Learning Decision Trees

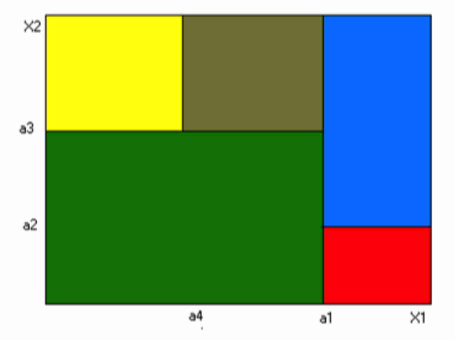
João Gama

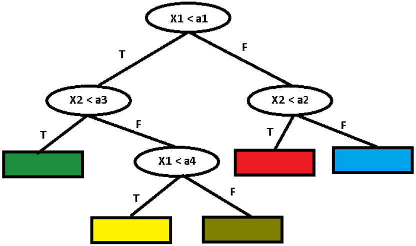
LIAAD-INESC Porto, University of Porto, Portugal
jgama@fep.up.pt

Outline

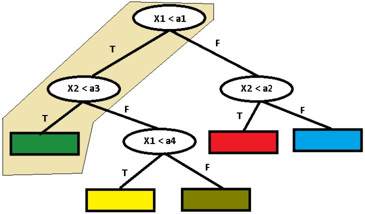
- A decision tree uses a divide-and-conquer strategy:
  - A complex problem is decomposed into simpler sub problems.
  - Recursively the same strategy is applied to the sub problems.
- The discriminant capacity of a decision tree is due to:
  - Its capacity to split the instance space into sub spaces.
  - Each sub space is fitted with a different function.
- There is increasing interest
  - CART (Breiman, Friedman, et.al.)
  - C4.5 (Quinlan)
  - Splus, Statistica, SPSS, R, ...
  - IBM IntelligentMiner, Microsoft SQL Server, ...

Partition of the Instance Space



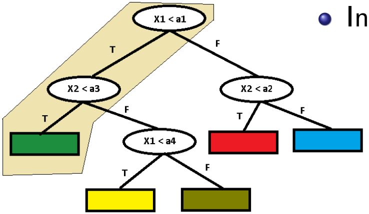


Representation of a Decision Tree



- Representation using decision trees:
  - Each decision node contains a test in one attribute
  - Each descendant branch correspond to a possible attribute-value.
  - Each terminal node (leaf) predicts a class label.
  - Each path from the root to the leaf corresponds to a classification rule.

Decision Tree Representation



- In the attribute space:
  - Each leaf corresponds to a decision region (Hyper-rectangle)
  - The intersection of the hyper-rectangles is Null
  - The union of the hyper-rectangles is the universe.

Decision Tree Representation

A Decision Tree represents a disjunction of conjunctions of restrictions in the attribute values.

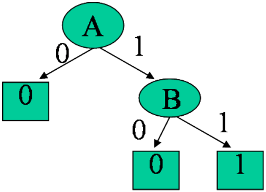
- Each branch in a tree corresponds to a conjunction of conditions.
- The set of branches are disjunct.
- DNF (disjunctive normal form)

Decision Tree Representation

Any Boolean function can be represented by a decision tree.

Example $a \wedge b$

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1



Outline

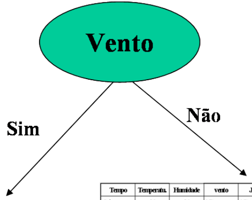
Growing a Decision Tree - The base Idea.

Tree is constructed in a top-down recursive divide-and-conquer
Input: A set of examples described by a set of attributes.
Output: A decision tree

1. Initial Tree $\leftarrow$ Empty Leaf
2. Select (might be random) one of the attributes
3. Expand the tree by adding a new branch and a leaf for each attribute-value.
4. Each example passes down to one of the new leaves, taking into account the value for the chosen attribute.
5. For each leaf
   - If all the examples are of the same class, attach that class to the leaf *(Terminal condition)*
   - Otherwise, repeat steps 2 to 5 *(Recursion)*

Illustrative Example

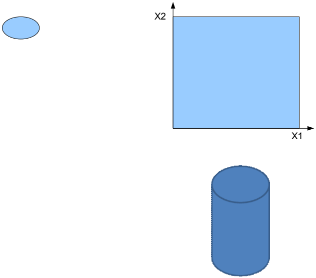
Tempo	Temperatu	Humidade	vento	Joga
Sol	85	85	Não	Não
Sol	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuva	70	96	Não	Sim
Chuva	68	80	Não	Sim
Chuva	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Sol	72	95	Não	Não
Sol	69	70	Não	Sim
Chuva	75	80	Não	Sim
Sol	75	70	Sim	Sim
Nublado	72	90	Sim	Sim
Nublado	81	75	Não	Sim
Chuva	71	91	Sim	Não



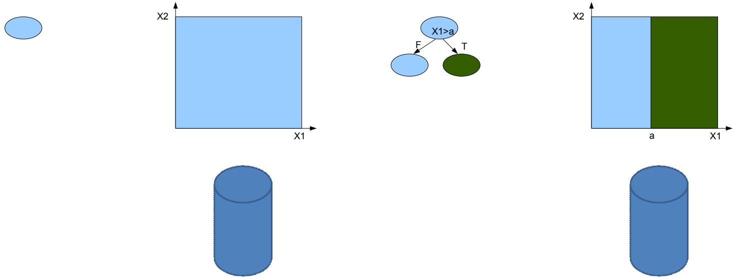
Tempo	Temperatu.	Humidade	vento	Joga
Sol	80	90	Sim	Não
Chuva	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Sol	75	70	Sim	Sim
Nublado	72	90	Sim	Sim
Chuva	71	91	Sim	Não

Tempo	Temperatu.	Humidade	vento	Joga
Sol	85	85	Não	Não
Nublado	83	86	Não	Sim
Chuva	70	96	Não	Sim
Chuva	68	80	Não	Sim
Sol	72	95	Não	Não
Sol	69	70	Não	Sim
Chuva	75	80	Não	Sim
Nublado	81	75	Não	Sim

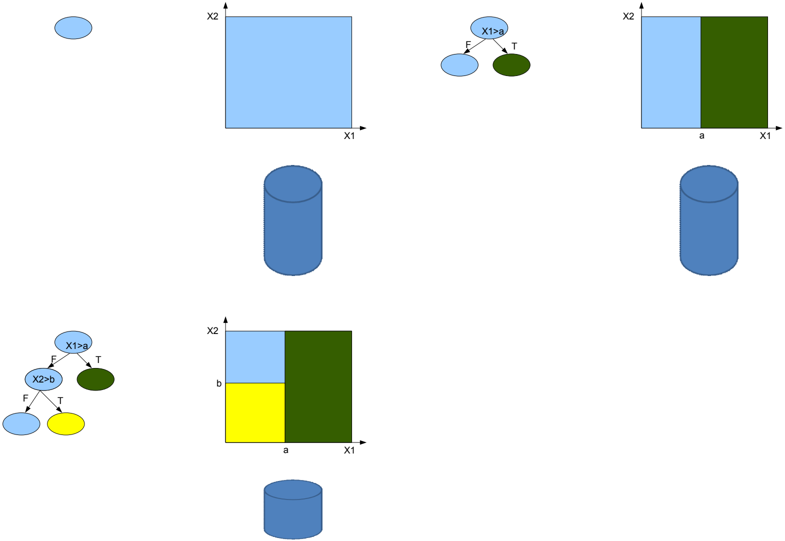
Illustrative Example



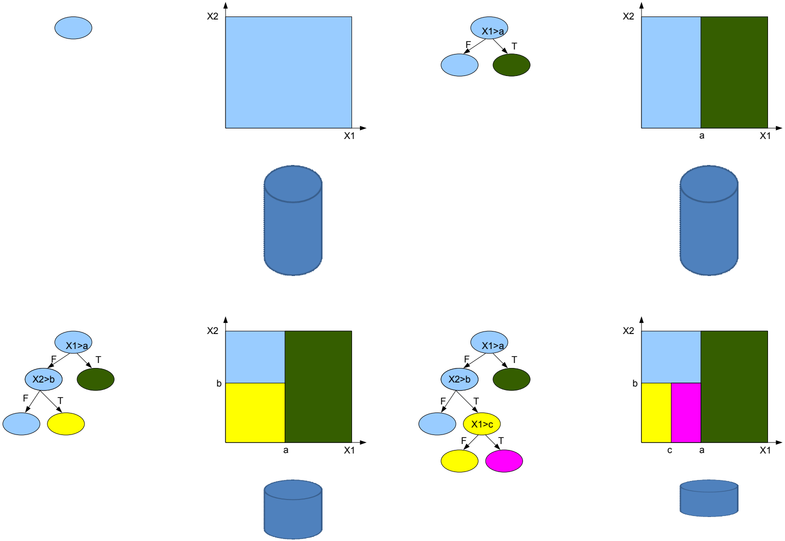
Illustrative Example



Illustrative Example



Illustrative Example



Splitting Criteria:

How to choose an attribute?

How to measure the ability of an attribute to discriminate between classes?

Many measures

There are many measures. All agree in two points:

- A split that maintains the class proportions in all partitions is useless.
- A split where in each partition all examples are from the same class has maximum utility.

Question

Which partition best discriminate the class?

Measuring the Purity of a Partition

Characterization of different strategies:

- Measure the difference given by a function based on proportions of classes between the current node and the nodes descendants.
  - Increases with the purity of the partitions.
  - Gini, entropy
- Measure the difference given by a function based on the proportions of classes among the descendant nodes.
  - increases with the disparity between the partitions.
  - Lopez de Mantaras
- independence measure:
  Measure of the degree of association between the attributes and the class.

Entropy

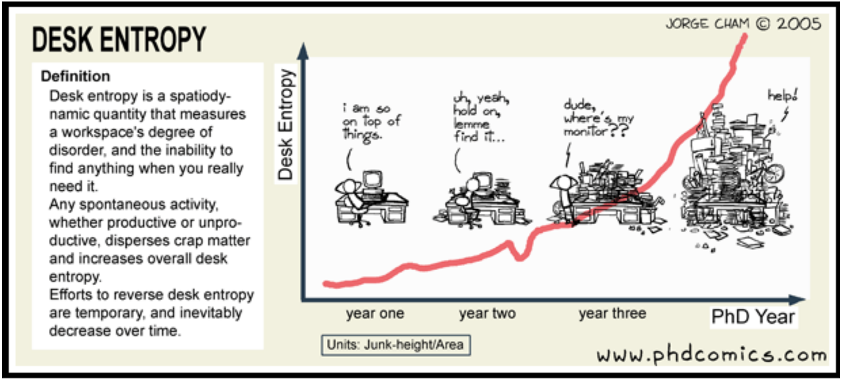
Entropy measures the degree of randomness of a random variable.
The entropy of a discrete random variable which domain is $\{V_1, \ldots V_i\}$:

$$H(X) = -\sum_{j=1}^{i} p_j log_2(p_j)$$

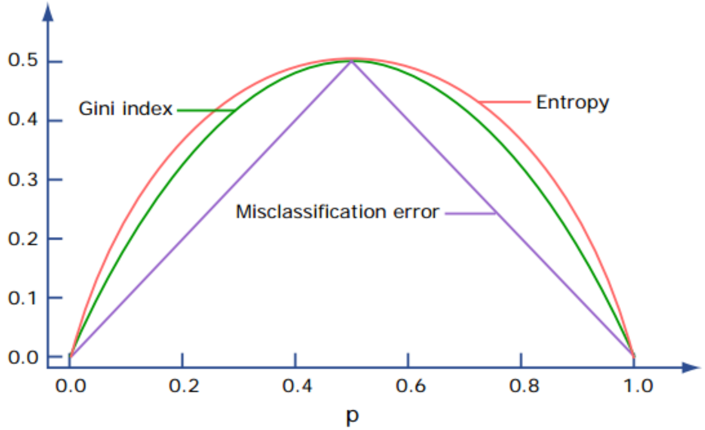
where p_j is the probability of observing value V_j.
Properties:

- $H(X) \geq 0$
- Maximum: $max(H(X)) = log_2 i$ iff $p_i = p_j$ for each $i, j, i \neq j$.
- Minimum: $H(X) = 0$ if there is i such that $p_i = 1$ assuming $0 * log_2 0 = 0$.

Entropy


DESK ENTROPY
JORGE CHAM © 2005
Definition
Desk entropy is a spatiodynamic quantity that measures a workspace's degree of disorder, and the inability to find anything when you really need it.
Any spontaneous activity, whether productive or unproductive, disperses crap matter and increases overall desk entropy.
Efforts to reverse desk entropy are temporary, and inevitably decrease over time.
Desk Entropy
i am so on top of things.
uh, yeah, hold on, lemme find it...
dude, where's my monitor??
help!
year one
year two
year three
PhD Year
Units: Junk-height/Area
www.phdcomics.com

Entropy



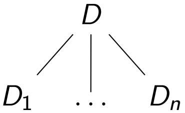
Node impurity measures for two-class classification, as a function of the proportion p in class 2. Cross-entropy has been scaled to pass through (0.5, 0.5).

Let p_i be the probability that an arbitrary example in D belongs to class C_i, estimated by $|C_i, D|/|D|$

Expected information (entropy) needed to classify an example in D: $H(D) = -\sum p_i \times log_2(p_i)$

Information needed (after using A to split D into v partitions) to classify D: $H_A(D) = \sum_1^v \frac{|D_j|}{|D|} \times H(D_j)$

Information gained by branching on attribute A: $Gain_A = H(D) - H_A(D)$.



Decision Trees and Entropy

Entropy is used to estimate the randomness or difficulty to predict, of the target attribute.

Entropy

- Given a set of classified examples, which attribute to chose for splitting test?
- The values of an attribute define partitions of the set of examples.
- Consider the set of partitions defined by one attribute
- Compute the entropy of each partition;
- Choose the attribute that most reduce the entropy.

Decision Trees and Entropy

Growing a decision tree is guided by reducing the entropy, that is the randomness or difficulty to predict the class.

Computing the Information Gain

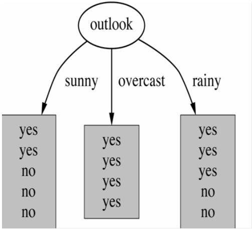
Tempo	Temperatu	Humidade	vento	Joga
Sol	85	85	Não	Não
Sol	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuva	70	96	Não	Sim
Chuva	68	80	Não	Sim
Chuva	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Sol	72	95	Não	Não
Sol	69	70	Não	Sim
Chuva	75	80	Não	Sim
Sol	75	70	Sim	Sim
Nublado	72	90	Sim	Sim
Nublado	81	75	Não	Sim
Chuva	71	91	Sim	Não

Computing the Information Gain

Before Splitting: Computing the Entropy of the class

- p(yes) = 9/14
- p(no) = 5/14
- Info(play) =
  $= -9/14 \times log_2(9/14) - 5/14 \times log_2(5/14) = 0.940$ bits

Entropy of a Nominal Attribute



- $p(yes|outlook = sunny) = 2/5$
- $p(no|outlook = sunny) = 3/5$
- $H(play|outlook = sunny) = -2/5 \times log_2(2/5) - 3/5 \times log_2(3/5) =$ 0.971 bits
- $H(play|outlook = overcast) = 0.0 bits$
- $H(play|outlook = rainy) = 0.971 bits$

Information of one attribute

Weighted sum of the entropy of all partitions:
$\text{Info}(\text{outlook}) = 5/14 \times 0.971 + 4/14 \times 0 + 5/14 \times 0.971 = 0.693$ bits

Information Gain of an Attribute

Information Gain

Gain(outlook) = 0.940 - 0.693 = 0.247 bits

Information Gain of a Continuous Attributes

- Continuous Attributes: domain is a subset of R.
- A split-test in a continuous attributes generates two partitions in the set of examples:
  - Set of examples where $Att_i < reference\_value$
  - Set of examples where $Att_i \geq refeence\_value$
- Example:
  - Temperature < 36.5;
  - Temperature ≥ 36.5.

Additional problem:

How to chose the *reference_value*?

Computing the Entropy of a continuous attribute

Temperatu.	Joga
64	Sim
65	Não
68	Sim
69	Sim
70	Sim
71	Não
72	Não
72	Sim
75	Sim
75	Sim
80	Não
81	Sim
83	Sim
85	Não

1. Sort the examples, ascending, by the attribute-values
2. Each mean value of two consecutive different values is a candidate *reference_value*
3. Compute the Entropy of the partitions obtained by each candidate *reference_value*
4. Chose the candidate *reference_value* with minimum Entropy.

Fayyard e Irani (1993) have shown that between all possible *reference_values* those that minimize entropy are between examples from different classes.

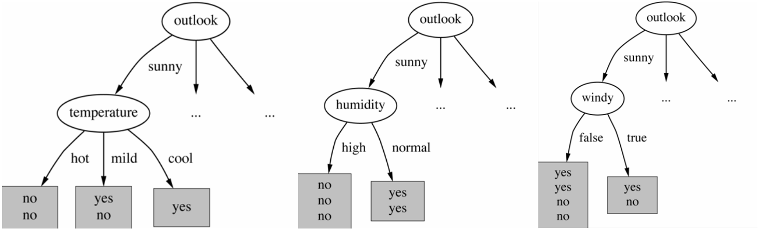
Computing the Entropy of a continuous attribute

Consider the *reference_point temperature = 70.5*.
How to compute the information gain of that partition?

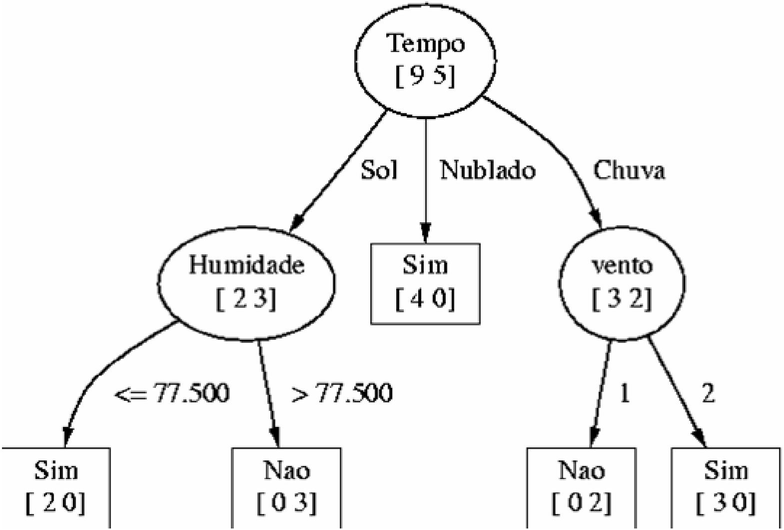
Temperatu.	Joga
64	Sim
65	Não
68	Sim
69	Sim
70	Sim
71	Não
72	Não
72	Sim
75	Sim
75	Sim
80	Não
81	Sim
83	Sim
85	Não

- $p(yes|temperatura < 70.5) = 4/5$
- $p(no|temperatura < 70.5) = 1/5$
- $p(yes|temperatura \geq 70.5) = 5/9$
- $p(no|temperatura \geq 70.5) = 4/9$
- $Info(joga|temperatura < 70.5) = -4/5 log_2(4/5) - 1/5 log_2(1/5) = 0.721$ bits
- $Info(joga|temperatura \geq 70.5) = -5/9 log_2(5/9) - 4/9 log_2(4/9) = 0.991$ bits
- $Info(temperatura) = 5/14 * 0.721 + 9/14 * 0.991 = 0.895$ bits
- $Ganho(temperatura) = 0.940 - 0.895 = 0.045$ bits

Growing the tree - Recursion



The final tree



Comparing Attribute Selection Measures

The three measures, in general, return good results but

- Information gain:
  biased towards multivalued attributes;
- Gain ratio:
  tends to prefer unbalanced splits in which one partition has examples of a single class;
- Gini index:
  - biased to multivalued attributes
  - has difficulty when # of classes is large
  - tends to favor tests that result in equal-sized partitions and purity in both partitions

Discussion

- The problem of learning the minimum (in the number of nodes) decision tree consistent with a set of examples is NP hard.
- The usual approach use heuristic search
    - One step lookahead
    - without backtracking

Two main problems:

- Which attribute to select for split-test ?
- When stop growing the tree?

Outline

Overfitting

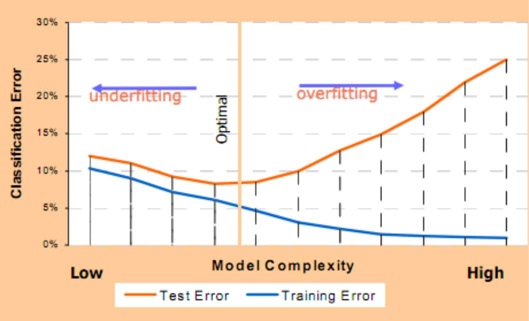
- The recursive-partitioning algorithm can generate trees with a perfect fit to the training set.
- Expanding the tree reduces the error in the training set,
  - The degrees of freedom increase linearly with the number of examples;
  - but, expanding the tree, reduces the number of examples available at each node.
  - decisions (e.g. chose a split-test) have less statistical support.
- Growing the tree too much, increases the error on independent test set.

Why?

- Noisy data,
- Over-search.

Overfiting

Comparison between training error and holdout error for increasing number of nodes in a decision tree:



Overfitting

Occam's razor: preference for simplicity.

- There are less simple hypothesis than complex ones;
- If a simpler hypothesis explain the data, it is less probable that it happens by chance;
- Complex hypothesis can explain data only by chance.

Simplifying the tree

Two possibilities:

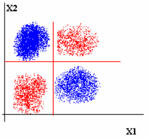
- **Pre-pruning** - Early stop growing the tree;
- **Pos-pruning** - Grow a large tree and prune it back.
  - *Growing and pruning is slower but more reliable*, Quinlan, 1988.

Pre-pruning

When to stop dividing the examples?

- All examples belong to the same class;
- All examples have the same attribute-values (but different classes).
- The number of examples is less than a minimum value.
- (?) The merit of all possible split-tests is low.

XOR problem: not linearly separable:



Post-Pruning

- **Reduced-Error Pruning**: minimize the error in an independent validation set (used in J48);
- **Error based Pruning**: minimize *pessimistic* training error estimates, (used in C4.5);
- **Cost Complexity Pruning**: minimize training error and number of nodes in the tree (used in rpart);

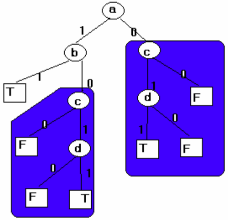
Outline

Advantages of Decision Trees

- Nonparametric method
  Does not assume any particular distribution for the data. Can build models for any function given a sufficient number of training examples.
- The structure of the decision tree is independent of the scale the variables. Monotone transformations of the variables ($log\ x, 2 * x, \ldots$) do not alter the structure of the tree.
- High degree of interpretability A complex decision (predict the class value) is decomposed into a succession of elementary decisions.
- It is efficient in building models: Average complexity $O(n\ log(n))$.
- Robust to the presence of extreme points and attributes redundant or irrelevant. Selection mechanism attributes.

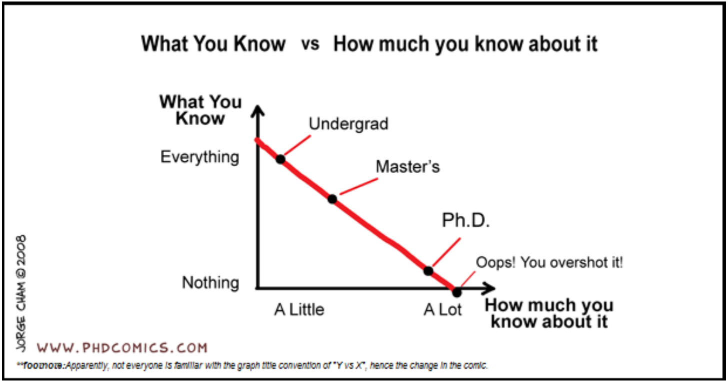
Disadvantages of Decision Trees

- Instability
  Small perturbations in the training set can generate large changes in the decision tree.
- Missing values
- Fragmentation of concepts;
- sub-tree replication.



Outline

- Online: http://www.Recursive-Partitioning.com/
- Tom Mitchell Machine Learning (chap.3) MacGrawHill, 1997
- Quinlan, R. *C4.5 Programs for Machine Learning* Morgan Kaufmann Publishers, 1993
- L.Breiman, J.Friedman, R.Olshen, C.Stone *Classification and Regression Trees* Wadsworth, 1984



Would you like to learn more? Wait for ECDII ...