembles

- Learn multiple alternative definitions of a concept using different training data or different learning algorithms.
- Combine decisions of multiple definitions, e.g. using weighted voting.
Value of Ensembles

- When combining multiple *independent* and *diverse* decisions each of which is at least:
  - More accurate than random guessing
  - Random errors cancel each other out
  - Correct decisions are reinforced
Different types of ensemble learning

- Different learning **algorithms**
- Algorithms with different choice for **parameters**
- Data set with different **features** (e.g. random subspace)
- Data set = different **subsets** (e.g. bagging, boosting)
Different types of ensemble learning (1)

- Different algorithms, same set of training data

Training Set L → A1 → C1
Training Set L → A2 → C2
Training Set L → An → Cn

A: Inductive Algorithm
C: Classifier (induced model)
Different types of ensemble learning (2)

- Same algorithm, different parameter settings

A: Inductive Algorithm
C: Classifier (induced model)
P: Parameters for the learning algorithm
Different types of ensemble learning (3)

- Same algorithm, different versions of data set, e.g.
  - Bagging: resample training data
  - Boosting: Reweight training data
  - Decorate: Add additional artificial training data
  - RandomSubSpace (random forests): random subsets of features

A: Inductive Algorithm
C: Classifier (induced model)
Li: Sampled/weighted training data
Bagging

- Create ensembles by repeatedly randomly resampling the training data [Breiman, 1996].

- Given a training set of size $n$, create $m$ samples of size $n$ by drawing $n$ examples from the original data, *with replacement*.
  - Each *bootstrap sample* will on average contain 63.2% of the unique training examples, the rest are replicates.

- Combine the $m$ resulting models using simple majority vote.

- Decreases error by decreasing the variance in the results due to *unstable learners*, algorithms (like decision trees) whose output can change dramatically when the training data is slightly changed.
Boosting

- Originally developed by computational learning theorists to guarantee performance improvements on fitting training data for a weak learner that only needs to generate a hypothesis with a training accuracy greater than 0.5 [Schapire, 1990].

- Revised to be a practical algorithm, AdaBoost, for building ensembles that empirically improves generalization performance [Freund & Shapire, 1996].

- Examples are given weights. At each iteration, a new hypothesis is learned and the examples are reweighted to focus the system on examples that the most recently learned classifier got wrong.
Boosting: basic algorithm

- General boosting algorithm:

  Set all examples to have equal uniform weights
  
  for $t$ from 1 to $T$ do
    Learn a hypothesis, $h_t$ from the weighted examples
    Decrease the weights of examples $h_t$ classifies correctly
  endfor

- Base (weak) learner must focus on correctly classifying the most highly weighted examples while strongly avoiding over-fitting.

- During testing, each of the $T$ hypotheses get a weighted vote proportional to their accuracy on the training data.
AdaBoost pseudocode

TrainAdaBoost(D, BaseLearn)

for each example $d_i$ in $D$
    let its weight $w_i = 1/|D|$
endfor

Let $H$ be an empty set of hypotheses

for $t$ from 1 to $T$
    Learn a hypothesis, $h_t$, from the weighted examples: $h_t = \text{BaseLearn}(D)$
    Add $h_t$ to $H$
    Calculate the error, $\varepsilon_t$, of the hypothesis $h_t$ as the total sum weight of the examples that it classifies incorrectly.
    If $\varepsilon_t > 0.5$ then exit loop, else continue.
    Let $\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$
    Multiply the weights of the examples that $h_t$ classifies correctly by $\beta_t$
    Rescale the weights of all of the examples so the total sum weight remains 1.
endfor

return $H$

TestAdaBoost(ex, $H$)

Let each hypothesis, $h_t$, in $H$ vote for ex’s classification with weight $\log(1/ \beta_t)$

return the class with the highest weighted vote total.
Learning with Weighted Examples

- Generic approach is to replicate examples in the training set proportional to their weights (e.g. 10 replicates of an example with a weight of 0.01 and 100 for one with weight 0.1).

- Most algorithms can be enhanced to efficiently incorporate weights directly in the learning algorithm so that the effect is the same (e.g. implement the WeightedInstancesHandler interface in WEKA).

- For decision trees, for calculating information gain, when counting example $i$, simply increment the corresponding count by $w_i$ rather than by 1.
Experimental Results on Ensembles [Freund & Schapire, 1996; Quinlan, 1996]

- Ensembles have been used to improve generalization accuracy on a wide variety of problems.
- On average, Boosting provides a larger increase in accuracy than Bagging.
- Boosting on rare occasions can degrade accuracy.
- Bagging more consistently provides a modest improvement.
- Boosting is particularly subject to over-fitting when there is significant noise in the training data.
- Bagging is easily parallelized.
- Boosting is not easily parallelized.
Random Forests


- **Motivation**: reduce error correlation between classifiers

- **Main idea**: build a larger number of un-pruned decision trees

- **Key**: using a random selection of features to split on at each node
How Random Forests Work

- Each tree is grown on a bootstrap sample of the training set of $N$ cases.
- A number $m$ is specified much smaller than the total number of variables $M$ (e.g. $m = \sqrt{M}$ or $m = \text{int}(\log_2 M + 1)$).
- At each node, $m$ variables are selected at random out of the $M$.
- The split used is the best split on these $m$ variables.
- Final classification is done by majority vote across trees.
Advantages of Random Forests

- Error rates compare favorably to Adaboost
- More robust with respect to noise.
- More efficient on large data
- Provides an estimation of the importance of features in determining classification
Miquel Sànchez i Marrè
(miquel@lsi.upc.edu)

http://kemlg.upc.edu/