

Decision Trees



Zhang Changshui
Dept. of Automation
Tsinghua University



Problems

- Task: Classify the fruit.
- Fruit: color: red, green, yellow, ...
- size: small, medium, big
- shape: round, thin
- taste: sweet, sour



Problems

- Classification problem involving **nominal** data.
- Discrete
- No natural notion of similarity.
- No order, in general.
- Fruit: color: red, green, yellow, ...
- size: small, medium, big
- shape: round, thin
- taste: sweet, sour



Representation

- Lists of attributes instead of vectors of real numbers.
- 4-tuple: {red, round, sweet, small}



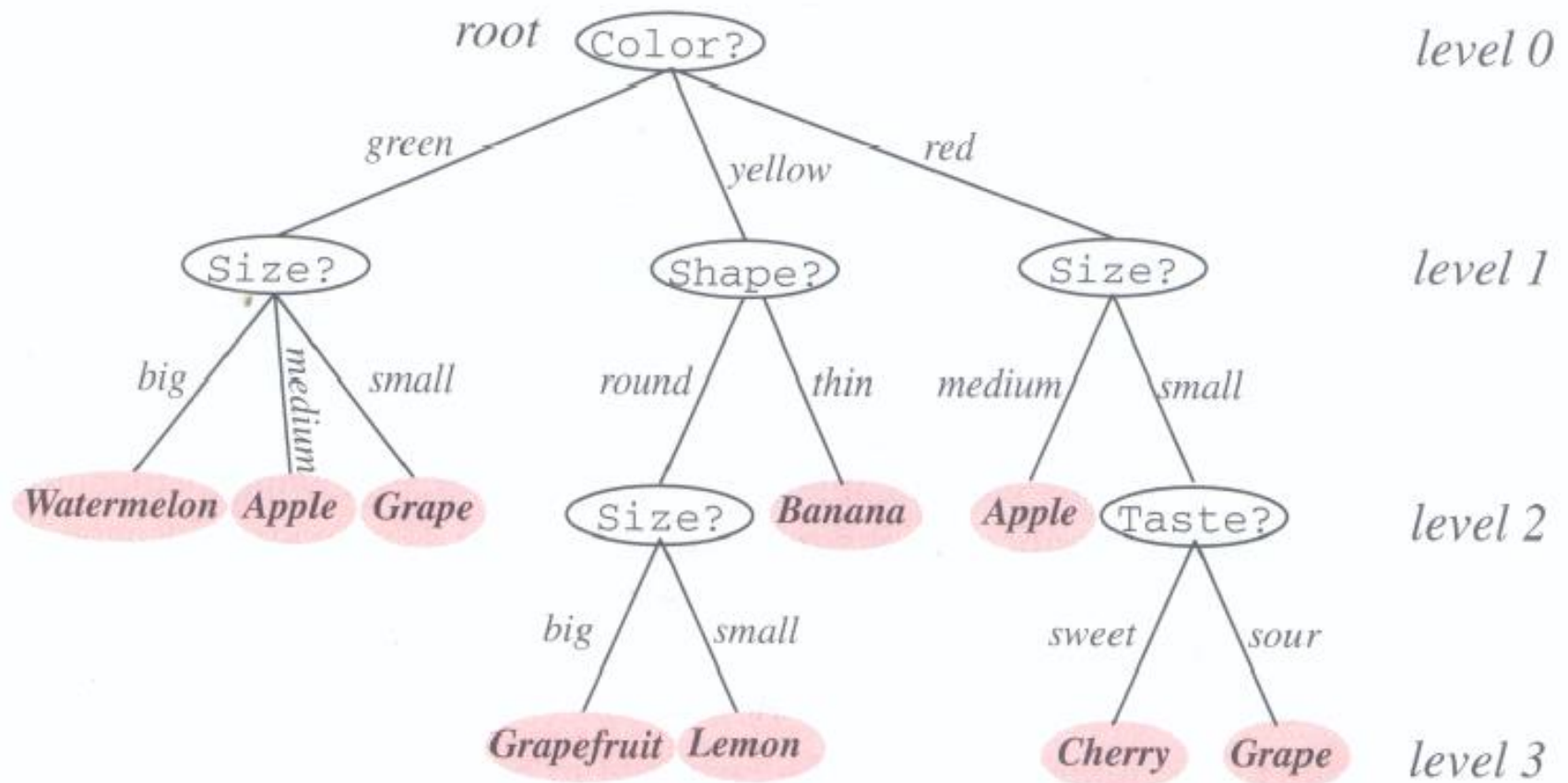
Problems

- How can we best use such nominal data for classification?
- How can we efficiently learn categories using such nonmetric data?



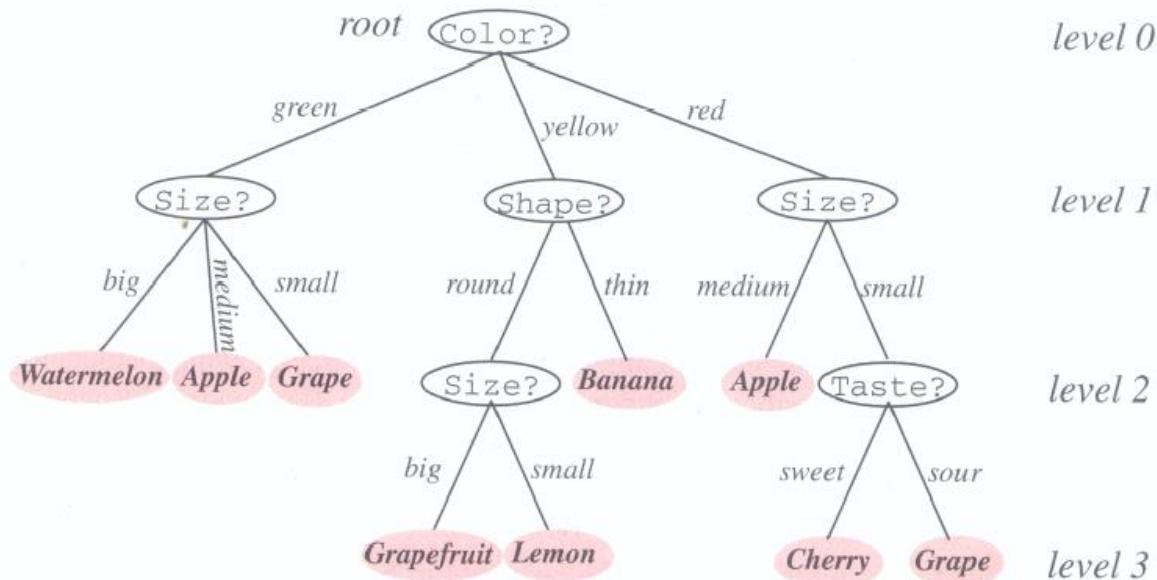
Training data

- Banana: yellow, thin, medium, sweet
- Watermelon: green, round, big, sweet,
- Banana: yellow, thin, medium, sweet
- Grape: green, round, small, sweet
- Grape: red, round, small, sour
- ...



Decision tree representation

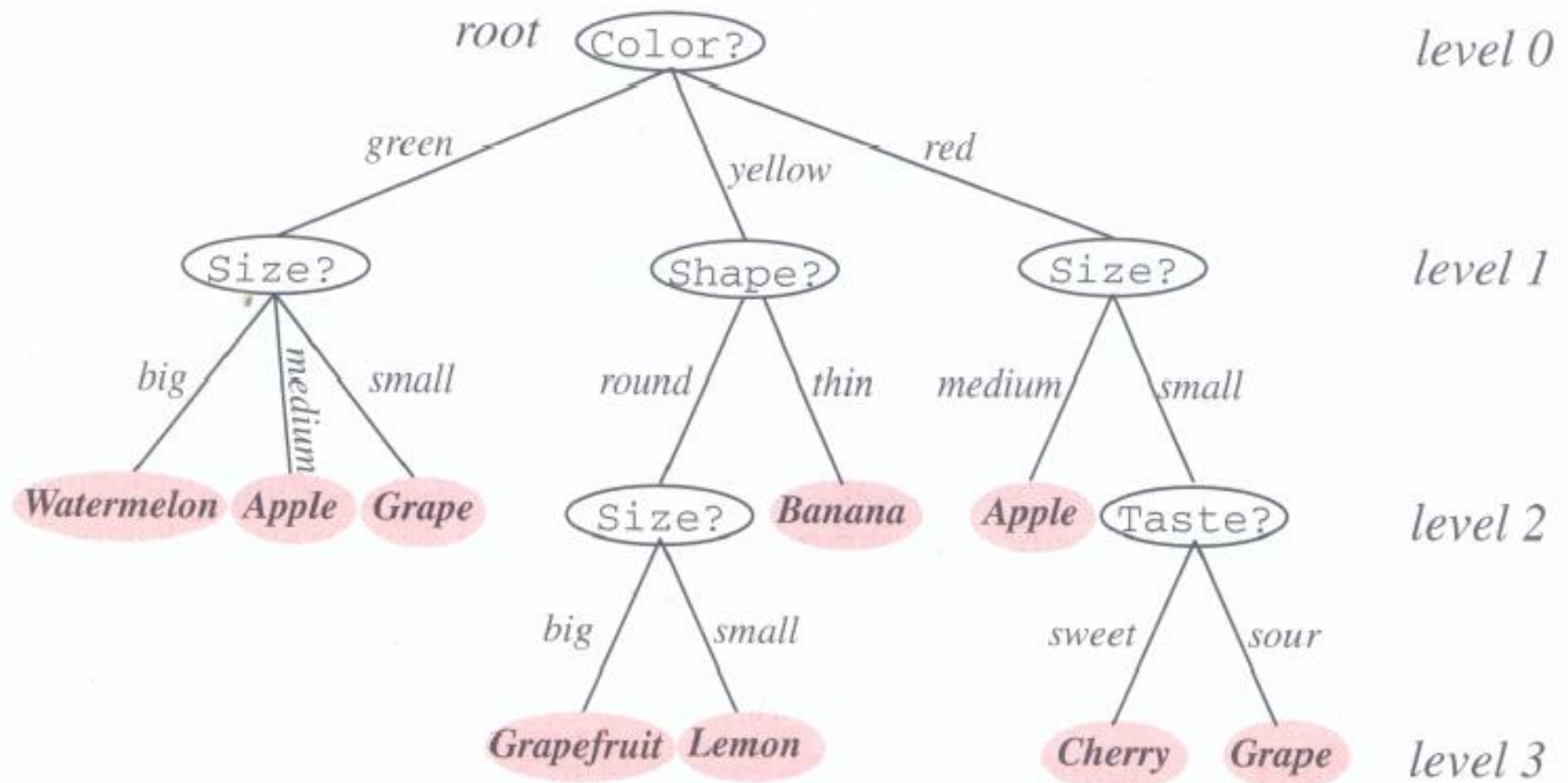
- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification





CART (classification and regression trees)

- A general framework:
- Create or grow a decision tree using training data
- Decision tree will progressively split the set of training examples into smaller and smaller subsets
- Stop splitting if each subset is pure
- Or accept an imperfect decision





Training data

- Banana: yellow, thin, medium, sweet
- Watermelon: green, round, big, sweet,
- Banana: yellow, thin, medium, sweet
- Grape: green, round, small, sweet
- Grape: red, round, small, sour
- ...



Problems

- Should the properties be restricted to binary-valued or allowed to be multivalued?
- Which property should be tested at a node?
- When should a node be declared a leaf?



Problems

- If the tree becomes “too large”, how can it be made smaller and simpler, that is, pruned?
- If a leaf node is impure, how should the category label be assigned?
- How should missing data be handled?

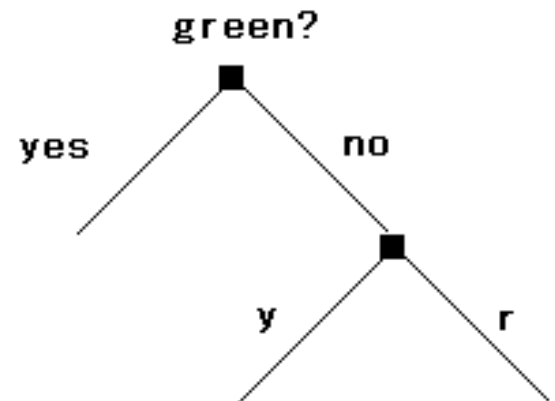
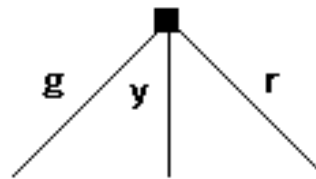


Number of splits

- Should the properties be restricted to binary-valued or allowed to be multivalued?

Number of splits

- The number of splits at a node is related to the property tested at the node.
- Every decision can be represented using just

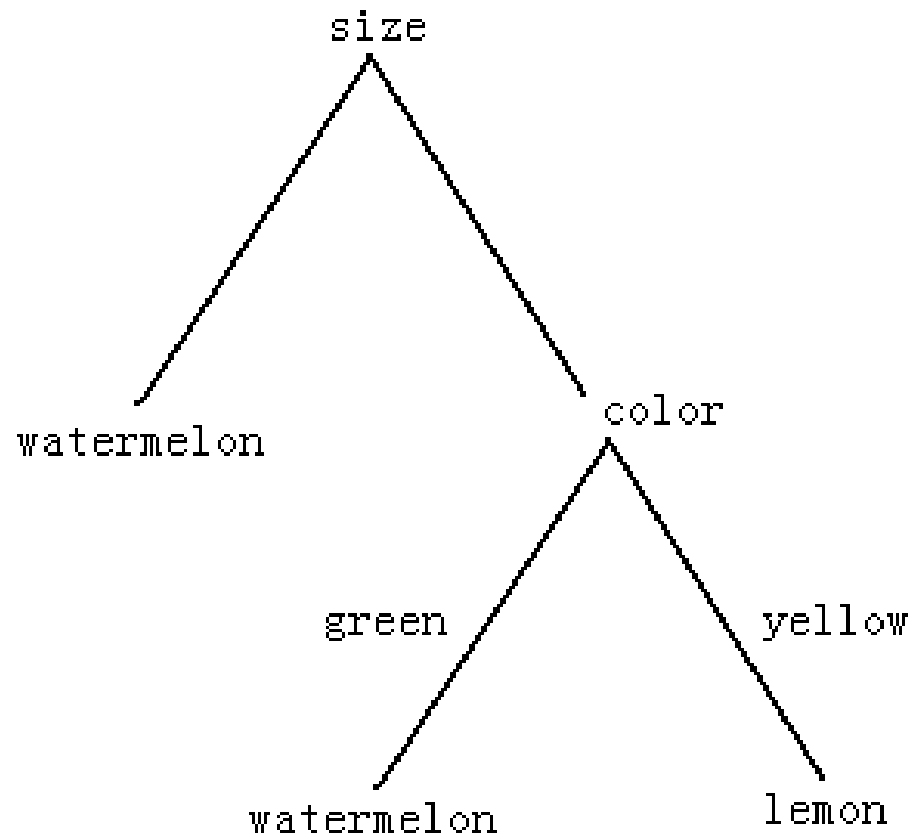
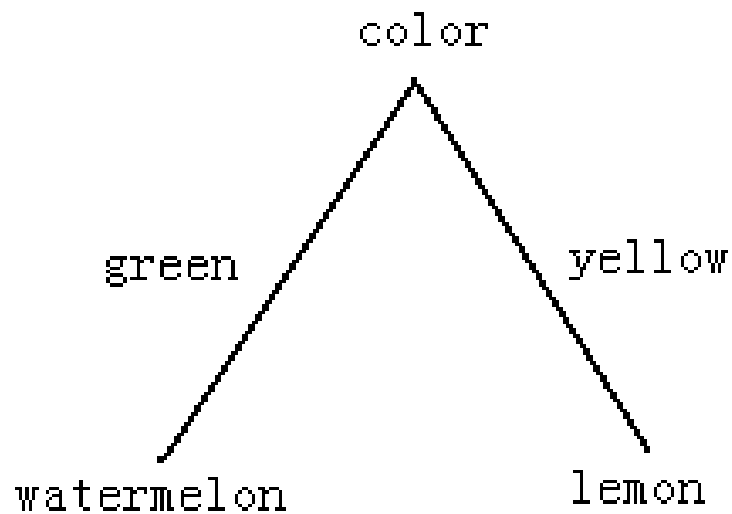




Query selection and node impurity

- Which property should be tested at a node?
- Fundamental principle: simplicity
 - We prefer decisions that lead to a simple, compact tree with few nodes
- We seek a property query T at each node N that makes the data reaching the immediate descendent nodes as “pure” as possible
- Purity – Impurity

Query selection and node impurity





How to measure impurity?



Impurity

- Entropy impurity (is frequently used)

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



Example: a simple tree

w1 (black)	
x1	x2
.15	.83
.09	.55
.29	.35
.38	.70
.52	.48
.57	.73
.73	.73
.47	.06

w2 (red)	
x1	x2
.10	.29
.08	.15
.23	.16
.70	.19
.62	.47
.91	.27
.65	.90
.75	.36(.32)



Example: a simple tree

- $$\begin{aligned} i(N_{root}) &= -\sum_j P(w_j) \log_2 P(w_j) \\ &= -[0.5 \log_2 0.5 + 0.5 \log_2 0.5] \\ &= 1 \end{aligned}$$



Impurity

- Gini impurity (Duda prefers Gini impurity)

$$i(N) = \sum_{i \neq j} P(w_i)P(w_j) = 1 - \sum_j P^2(w_j)$$

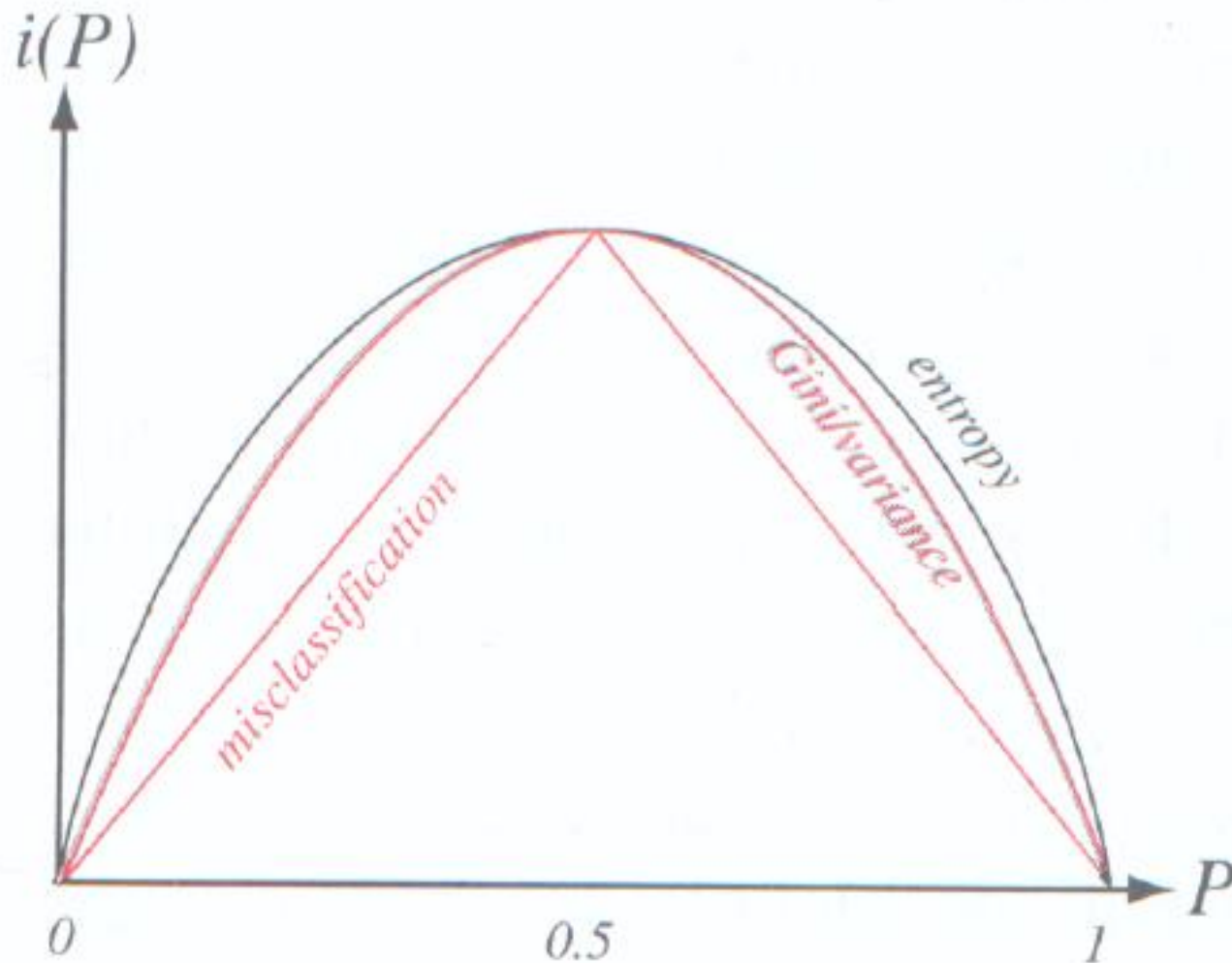


Impurity

- Misclassification impurity

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

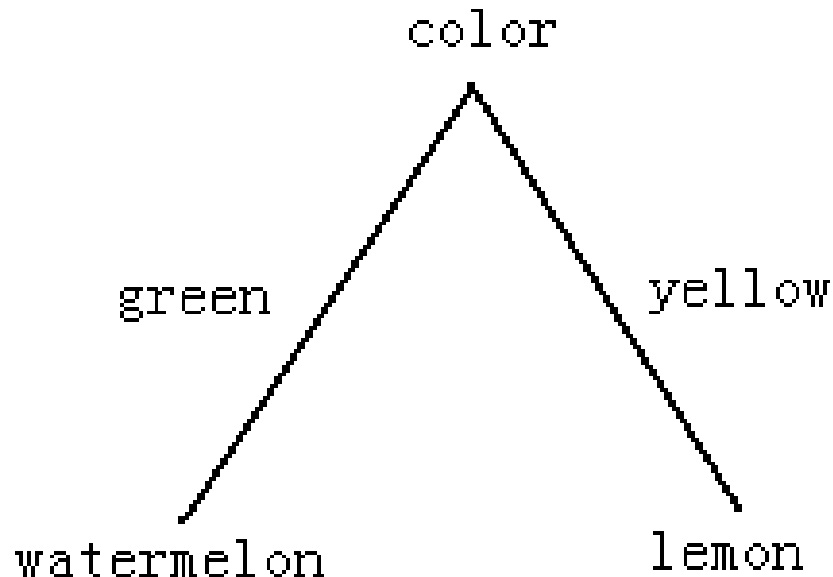
Impurity





Impurity

- Choose the query that decreases the impurity as much as possible.



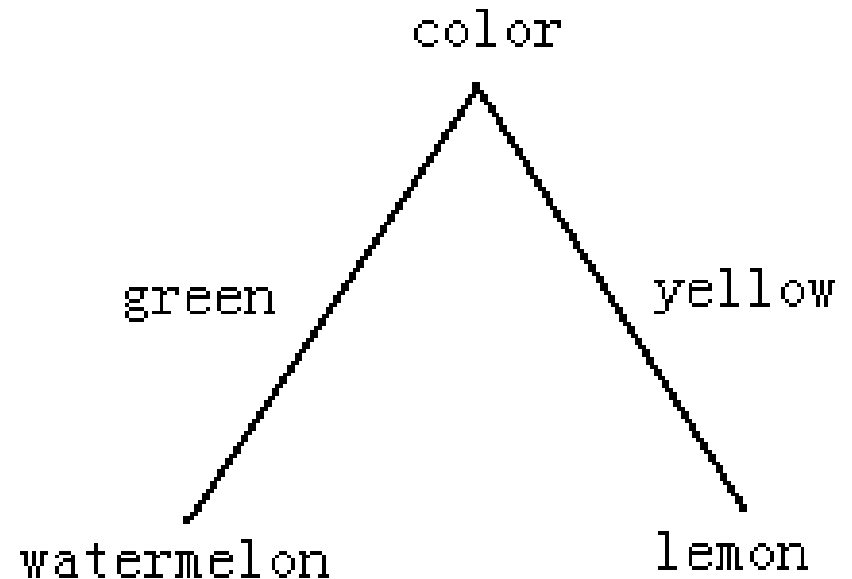


Impurity

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

$$\max \Delta i(N)$$

Is it the best method?





Impurity

- Greedy method to local optimum.
- Global optimum is the smallest tree.



Impurity

- {23, male} young
 - {25, female} young
 - {60, male} old
 - {58, female} old
-
- How to determine a threshold to distinguish the young and old with attribute “age”?



Impurity

- There are several decision s that lead to the same reduction in impurity, how to choose among them?
- For example: real -valued $x_l < x_s < x_u$

$$x_s = (x_l + x_u) / 2$$

$$x_s = (1 - P)x_l + Px_u$$



Is larger B better?

- {20030101, 23, male} young
- {20030102, 25, female} young
- {20030103, 60, male} old
- {20030104, 58, female} old



Is larger B better ?

- What is wrong with the attribute *Date*?
- Gain ratio measure:

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

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

When to stop splitting?

Traditional approach

- train tree using a subset of the data (e.g., 90%), with the remaining kept as a validation set.
- Until the error on the validation data is minimized.

When to stop splitting?

Threshold value method

- Set a small threshold value, splitting is stopped if $\Delta i(s) \leq \beta$
- Benefits: ?
- Drawback: ?

When to stop splitting?

Threshold value method

- Benefits: use all the training data. Leaf nodes can lie in different levels of the tree.
- Drawback: difficult to set a good threshold



When to stop splitting?

- Overfit: each leaf corresponds to a single training point and the full tree is merely a convenient implementation of a lookup table



Pruning

- A tree is grown fully
- All pairs of neighboring leaf nodes are considered for elimination.
- Any pair whose elimination yields a satisfactory (small) increase in impurity is eliminated
- And the common antecedent node is declared a leaf.
- Use all information in training set.
- Greater computational expense than stopped splitting.

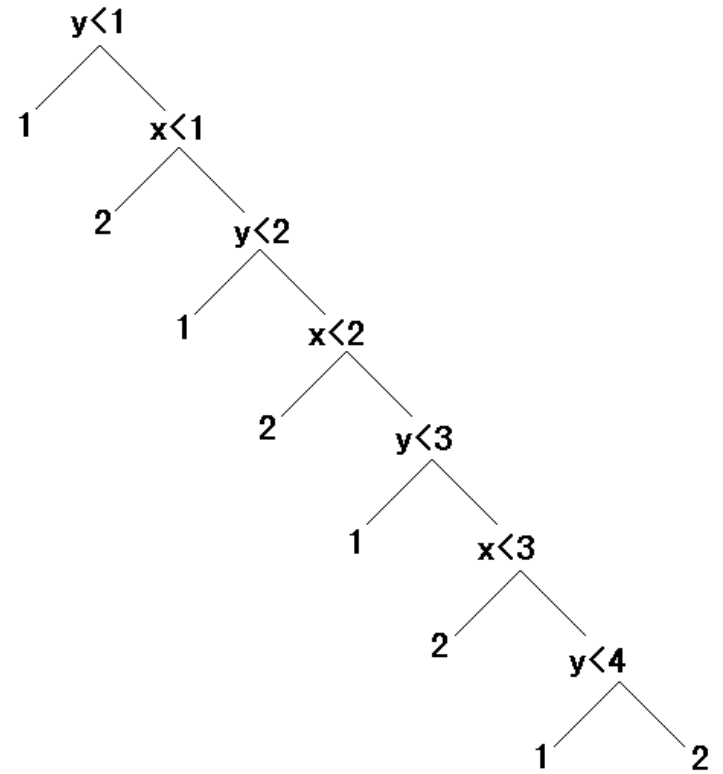
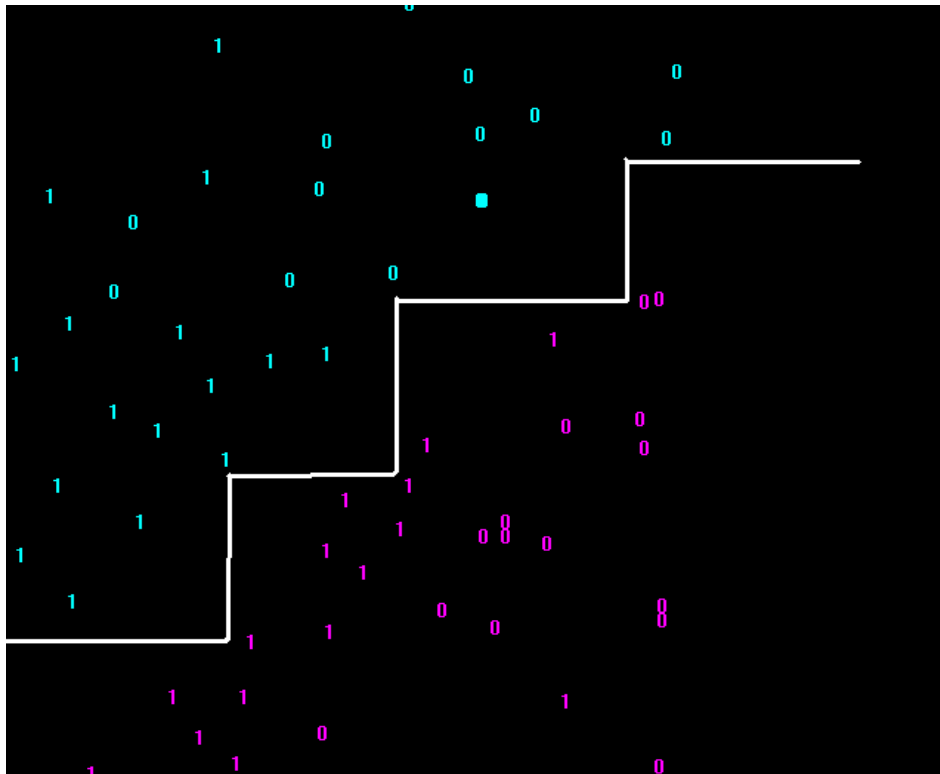
Assignment of leaf node labels



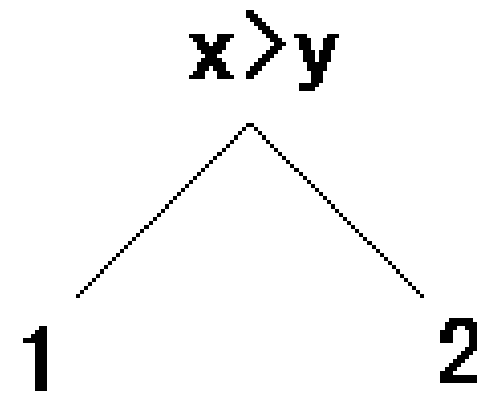
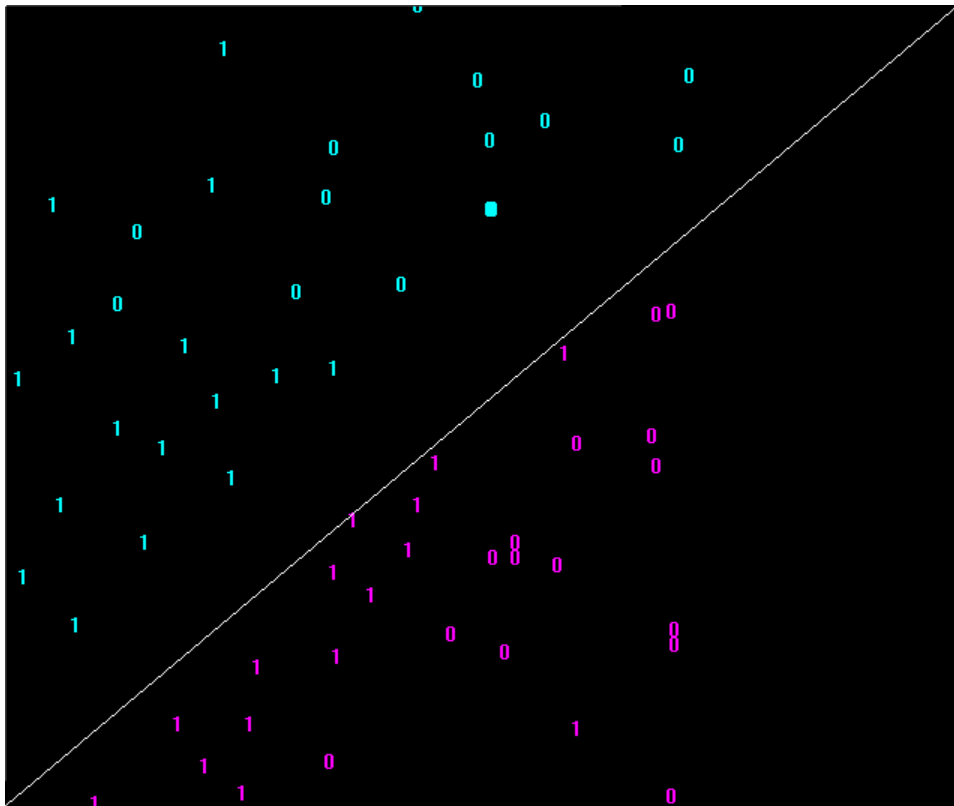
- Each leaf should be labeled by the category that has most points represented.

-

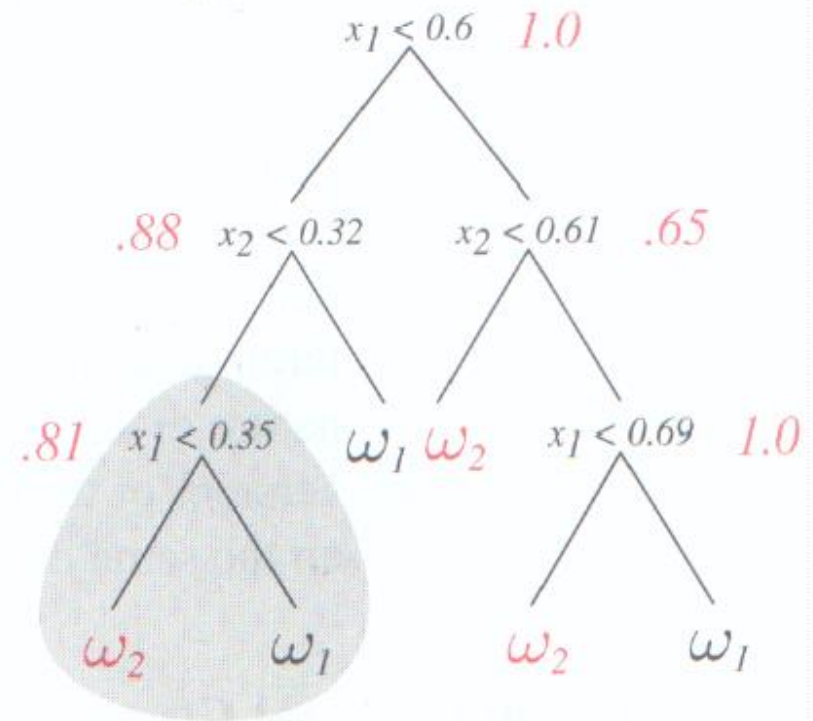
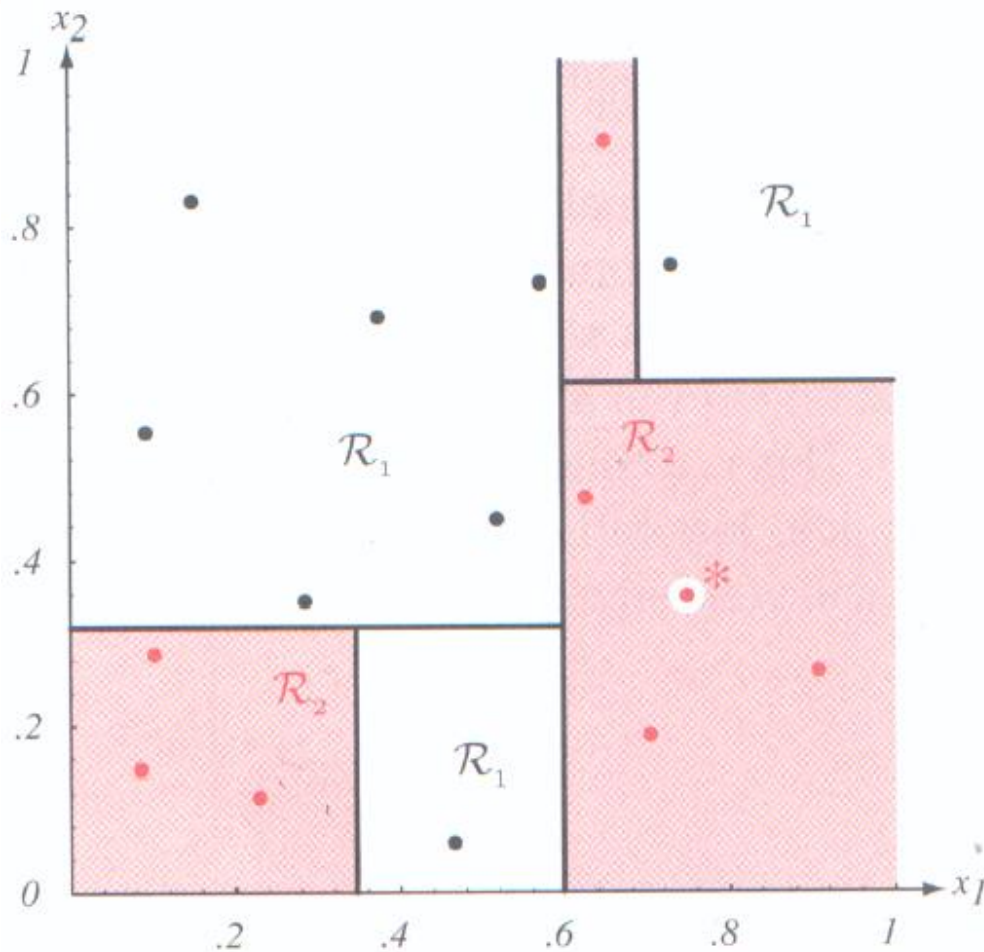
Feature choice



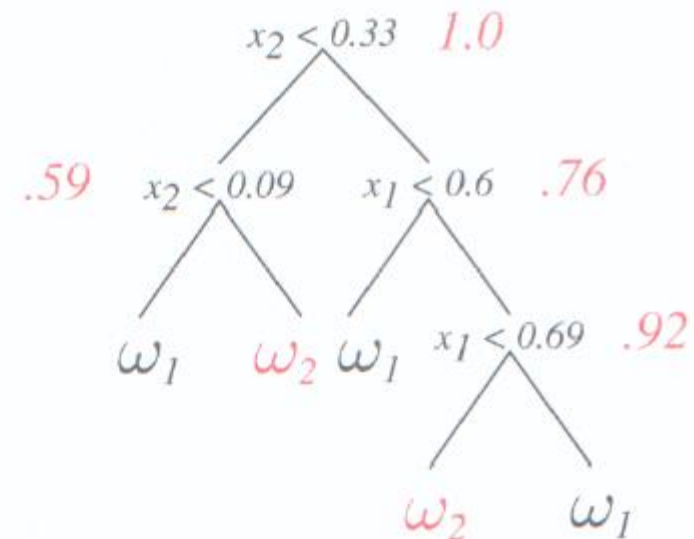
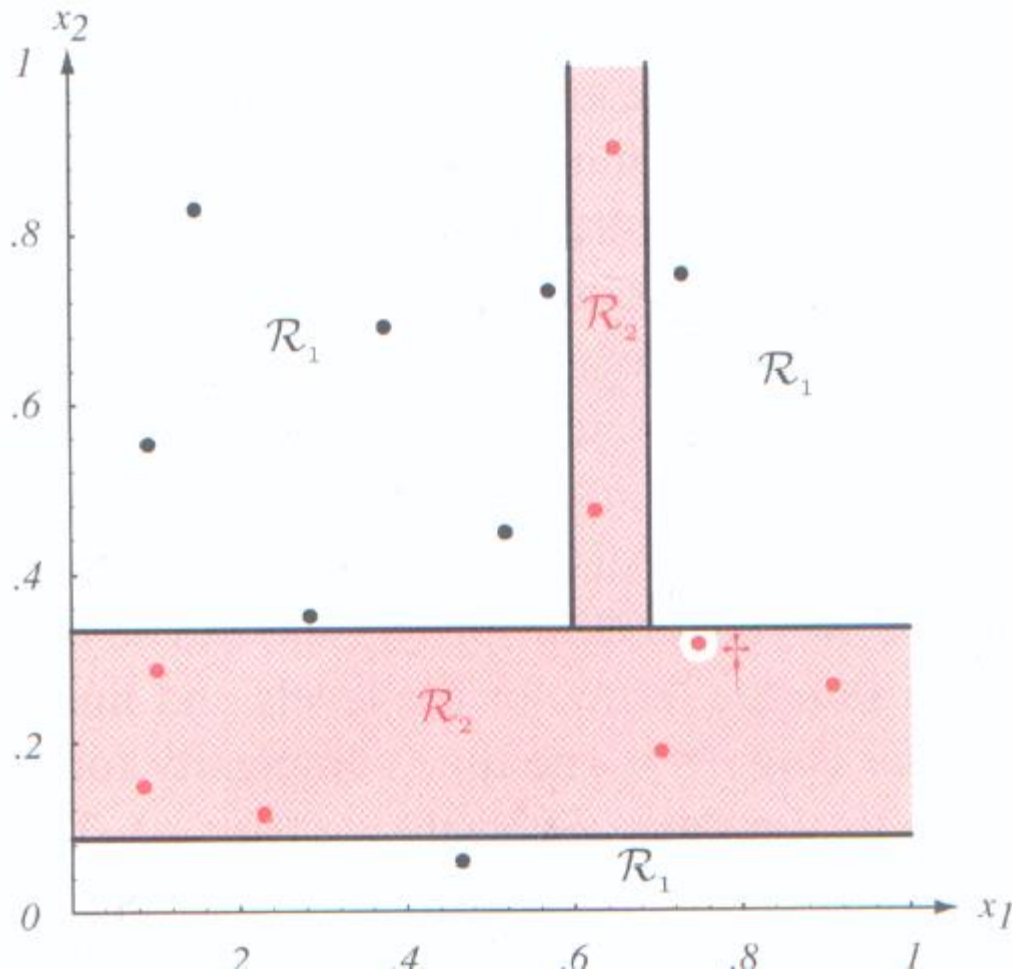
Feature choice



Different tree because of only one point



Different tree because of only one point

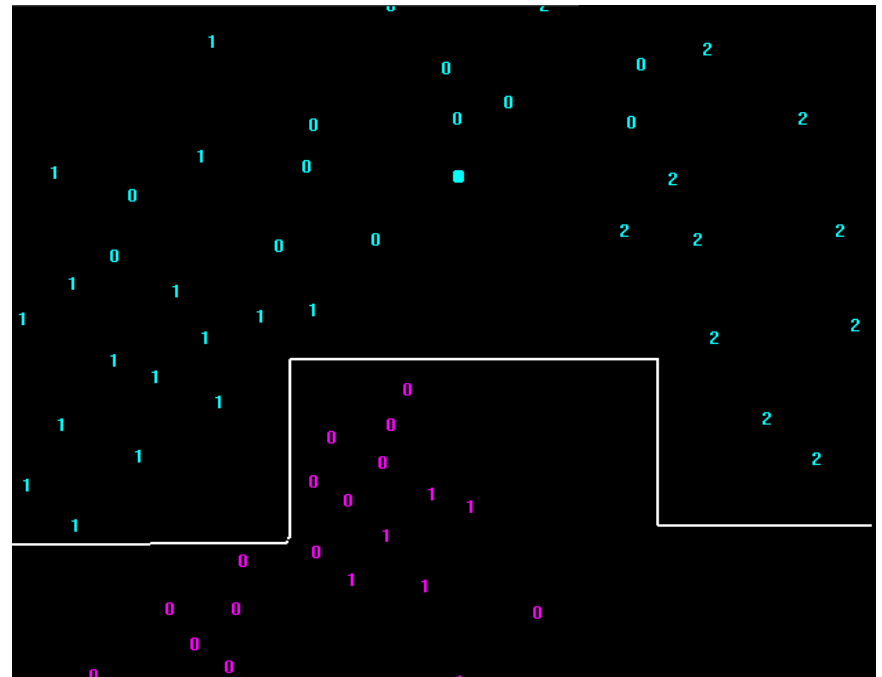
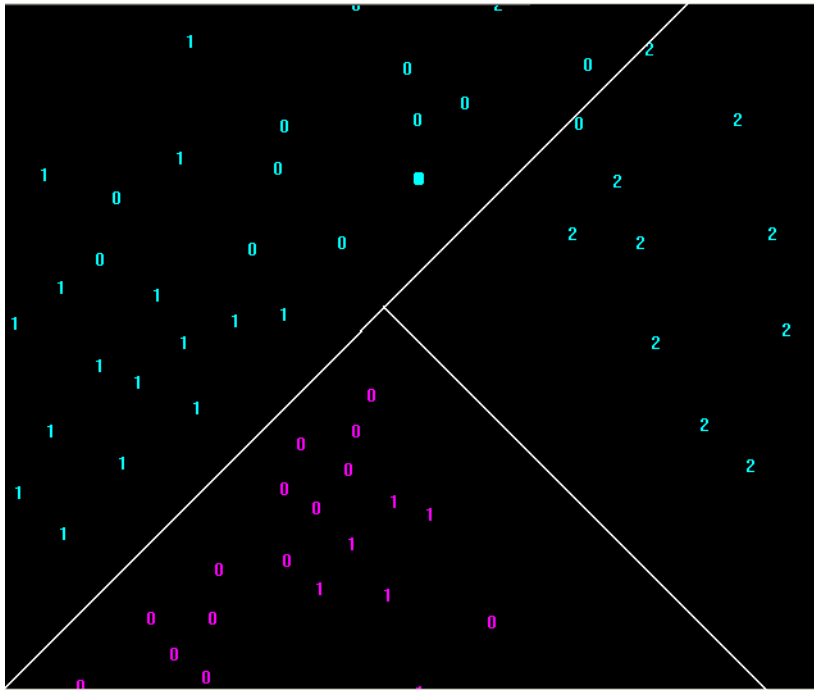




Multivariate decision tree

- If the “natural” splits of real-valued data do not fall parallel to the feature axes,?
- Allow splits that are not parallel to the feature axes, use a general linear classifier.

Multivariate decision tree





Missing attributes

- Delete any deficient patterns
- Drawback: wasteful (there are many samples)
- Calculate impurities at a node N using only the attribute information present



ID3 method

- For real-valued variables, they are first binned into intervals.
- Every split has a branching factor B_j , where a B_j is the number of discrete attribute bins of the variable j chosen for splitting.
- Such trees have their number of levels equal to the number of input variables.
- It continues until all nodes are pure or there are no more variables to split on.(no pruning)



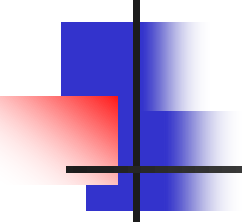
C4.5 algorithm

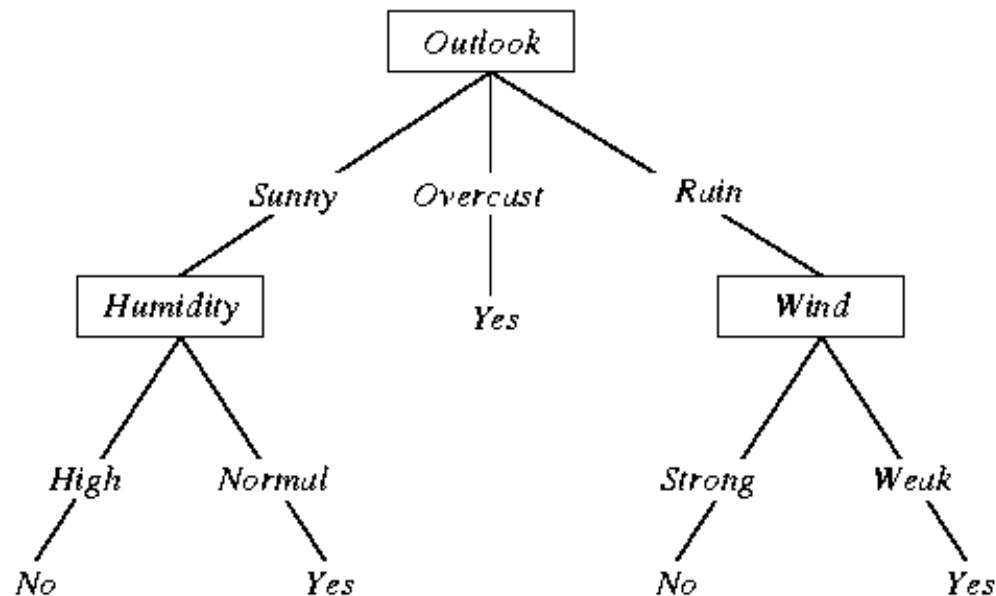
- The successor and refinement of ID3
- The most popular algorithm for decision tree.
- C4.5 follows all B possible answers to the descendent nodes and ultimately B leaf nodes for missing features.
-



C4.5 rule pruning

- Grow the tree until the training data is fit as well as possible and allowing overfitting to occur
- Convert the learned tree into an equivalent set of rules by creating one rule for each path from the root node to a leaf node.
-

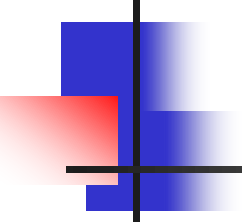
- 
- IF (Outlook = Sunny) & (Humidity = High)
 - TEHN PlayTennis = No
 -





C4.5 rule pruning

- Prune (generalize) each rule by removing any preconditions that result in improving its estimated accuracy.
- Sort the pruned rules by their estimated accuracy, and consider them in this sequence when classifying subsequent instances.
-

- 
-
- IF (Outlook = Sunny) & (Humidity = High)
 - TEHN PlayTennis = No

 - 1. IF (Humidity = High)
 - TEHN PlayTennis = No

 - 2. IF (Outlook = Sunny)
 - TEHN PlayTennis = No

 - Select the pruned one which produces the greatest improvement in estimated rule accuracy.
 - Then consider pruning the second precondition



Why convert the decision tree to rule before pruning?

- Independent to contexts. If the tree were pruned, two choices: remove the node completely, or retain it there.
- No difference between root node and leaf nodes.
- Improve readability
-



Decision Forest

- Many decision trees by C4.5
-