4.4: DECISION TREES: Non Metric Methods

Some Figures in these slides were taken from
Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000
with the permission of the authors

Febrero-Mayo 2005

M. Cabrera, J. Vidal

INDEX

1 INTRODUCTION: Decision Trees

- Previous studied Classification Methods work with real value feature vectors and compute some metric from them: Distance, Similarity, etc. Decision Tree based Methods are non metric.
- Other Alternatives: List of attributes, Discrete Features, forming a property *d*-tuple.
- Discrete Problems solved with Decision Trees, Rule-based or Syntatic Pattern Recognition.

1 INTRODUCTION: Decision Trees

- Sequence of questions to classify a pattern.
- Root Node and successive branches linked to other nodes.
- Links must be mutually distinct and exhaustive.
- Questions finish at leaf nodes.

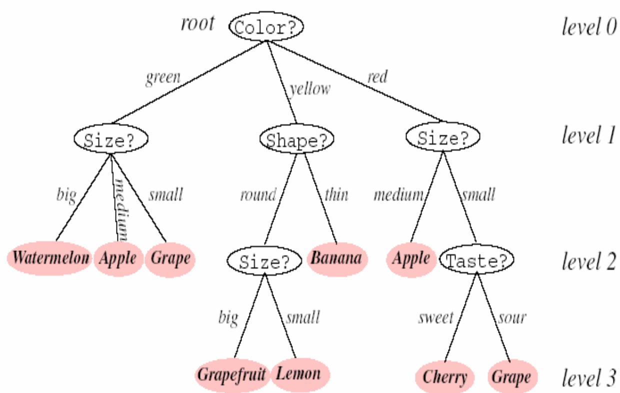


FIGURE 8.1. Classification in a basic decision tree proceeds from top to bottom. The questions asked at each node concern a particular property of the pattern, and the downward links correspond to the possible values. Successive nodes are visited until a terminal or leaf node is reached, where the category label is read. Note that the same question, Size?, appears in different places in the tree and that different questions can have different numbers of branches. Moreover, different leaf nodes, shown in pink, can be labeled by the same category (e.g., **Apple**). From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

1 INTRODUCTION: Decision Trees

DECISION BOUNDARIES:

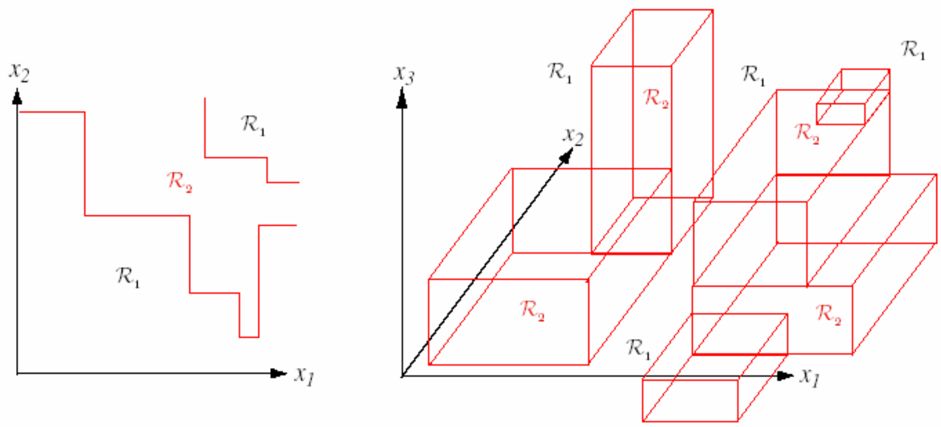


FIGURE 8.3. Monothetic decision trees create decision boundaries with portions perpendicular to the feature axes. The decision regions are marked $\mathcal{R}_1$ and $\mathcal{R}_2$ in these two-dimensional and three-dimensional two-category examples. With a sufficiently large tree, any decision boundary can be approximated arbitrarily well in this way. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

3 CART: Classification And Regression Trees BINARY DECISIONS

- A Training Labeled Dataset is used to create a Classification Tree
- A decision tree progressively splits the set of training samples into smaller and smaller subsets.
- When all the samples in a subset have the same category the branch of the tree is terminated.
- A branch can be alternatively terminated with a mixture subset and declared leaf using CARTs
- Objective: To obtain a small binary tree

- Important questions working with CARTS:
  – Which feature must be tested at each node???
  – How pruning the tree?

3 CART: Classification And Regression Trees

We seek a property to test at each node that makes the immediate descendent node as pure as possible. Impurity in minimized

- ENTROPY IMPURITY: 'infcrit'

$$i_E(N) = -\sum_j P(\omega_j)\log_2 P(\omega_j); \qquad P(\omega_j) = \frac{n_j}{n_{TOTAL}}$$

- GINI IMPURITY: 'maxcrit'

$$i_G(N) = \sum_{i \neq j} P(\omega_i) P(\omega_j) = 1 - \sum_j P^2(\omega_j)$$

- MISSCLASSIFICATION IMPURITY: 'fishcrit' NO ES EL FISHCRIT DEL PRTOOLS (QUE ESTA SOLO PARA DOS CLASES)

$$i_M(N) = 1 - \max_j \left(P(\omega_j)\right)$$

3 CART: Classification And Regression Trees

- ENTROPY, GINI IMPURITY, MISSCLASSIFICATION for c=2 classes.

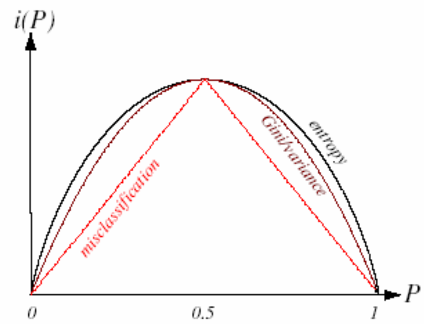


FIGURE 8.4. For the two-category case, the impurity functions peak at equal class frequencies and the variance and the Gini impurity functions are identical. The entropy, variance, Gini, and misclassification impurities (given by Eqs. 1–4, respectively) have been adjusted in scale and offset to facilitate comparison here; such scale and offset do not directly affect learning or classification. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

BINARY TREES (Pre Pruning)

- Given a node N: What feature "T >< s" should be chosen for the test??.
- Decrease the impurity at N node (N-local optimization):

$$\Delta i(N) = i(N) - \tfrac{n_L}{n_L + n_R} i(N_L) - \tfrac{n_R}{n_L + n_R} i(N_R) < 1 \text{ bit}$$

- Sometimes several decisions implies same impurity variation. With real values:

$$x_s = \frac{n_L}{n_L + n_R} x_L - \frac{n_R}{n_L + n_R} x_R$$

- Practical Problem: If two different patterns have the same attributes the impurity at leafs cannot be reduced to zero.
- Multiclass binary tree: The set of c categories is divided into two subsets and the problem is solved as a binary one. Computationally it is expensive because all possible subset division must be tested.
- The particular choice of an impurity function rarely affects the final structure for the tree.

3 CART: Classification And Regression Trees

WHEN TO STOP SPLITTING??? Pre Pruning Methods

If the training set is very big, the obtained tree can be over fitted.

In the extreme each leaf corresponds to a single training point.

1.- UNBALANCED TREES: The impurity lost of the best split must be higher than a threshold.

2.- MDL: Minimum Description Length: A Criterion function is minimized (size is the number of nodes). α is aTunning positive parameter that governs the tradeoff between tree size and it goodness of fit to the data. Large values α of result in smaller tree size.

$$\alpha.\text{size} + \sum_{N \ \text{leaf}} i(N)$$

3.-VALIDATION TECHNIQUES: A α% remaining validation set is used to test the tree trained with the (1-α)% set. The training process finishes when the test error is minimized.

With CROSS-VALIDATION TECHNIQUES: Different independent validation sets are used.

3 Pruning CART

PRUNING (Alternative to stopped splitting: postpruning):

- In a first step the tree is grown fully or until some minimum size
- In a second step some sub-trees are substituted for leafs following different pruning criteria.
- The final tree results unbalanced

- MATLAB(PrTools alternatives for pruning):

  W = treec(A,crit,prune,T)

  prune = -1 pessimistic pruning as defined by Quinlan (post pruning PEP).

  prune = -2 testset pruning using the dataset T (post pruning REP)

  prune = 0 no pruning

  prune > 0 early pruning (pre pruning), e.g. prune = 3

  prune = 10 causes heavy pruning.

3 Post-Pruning CART

PRUNING STRATEGIES (Pessimistic or Quinlan Criteria PEP):

- It is a TOP-DOWN approach:

$$T_0 \supset T_1 \supset .... \supset T_k \quad \Rightarrow \quad \begin{cases} C_\alpha(T_i) = \sum_{n=1}^{L(T_i)} \sum_{x \in w_n} \left(x - \hat{x}_n\right)^2 + \alpha L(T_i) \\ C_\alpha(T_i) = \sum_{n=1}^{L(T_i)} \varepsilon_L(T_i) + \alpha L(T_i) \end{cases}$$

$$\alpha = 0.5$$

- The node T_i is replaced by a leaf if:

$$C_\alpha(T_i) < \sum_i C_\alpha(T_{i+1})$$

3 Post-Pruning CART

PRUNING STRATEGIES (Reduced Error Pruning REP):

- It is a DOWN-TOP approach:
  - It uses an additional training set, called the "pruning set" unseen during the growing stage.
  - A simple error check is calculated for all nonleaf nodes.

- The node T_i is replaced by a leaf if the error is smaller than the error of the whole tree on the pruning set.
  - The leaf is labeled to the majority class.
  - Danger: Tendency towards overpruning.

4 Other Methods

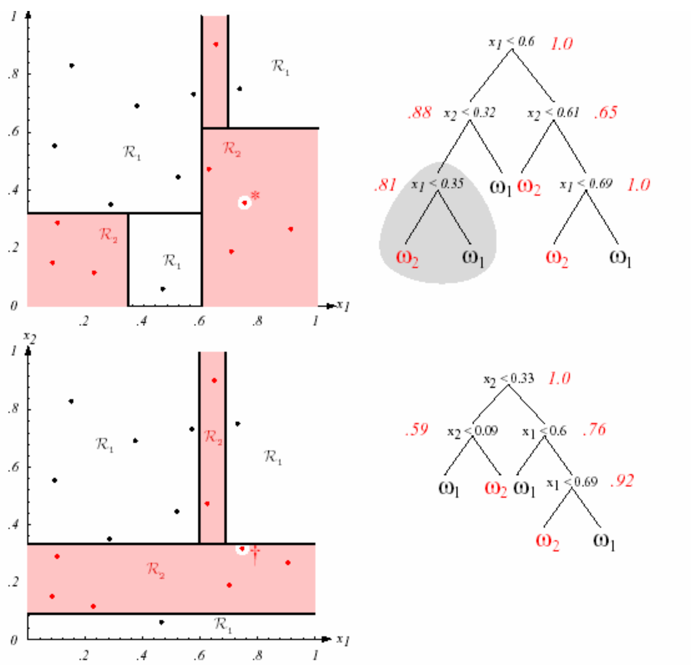
ID3: Interactive dichotomizer 3 (Non Binary):

- It is designed for nominal data.
- Discrete Feature Vectors (or discretized)
- Categories are treated as unordered
- Each node has as many children nodes as the number of categories of the (nominal) feature at that node.
- The maximum number of levels is equal to the number of features.
- The algorithm continues until all nodes are pure or there are no more variables to split on.

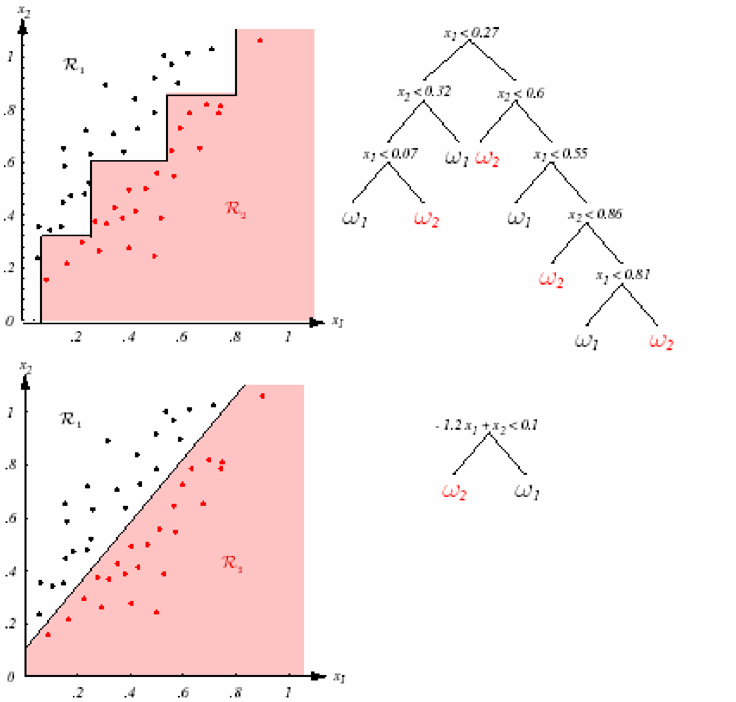
C4.5 is an extension of ID3 that accounts for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation, and so on. Last Version C5.0 [Quinlan, 1993]

- It uses continuous valued variables as in CARTS and the nominal variables as in ID3.

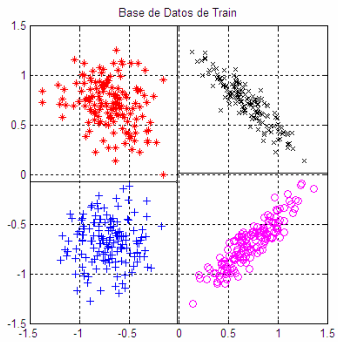
EXAMPLE: Sensibility of Decision Trees with data



Example: Feature Choice

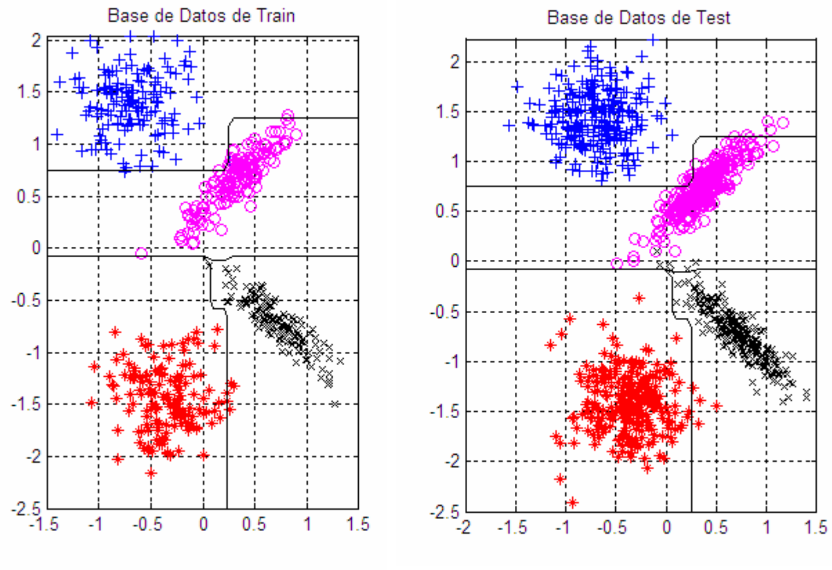


Example: 4 Gaussian Classes



Step	feature number to be used in node n /CLASS (If leaf)	threshold t to be used	node to be processed if value <= t	node to be processed if value>= t	aposteriori probabilities for CLASS 1	aposteriori probabilities for CLASS 2	aposteriori probabilities for CLASS 3	aposteriori probabilities for CLASS 4
1	1	-0.0069538	2	5	0.25	0.25	0.25	0.25
2	2	-0.061336	3	4	0.49671	0.49671	0.0032895	0.0032895
3	1	0	0	0	0.98052	0.0064935	0.0064935	0.0064935
4	2	0	0	0	0.0064935	0.98052	0.0064935	0.0064935
5	2	0.023338	6	7	0.0032895	0.0032895	0.49671	0.49671
6	3	0	0	0	0.0064935	0.0064935	0.98052	0.0064935
7	4	0	0	0	0.0064935	0.0064935	0.0064935	0.98052

Example: 4 Gaussian Classes



Example: 4 Gaussian Classes

Step	feature number to be used in node n /CLASS (If leaf)	threshold t to be used	node to be processed if value <= t	node to be processed if value>= t	aposteriori probabilities for CLASS 1	aposteriori probabilities for CLASS 2	aposteriori probabilities for CLASS 3	aposteriori probabilities for CLASS 4
1	2	0.07185	2	13	0.25	0.25	0.25	0.25
2	1	0.094422	3	8	0.003268	0.49346	0.0098039	0.49346
3	2	-0.4957	4	5	0.0064935	0.94805	0.019481	0.025974
4	2	0	0	0	0.0067114	0.97987	0.0067114	0.0067114
5	1	-0.17307	6	7	0.11111	0.11111	0.33333	0.44444
6	3	0	0	0	0.16667	0.16667	0.5	0.16667
7	4	0	0	0	0.14286	0.14286	0.14286	0.57143
8	2	-11.238	9	12	0.0064103	0.038462	0.0064103	0.94872
9	1	0.69391	10	11	0.045455	0.27273	0.045455	0.63636
10	2	0	0	0	0.11111	0.66667	0.11111	0.11111
11	4	0	0	0	0.058824	0.058824	0.058824	0.82353
12	4	0	0	0	0.0072464	0.0072464	0.0072464	0.97826
13	1	-0.17895	14	17	0.5	0.0033113	0.49338	0.0033113
14	2	0.75839	15	16	0.94904	0.0063694	0.038217	0.0063694
15	3	0	0	0	0.11111	0.11111	0.66667	0.11111
16	1	0	0	0	0.98026	0.0065789	0.0065789	0.0065789
17	1	-0.0026306	18	21	0.020134	0.0067114	0.96644	0.0067114
18	2	0.76888	19	20	0.2	0.066667	0.66667	0.066667
19	3	0	0	0	0.076923	0.076923	0.76923	0.076923
20	1	0	0	0	0.5	0.16667	0.16667	0.16667
21	3	0	0	0	0.0072464	0.0072464	0.97826	0.0072464

5 Conclusions

- Entropy impurity measure is the most acceptable in most cases.
- Pruning (Post pruning) is preferred over stopped splitting (Pre pruning) but computationally worst.
- To bin real values as in ID3 is only useful if computational advantages are high.
- High training set size can produce over fitted trees.
- It is recommended to exploit designer information on feature pre-processing steps.
- They are particularly useful with non-metric data.