Decision Tree Induction

Non-metric Methods

- Numerical Attributes
  - Nearest-neighbor -- distance
  - Neural networks: two similar inputs leads to similar outputs
  - SVMs: Dot Product

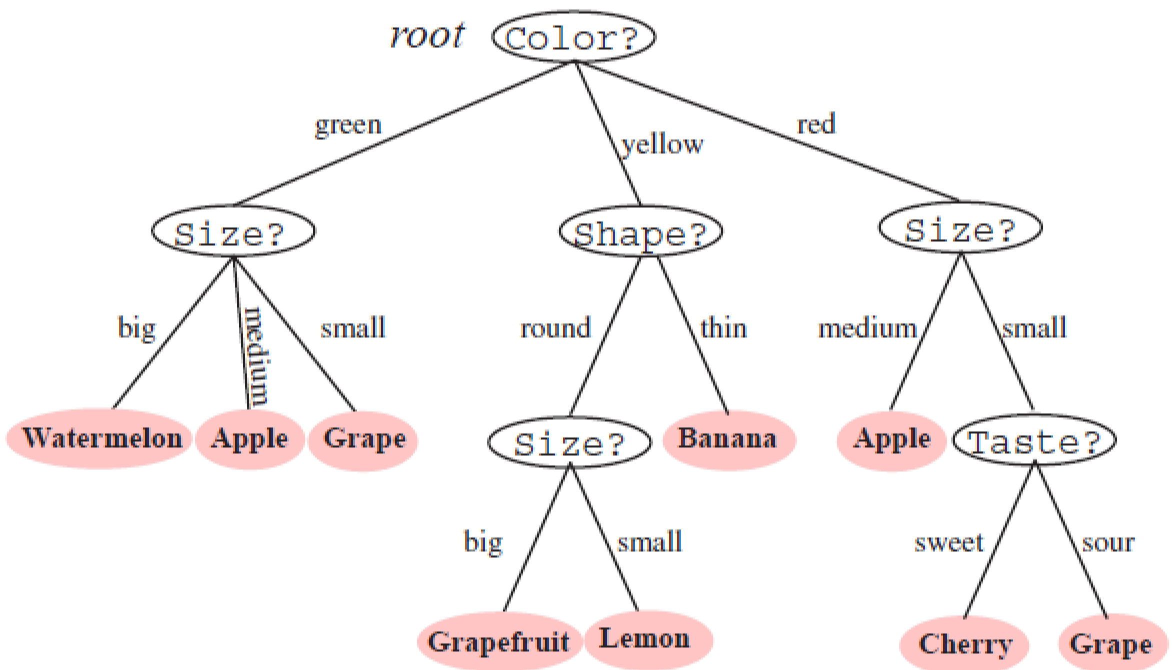
Non-metric data

- Nominal attributes
- Color, taste
- Strings: DNA
-

- Probability based
  - Naïve Bayes
- Rule based
  - Decision trees

Decision Tree

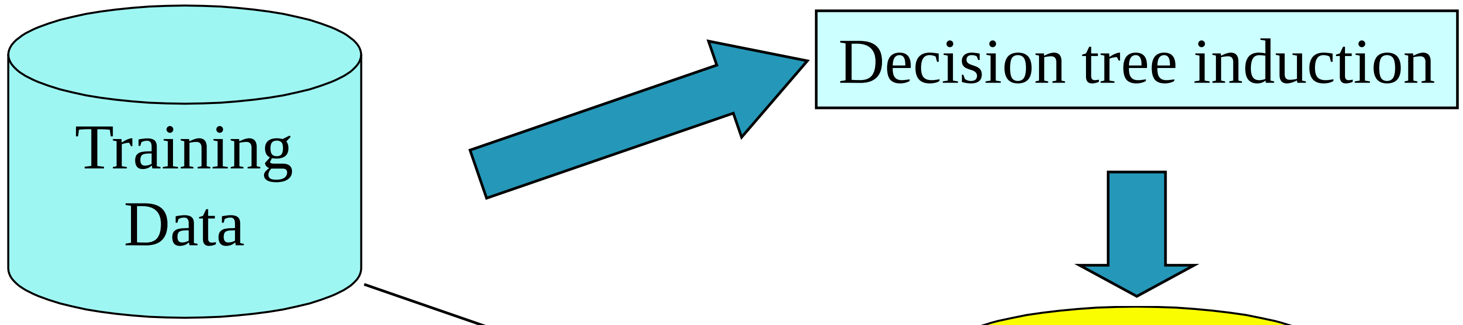
- Rules in the form of a hierarchy.



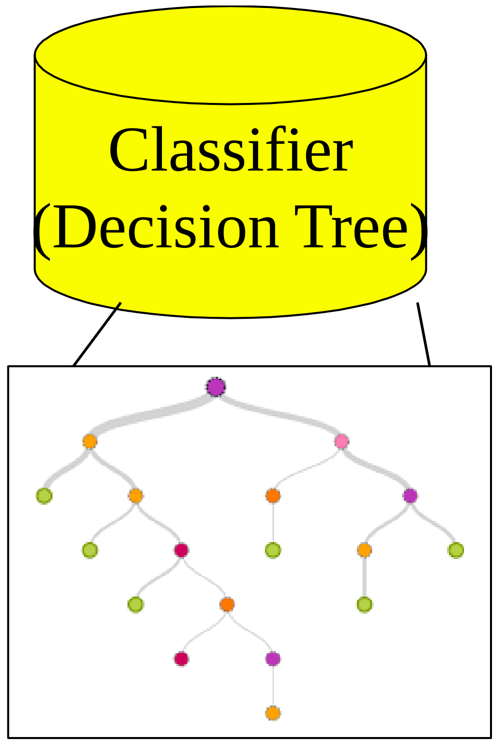
- Why are decision trees so popular?

- Why are decision trees so popular?
- Interpretability
  - You can give human understandable explanation for the decision being made.

We need to work with a training set



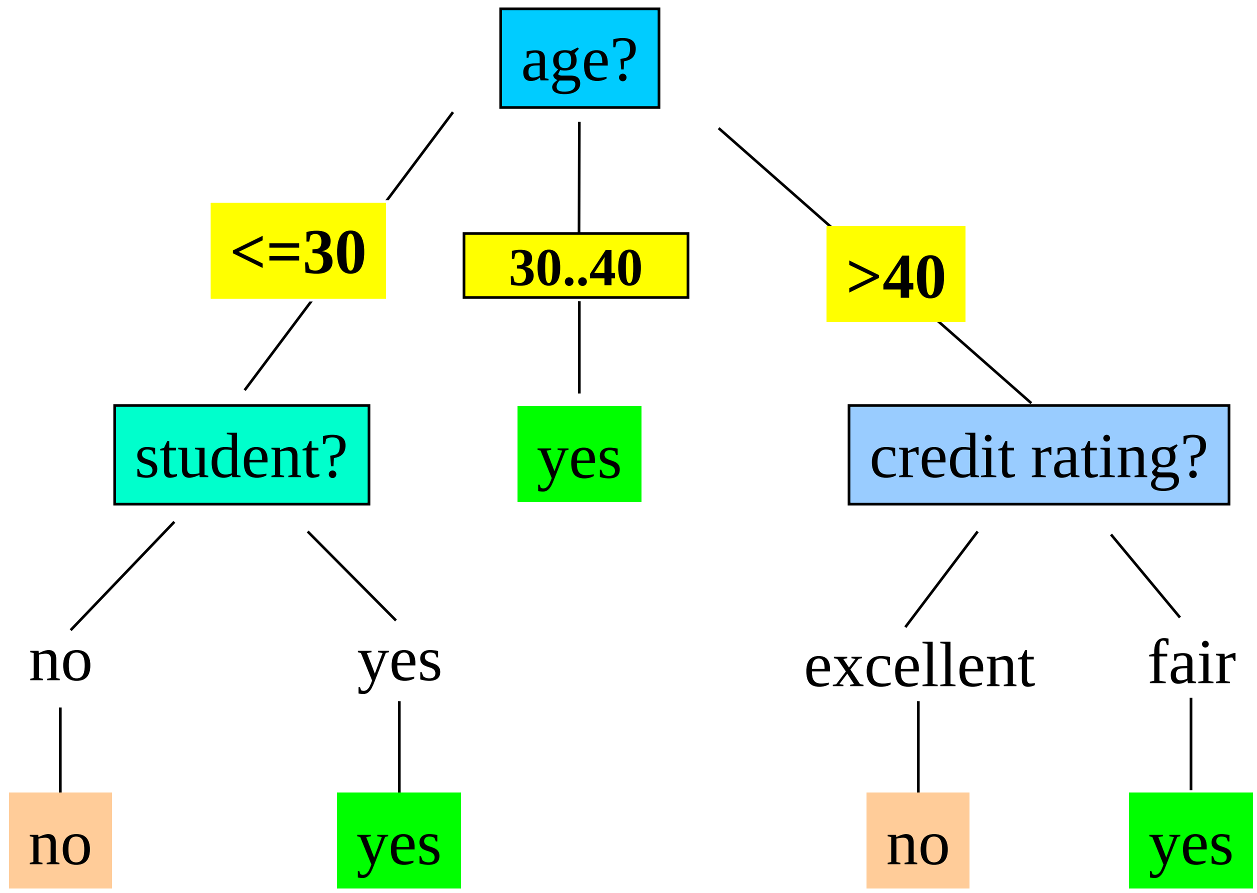
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no



You need to work with a training set

age	income	student	cred_rati	buys_comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Output: A Decision Tree for “*buys_computer*”



How to create a Decision Tree

- From root to leaves it is created
- For root, choose the attribute to be used for testing. This gives child nodes for the root.
- For each child, again choose the attribute to be tested. ….

Issues

- Criteria for choosing an attribute?
- You can achieve 100% accuracy with training set?!
  - Overfitting
- When you stop building the tree?

- Are there various types of DT induction methods?? ID3, C4.5 and CART.

Decision tree induction

- They adopt a greedy (i.e., nonbacktracking), top-down recursive divide-and-conquer approach.

- Node → subset of training patterns
- Root → training set.
- Leaf → class label.

Impurity measures

- Entropy impurity (information impurity)

$$i(N) = -\sum_j P(\omega_j) \log_2 P(\omega_j)$$

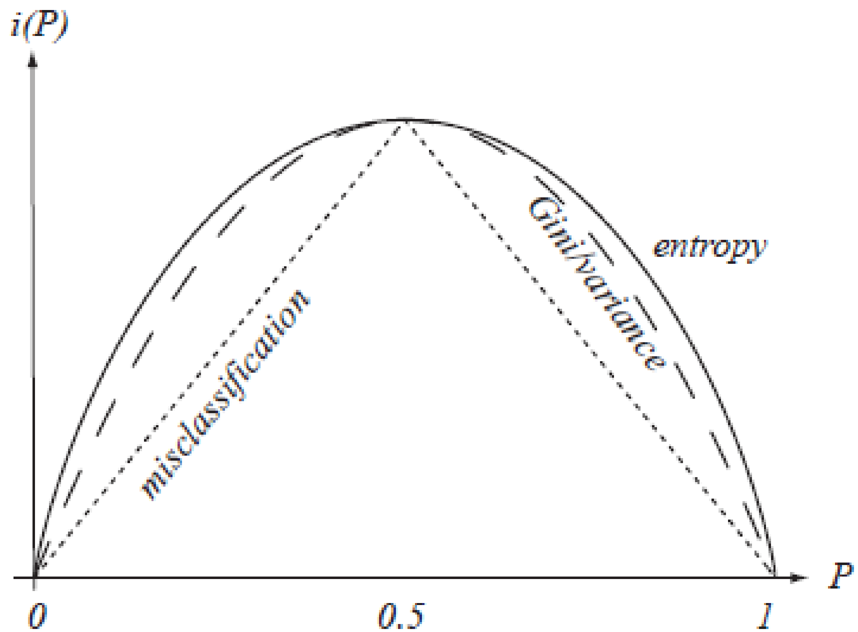
- Gini impurity (variance impurity)

$$i(N) = 1 - \sum_j P^2(\omega_j)$$

- Misclassification impurity

$$i(N) = 1 - \max_j P(\omega_j)$$

For a two category case



Which test?

- That which drops the impurity greater.
  - Try to become pure quickly.

$$\Delta i(N) = i(N) - \big(P_L\, i(N_L) + (1 - P_L)\, i(N_R)\big)\,,$$

Which test?

- That which drops the impurity greater.
  - Try to become pure quickly.

$$\Delta i(N) = i(N) - \big(P_L\, i(N_L) + (1 - P_L)\, i(N_R)\big),$$

where N_L and N_R are the left and right descendent nodes,
$i(N_L)$ and $i(N_R)$ their impurities,
and P_L is the fraction of patterns at node N that will go to N_L

Which test?

- That which drops the impurity greater.
  - Try to become pure quickly.

$$\Delta i(N) = i(N) - \big(P_L\, i(N_L) + (1 - P_L)\, i(N_R)\big),$$

where N_L and N_R are the left and right descendent nodes,
$i(N_L)$ and $i(N_R)$ their impurities,
and P_L is the fraction of patterns at node N that will go to N_L

Then the “best” test value s is the choice for T that maximizes $\Delta i(T)$.

Which test?

- That which drops the impurity greater.
  - Try to become pure quickly.

$$\Delta i(N) = i(N) - \left(P_L\, i(N_L) + (1 - P_L)\, i(N_R)\right),$$

Which test?

- That which drops the impurity greater.
  - Try to become pure quickly.

$$\Delta i(N) = i(N) - \left(P_L \, i(N_L) + (1 - P_L) \, i(N_R)\right),$$

Information gain

- This is drop in entropy impurity !!
- For an attribute A, often written as $Gain(A)$

Gain(age) ??

age	income	student	cred_rati	buys_comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

(yes, no) = (9, 5)

Gain(age) ??

age	income	student	cred_rati	buys_comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

(yes, no) = (9, 5)

$$i(root) = I(9,5)$$
$$I(9,5) = -\frac{9}{14}\log\frac{9}{14} - \frac{5}{14}\log\frac{5}{14}$$
$$= 0.94$$

age	Yes	No	Impurity
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

$$\Delta i(age) = 0.94 - \left(\frac{5}{14}I(2,3) + \frac{4}{14}\mathrm{I}(4,0) + \frac{5}{14}I(3,2)\right)$$
$$= 0.69$$

We call this *Gain(age) = 0.69*.

For other attributes, their GAIN

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

- So we choose age as the splitting attribute.

- Similarly one can use other impurity measures

Gini Index (IBM IntelligentMiner)

- If a data set T contains examples from n classes, gini index, *gini*(T) is defined as

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class j in T.

- If a data set T is split into two subsets T_1 and T_2 with sizes N_1 and N_2 respectively, the *gini* index of the split data contains examples from n classes, the *gini* index *gini*(T) is defined as

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

- The attribute provides the smallest $gini_{split}(T)$ is chosen to split the node (*need to enumerate all possible splitting points for each attribute*).

- But, there is one drawback with this approach!

- A split with large branching factor is often chosen.
  - So, telephone number is chosen.

$$\Delta i(s) = i(N) - \sum_{k=1}^{B} P_k i(N_k)$$

$$\sum_{k=1}^{B} P_k = 1.$$

So, we penalize large branching factors

- This is called ***gain ratio*** (very often used with *information gain*).

$$\Delta i_B(s) = \frac{\Delta i(s)}{-\sum_{k=1}^{B} P_k \log_2 P_k}.$$

- Branching factor is more, the denominator is more.

Extracting Classification Rules from Trees

- Represent the knowledge in the form of IF-THEN rules
- One rule is created for each path from the root to a leaf
- Each attribute-value pair along a path forms a conjunction
- The leaf node holds the class prediction
- Rules are easier for humans to understand
- Example

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* = “*yes*”

IF *age* = “>40” AND *credit_rating* = “*fair*” THEN *buys_computer* = “*no*”

Avoid Overfitting in Classification

- The generated tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Result is in poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: Remove branches from a “fully grown” tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the “best pruned tree”

Classification in Large Databases

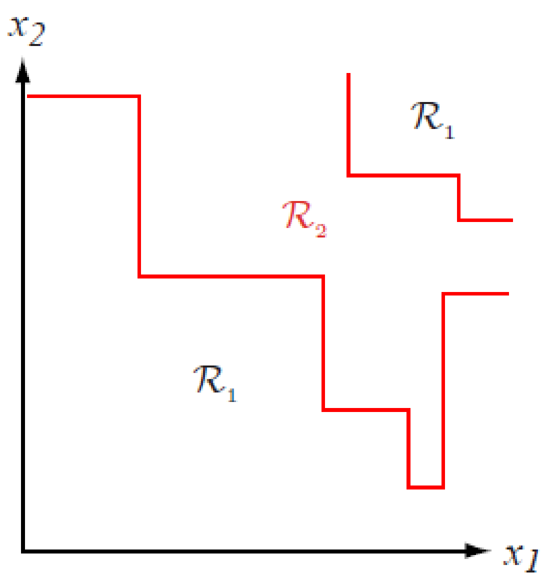
- Classification—a classical problem extensively studied by statisticians and machine learning researchers
- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed
- Why decision tree induction in classification?
  - relatively faster learning speed (than other classification methods)
  - convertible to simple and easy to understand classification rules
  - can use SQL queries for accessing databases
  - comparable classification accuracy with other methods

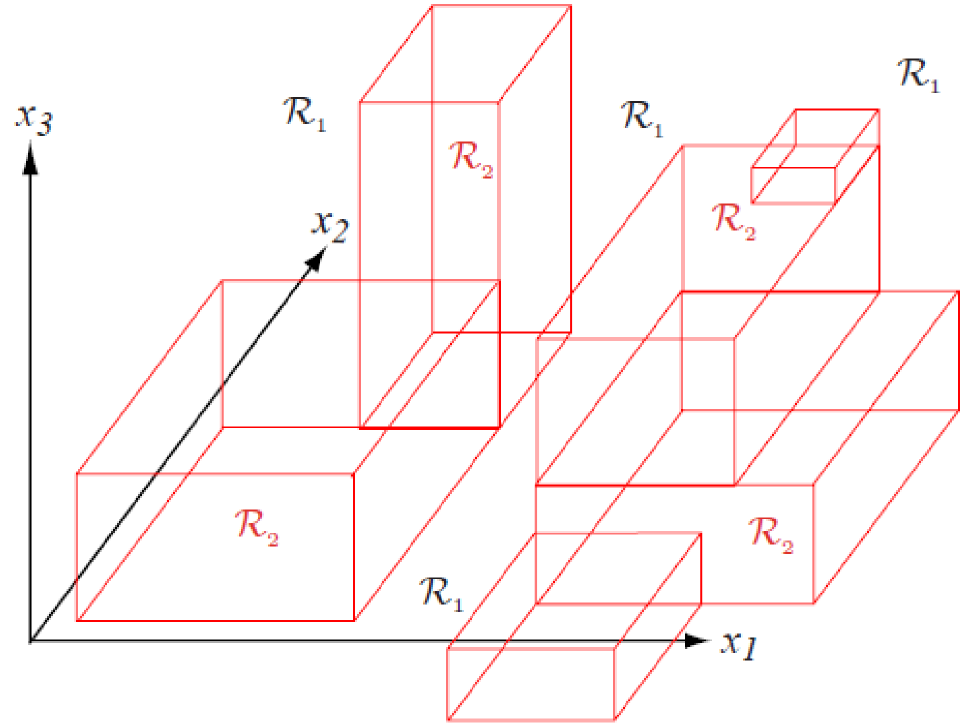
Scalable Decision Tree Induction Methods in Data Mining Studies

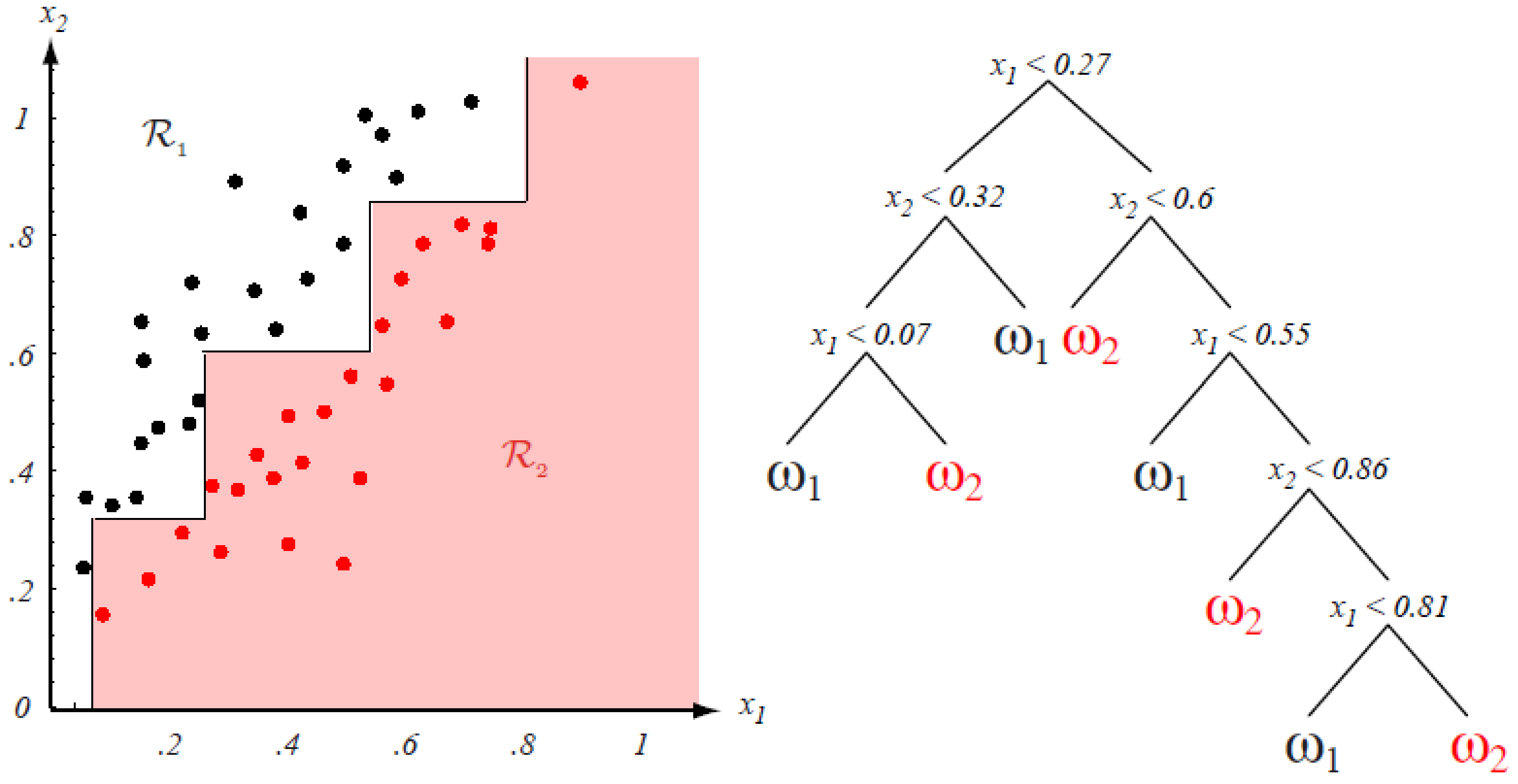
- SLIQ (EDBT'96 — Mehta et al.)
  - builds an index for each attribute and only class list and the current attribute list reside in memory
- SPRINT (VLDB'96 — J. Shafer et al.)
  - constructs an attribute list data structure
- PUBLIC (VLDB'98 — Rastogi & Shim)
  - integrates tree splitting and tree pruning: stop growing the tree earlier
- RainForest (VLDB'98 — Gehrke, Ramakrishnan & Ganti)
  - separates the scalability aspects from the criteria that determine the quality of the tree
  - builds an AVC-list (attribute, value, class label)

Drawbacks

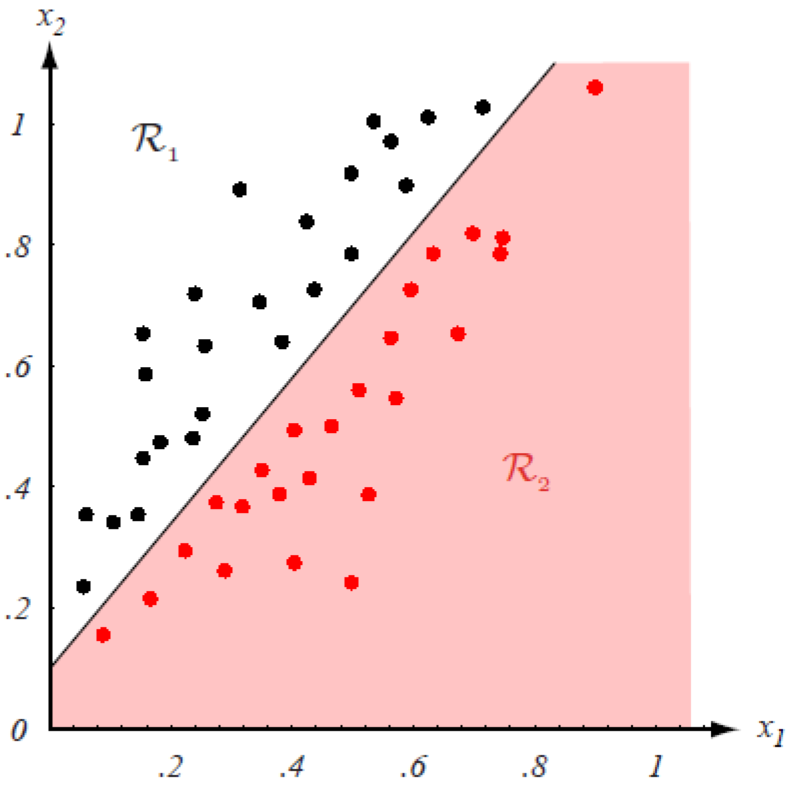
- What we discussed are axis parallel
- For continuous valued attributes cut-points can be found.
  - Can be discretized (CART does).


x_2
$\mathcal{R}_1$
$\mathcal{R}_2$
$\mathcal{R}_1$
x_1

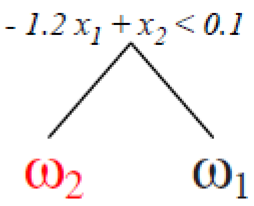

x_3
$\mathcal{R}_1$
$\mathcal{R}_2$
$\mathcal{R}_1$
$\mathcal{R}_1$
$\mathcal{R}_2$
x_2
$\mathcal{R}_2$
$\mathcal{R}_2$
$\mathcal{R}_1$
x_1



x_2
1
.8
.6
.4
.2
0
.2
.4
.6
.8
1
x_1
$\mathcal{R}_1$
$\mathcal{R}_2$
$x_1 < 0.27$
$x_2 < 0.32$
$x_2 < 0.6$
$x_1 < 0.07$
ω_1
ω_2
$x_1 < 0.55$
ω_1
ω_2
ω_1
$x_2 < 0.86$
ω_2
$x_1 < 0.81$
ω_1
ω_2



x2
1
.8
.6
.4
.2
0
.2
.4
.6
.8
1
x1
R1
R2



- 1.2 x1 + x2 < 0.1
ω2
ω1