BABEŞ-BOLYAI UNIVERSITY
Faculty of Computer Science and Mathematics

ARTIFICIAL INTELLIGENCE

Intelligent systems

Machine learning

Decision trees

Intelligent systems – Machine Learning (ML)

- Typology
  - Based on algorithm
    - **Decision trees**
    - Artificial Neural Networks
    - Evolutionary algorithms
    - Support Vector Machines
    - Hidden Markov Models

Intelligent systems – decision trees (DT)

- Decision trees (DTs)
  - Aim
  - Definition
  - Solved problems
  - Example
  - Process
  - Tools
  - Advantages and limits

Intelligent systems – decision trees (DT)

- Aim
  - Divide a collection of articles in smaller sets by successively applying some decision rules → asking more questions
    - Each question is addressed based on the answer of the previous question
  - Elements are characterized by non-metric information
- Definition
  - Decision tree
    - A special graph → bicolour and oriented tree
    - Contains three node types:
      - Decision nodes → possibilities of decider (a test on an attribute of item that must be classified)
      - Hazard nodes → random events outside the control of decider (exam results, therapy consequences)
      - Result nodes → final states that have a utility or a label
    - Decision and hazard nodes alternate on the tree levels
    - Result nodes → leaf (terminal nodes)
    - (oriented) Edges of the tree consequences of decisions (can be probabilisitic)
  - Each internal node corresponds to an attribute
  - Each branch under a node (attribute) corresponds to the value of that attribute
  - Each leaf corresponds to a class

Intelligent systems – decision trees (DT)

- Problems solved by DTs
  - Problem’s instances are represented by a fixed number of attributes, each attribute having a finite number of values
  - Objective function takes discrete values
  - DT represents a disjunction of more conjunctions, each conjunction being “atribute a_i has value v_j”
  - Training data could contain errors
  - Training data could be incomplete
    - Some data have not all attributes

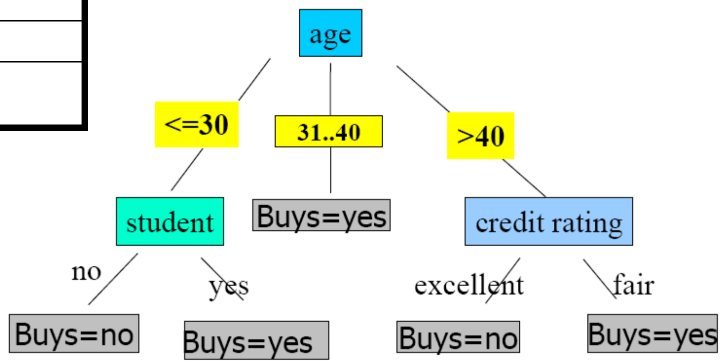
  - Classification problem
    - Binary classification
      - Instances are [(attribute$_{ij}$, value*ij*), class$_i$, i=1,2,...,n, j=1,2,...,m, classi taking 2 values]
    - Multi-class (k-class)
      - Instances are [(attribute$_{ij}$, value$_{ij}$), class$_i$, i=1,2,...,n, j=1,2,...,m, classi taking *k* values]

  - Regression problems
    - DTs are constructed in a similar manner to those of classification problems, but instead to label each node by the label of a class, each node has associated a real value or a function that depends on the inputs of that node
    - Input space is split in decision regions by parallel cuttings to Ox and Oy
    - Discrete outputs are transformed in continuous functions
    - Quality of problem solving
      - Prediction error (square or absolute)
      - Eroare (pătratică sau absolută) de predicţie

Intelligent systems – decision trees (DT)

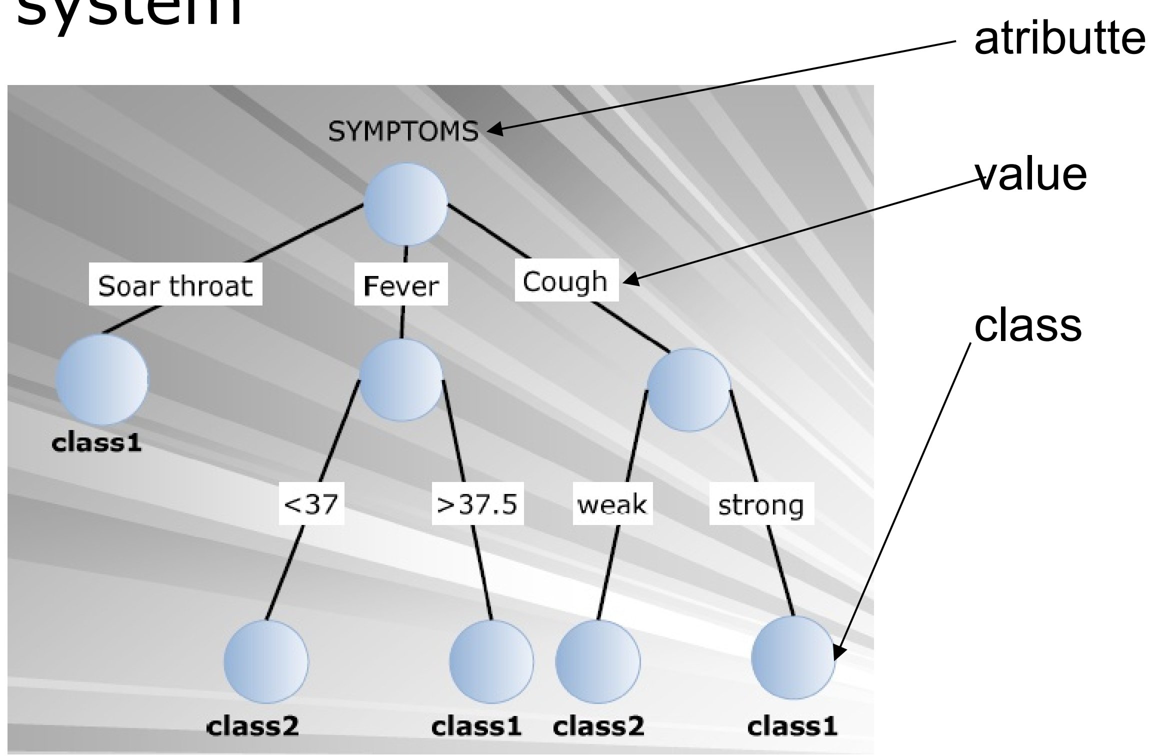
- Example

rec	Age	Income	Student	Credit_rating	Buys_computer(CLASS)
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31…40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31…40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>40	Medium	Yes	Fair	Yes
r11	<=30	Medium	Yes	Excellent	Yes
r12	31…40	Medium	No	Excellent	Yes
r13	31…40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No



Intelligent systems – decision trees (DT)

- Example
  - Medical system



Intelligent systems – decision trees (DT)

- Example
  - Credits

Approved or not

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Intelligent systems – decision trees (DT)

- Process
  - Tree construction (induction)
    - Based on training data
    - Works bottom-up or top-down (splitting)
  - Using the tree as a problem solver
    - All decisions performed along a path from the root to a leaf form a rule
    - Rules from DT are used for labeling new data
  - Pruning
    - Identify and move/eliminate branches that reflect noise or exceptions

Intelligent systems – decision trees (DT)

- Process → Tree construction (induction)
  - Split the training data into subsets based on the characteristics of data
    - A node → Question related to a property
    - Branches of a node → possible answers to the question of the node
    - Initially, all examples are located in the root
      - An attribute gives the root ant its values give the branches
    - On next levels, examples are partitioned based on their attributes → order of attributes
      - For each node, an attribute is (recursively) chosen – its values → branches
    - Splitting → greedy decision making
  - Iterative process
    - Stop conditions
      - All examples from a node belong to the same class → node is a leaf and is labeled by $class_i$
      - There are no examples → node becomes a leaf and is labeled by the majority class of training data
      - There are no attributes

Intelligent systems – decision trees (DT)

- Process → Tree construction (induction)
  - Example

rec	Age	Income	Student	Credit_rating	Buys_computer(CLASS)
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31…40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31…40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>40	Medium	Yes	Fair	Yes
r11	<=30	Medium	Yes	Excellent	Yes
r12	31…40	Medium	No	Excellent	Yes
r13	31…40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No

Intelligent systems – decision trees (DT)

- Process → Tree construction (induction)
  - Example
    - Attribute *age* is selected for the root

age

<=30

income	student	credit	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

>40

income	student	credit	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

31…40

income	student	credit	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

age

<=30

income	student	credit	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

>40

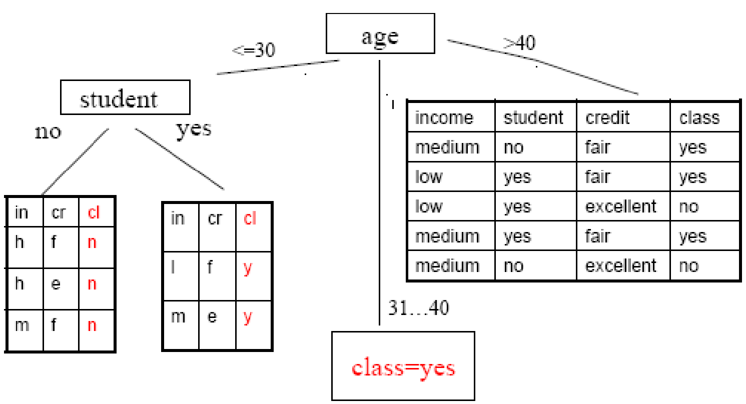
income	student	credit	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

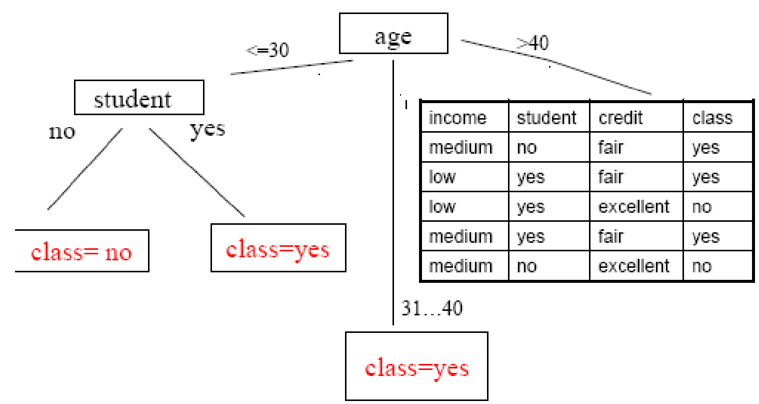
31…40

class=yes

Intelligent systems – decision trees (DT)

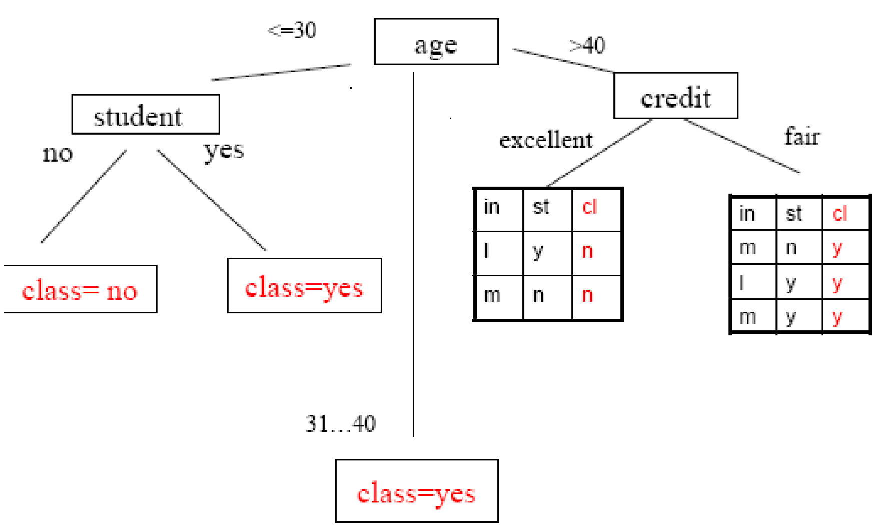
- Process → Tree construction (induction)
  - Example
    - Attribute *age* is selected for the root
    - Attribute *student* is selected on branch *age <=30*

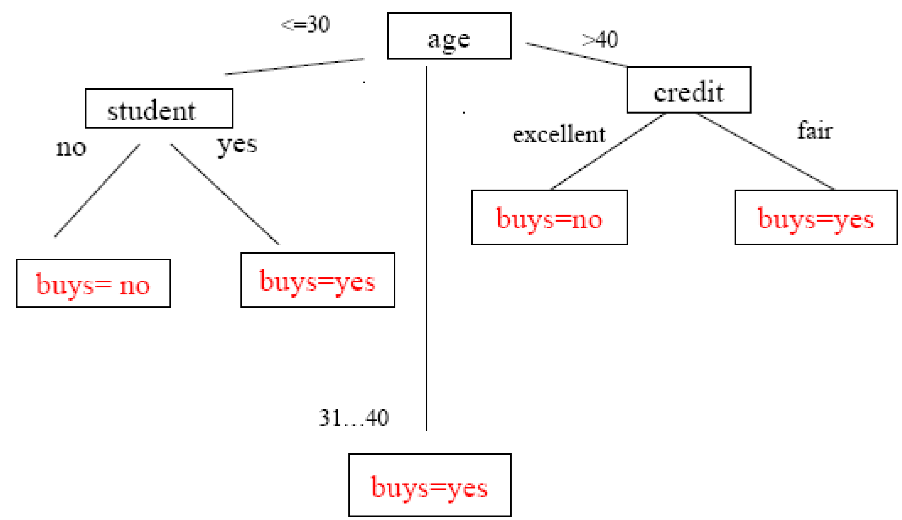




Intelligent systems – decision trees (DT)

- Process → Tree construction (induction)
  - Example
    - Attribute *age* is selected for the root
    - Attribute *student* is selected on branch *age <=30*
    - Attribute *credit* is selected on branch *age > 40*





Intelligent systems – decision trees (DT)

- Process → tree construction → ID3/C4.5 algorithm
  - Greedy, recursive, top-down, divide-and-conquer

```
generate(D, A){          //D – a partitioning of training data, A – list of attributes
    create a new node N
    if examples from D belong to a single class C then
        node N becomes a leaf and is labeled by C
        return node N
    else
        if A=∅ then
            node N becomes a leaf and is labeled by majority class of D
            return node N
        else
            separation_attribute = AttributeSelection(D, A)
            label node N by separation_attribute
            for all possible values vj of separation_attribute
                let Dj – set of examples from D that have separation_attribute=vj
                if Dj =∅ then
                        add a leaf (to node N) labeled by majority class of D
                else
                    add a node (to node N) return by generate(Dj, A–separation_attribute)
          return node N
}
```

Intelligent systems – decision trees (DT)

- Process → tree construction → ID3/C4.5 algorithm
  - AttributeSelection(D,A) → select the attribute that corresponds to a node (root or internal node)
    - Random
    - Attribute with the fewest/most values
    - Based on a pre-established order
      - Information gain
      - Gain rate
      - Gini index
      - Distance between partitions created by the attribute

Intelligent systems – decision trees (DT)

- Process → tree construction → ID3/C4.5 algorithm → Attribute Selection
  - Information gain
    - An impurity measure
      - 0 (minim) – if all examples belong to the same class
      - 1 (maxim) – if examples are uniform distributed over classes
    - Based on data entropy
      - Expected number of bits required by coding the class of an element from data
      - Binary classification (2 classes): $E(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$, where
        - p_+ - proportion of positive examples in dataset S
        - p_- - proportion of negative examples in dataset S
      - Multi-class classification: $E(S) = \sum_{i=1, 2, \ldots, k} - p_i \log_2 p_i$ – data entropy related to target attribute (output attribute), where
        - p_i – proportion of examples from class *i* in dataset S
    - Information gain of an attribute
      - How the elimination of attribute *a* reduces the dataset's entropy
      - $Gain(S, a) = E(S) - \sum_{v \in valori(a)} |S_v| / |S| \, E(S_v)$
      - $\sum_{v \in values(a)} |S_v| / |S| \, E(S_v)$ – expected information

Intelligent systems – decision trees (DT)

Process → tree construction → ID3/C4.5 algorithm → Attribute Selection

- Information gain
  - Example

	a1	a2	a3	Clasa
d1	mare	roşu	cerc	clasa 1
d2	mic	roşu	pătrat	clasa 2
d3	mic	roşu	cerc	clasa 1
d4	mare	albastru	cerc	clasa 2

S = {d1, d2, d3, d4} ➔ $p_+ = 2 / 4$, $p_- = 2 / 4$ ➔ $E(S) = - p_+\log_2 p_+ - p_-\log_2 p_- = 1$

$S_{v=mare}$ = {d1, d4} ➔ $p_+ = ½$, $p_- = ½$ ➔ $E(S_{v=mare}) = 1$

$S_{v=mic}$ = {d2, d3} ➔ $p_+ = ½$, $p_- = ½$ ➔ $E(S_{v=mic}) = 1$

$S_{v=rosu}$ = {d1, d2, d3} ➔ p+ = 2/3, p- = 1/3 ➔ $E(S_{v=rosu}) = 0.923$

$S_{v=albastru}$ = {d4} ➔ p+ = 0, p- = 1 ➔ $E(S_{v=albastru}) = 0$

$S_{v=cerc}$ = {d1, d3, d4} ➔ p+ = 2/3, p- = 1/3 ➔ $E(S_{v=cerc}) = 0.923$

$S_{v=patrat}$ = {d2} ➔ p+ = 0, p- = 1 ➔ $E(S_{v=patrat}) = 0$

$Gain(S, a) = E(S) - \sum_{v \in values(a)} |S_v| / |S| \, E(S_v)$

$Gain(S, a_1) = 1 - (|S_{v=mare}| / |S| \, E(S_{v=mare}) + |S_{v=mic}| / |S| \, E(S_{v=mic})) = 1 - (2/4 * 1 + 2/4 * 1) = 0$

$Gain(S, a_2) = 1 - (|S_{v=rosu}| / |S| \, E(S_{v=rosu}) + |S_{v=albastru}| / |S| \, E(S_{v=albastru})) = 1 - (3/4 * 0.923 + 1/4 * 0) = 0.307$

Intelligent systems – decision trees (DT)

- Process → tree construction → ID3/C4.5 algorithm → Attribute Selection
  - Gain rate
    - Penalises an attribute by integrating a new term – *split information* – that depends on spreading degree and on uniformity degree of separation
      - *Split information* – entropy related to possible values of attribute a
      - Sv – proportion of examples from dataset S that have attribute a with value v
      - *splitInformation(S,a)=* $-\sum_{v=value(a)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$

Intelligent systems – decision trees (DT)

- Process
  - Tree construction
  - Using the tree as a problem solver
    - Main idea
      - Extract the rules from the constructed tree
        - IF *age* = *"<=30"* AND *student* = *"no"* THEN *buys_computer* = *"no"*
        - IF *age* = *"<=30"* AND *student* = *"yes"* THEN *buys_computer* = *"yes"*
        - IF *age* = *"31…40"* THEN *buys_computer* = *"yes"*
        - IF *age* = *">40"* AND *credit_rating* = *"excellent"* THEN *buys_computer* = *"no"*
        - IF *age* = *">40"* AND *credit_rating* = *"fair"* THEN *buys_computer* = *"yes"*
      - Use the rules for classifying the test data (new data)
        - Let x a data without class → rules can be written as predicates
        - IF *age(x, <=30)* AND *student(x, no)* THEN *buys_computer (x, no)*
        - IF *age(x, <=30)* AND *student (x, yes)* THEN *buys_computer (x, yes)*

Intelligent systems – decision trees (DT)

- Process
  - Tree construction
  - Using the tree as a problem solver
    - Difficulties
      - *Underfitting* → DT constructed on training data is to simple → large classification error during training and testing
      - *Overfitting* → DT constructed on training data match the training data, but it ca not generalise new data
    - Solutions
      - Pruning → remove some branches (un-useful, redundant) → small tree
      - Cross-validation

Intelligent systems – decision trees (DT)

- Process
  - Tree construction
  - Using the tree as a problem solver
  - Pruning
    - Why?
      - After the DT is constructed, classification rules are extracted in order to represent the knowledge as if-then rules (easy to understand)
      - A rule is create by traversing the DT from root to a leaf
      - Each pair (attribute, value) – (node, edge) – is a conjunction in the premise of the rule (if part), except the last node of the path that is a leaf and represents the consequence (output, then part) of the rule
    - Typology
      - *pre-pruning*
        - Increasing the tree is stopped during construction by stopping the division of nodes that become leaf labeled by majority class of examples from that node
      - *post-pruning*
        - After the DT is constructed, eliminate the branches of some nodes that become leaf → classification error reduces (on testing data)

Intelligent systems – decision trees (DT)

- Tools
  - http://webdocs.cs.ualberta.ca/~aixplore/learning/D
  - WEKA → J48
  - http://id3alg.altervista.org/
  - http://www.rulequest.com/Personal/c4.5r8.tar.gz

- Biblio
  - http://www.public.asu.edu/~kirkwood/DAStuff/decisiontrees/index.html

Intelligent systems – decision trees (DT)

- Advantages
  - Easy to understand and interpret
  - Can use nominal or categorial data
  - Decision logic can be easy followed (rules are visible)
  - Works better with large data

- Disadvantages
  - Instability → change the training data
  - Complexity → representation
  - Difficult to use
  - The DT construction is expensive
  - The DT construction requires a lot of information

Intelligent systems – decision trees (DT)

- Difficulties
  - There can be more trees
    - To small
    - With a better accuracy (easy to be read and with good performances)
    - Identify the best tree → NP-problem
  - Select the best tree
    - Heuristic algorithms
    - ID3 → the smallest tree
      - Occam teorem: "always choose the simplest explanation"
  - Continuous attributes
    - Range spliting
      - How many intervals?
      - How large intervals?
  - To large trees
    - Pre-pruning → stops to construct the tree earlier
    - Post-pruning → remove some branches

Review

- Automatic learning systems
  - Machine Learning – ML
    - Supervised learning → annotated train data (by label from a predefined set) and test data have to be annotated by using the learnt model (by one of the known labels)
    - Unsupervised learning → not-annotated train data; a labeling model has to be learnt in order to annotate the test data; the set of labels is unknown befor training
  - Systems
    - Decision trees
      - Each internal node → attribute
      - Each branch of a node (attribute) → value of that attribute
      - Each leaf → class (label) – contains all data from that class

Next lecture

A. Short introduction in Artificial Intelligence (AI)

B. Solving search problems
   A. Definition of search problems
   B. Search strategies
      A. Uninformed search strategies
      B. Informed search strategies
      C. Local search strategies (Hill Climbing, Simulated Annealing, Tabu Search, Evolutionary algorithms, PSO, ACO)
      D. Adversarial search strategies

C. **Intelligent systems**
   A. Rule-based systems in certain environments
   B. Rule-based systems in uncertain environments (Bayes, Fuzzy)
   C. **Learning systems**
      A. **Decision Trees**
      B. **Artificial Neural Networks**
      C. Support Vector Machines
      D. Evolutionary algorithms
   D. Hybrid systems

Next lecture – useful information

- Chapter VI (19) of *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- Chapter 8 of *Adrian A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001*
- Chapters 12 and 13 of *C. Groşan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*
- Chapter V of *D. J. C. MacKey, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003*
- Chapter 4 of *T. M. Mitchell, Machine Learning, McGraw-Hill Science, 1997*

- Presented information have been inspired from different bibliographic sources, but also from past AI lectures taught by:
  - PhD. Assoc. Prof. Mihai Oltean – www.cs.ubbcluj.ro/~moltean
  - PhD. Assoc. Prof. Crina Groşan - www.cs.ubbcluj.ro/~cgrosan
  - PhD. Prof. Horia F. Pop - www.cs.ubbcluj.ro/~hfpop