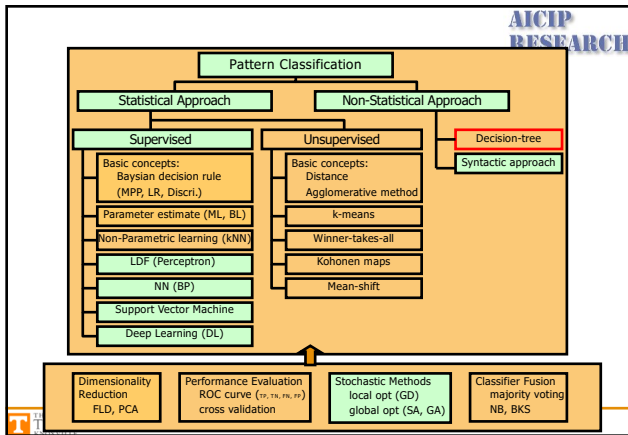



ECE471-571 – Pattern Recognition

Lecture 13 – Decision Tree

Hairong Qi, Gonzalez Family Professor
 Electrical Engineering and Computer Science
 University of Tennessee, Knoxville
<http://www.eecs.utk.edu/faculty/qj>
 Email: hqi@utk.edu



Review - Bayes Decision Rule



$$P(\omega_j | x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$$

Maximum Posterior Probability: For a given x , if $P(\omega_1|x) > P(\omega_2|x)$, then x belongs to class 1, otherwise 2

Likelihood Ratio: If $\frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{P(\omega_2)}{P(\omega_1)}$, then x belongs to class 1, otherwise, 2.


Discriminant Function: The classifier will assign a feature vector x to class ω_j if $g_j(x) > g_k(x)$

Case 1: Minimum Euclidean Distance (Linear Machine), $\Sigma_i = \sigma^2 I$
 Case 2: Minimum Mahalanobis Distance (Linear Machine), $\Sigma_i = \Sigma$
 Case 3: Quadratic classifier, $\Sigma_i =$ arbitrary

Non-parametric kNN: For a given x , if $k_1/k > k_2/k$, then x belongs to class 1, otherwise 2

Estimate Gaussian (Maximum Likelihood Estimation, MLE),
Two-modal Gaussian

Dimensionality reduction
Performance evaluation
ROC curve

 3

Nominal Data

- ◆ Descriptions that are discrete and without any natural notion of similarity or even ordering

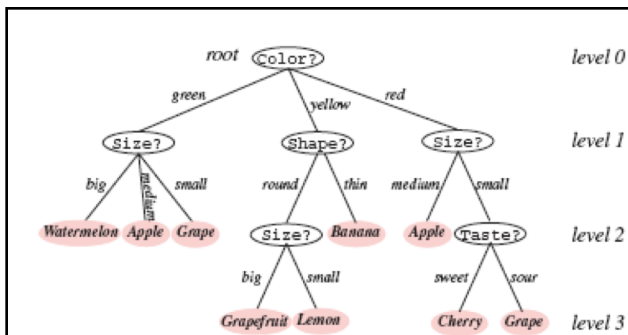


FIGURE 8.1. Classification in a basic decision tree proceeds from top to bottom. The questions at each node concern a particular property of the pattern, and the downward links correspond to the responses. Successive nodes are visited until a terminal or leaf node is reached, where the category label is placed. Note that the same question, *Size?*, appears in different places in the tree and that different questions have different numbers of branches. Moreover, different leaf nodes, shown in pink, can be labeled differently.

CART

- ◆ Classification and regression trees
- ◆ A generic tree growing methodology
- ◆ Issues studied
 - How many splits from a node?
 - Which property to test at each node?
 - When to declare a leaf?
 - How to prune a large, redundant tree?
 - If the leaf is impure, how to classify?
 - How to handle missing data?

Number of Splits

- ◆ Binary tree
- ◆ Expressive power and comparative simplicity in training

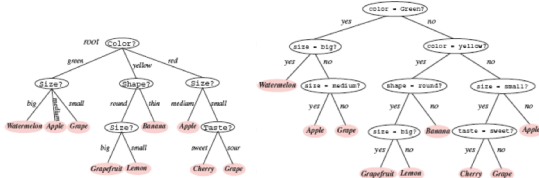


FIGURE 6.1 Classification in a basic decision tree proceeds from top to bottom. The q of each node concern a particular property of the pattern, and the downward links correspond to the values. Successive nodes are visited until a terminal or leaf node is reached, where the class label is assigned. Note that the same question, q_i , appears in different places in the tree and that different nodes have different numbers of branches. Moreover, different leaf nodes, shown in pink, can belong to the same category (e.g., Apple). From Richard O. Duda, Peter E. Hart, and David G. Stork, eds. Copyright © 2001 by John Wiley & Sons, Inc.

FIGURE 6.2 A tree with arbitrary branching factors at different nodes can always be represented by a functionally equivalent binary tree—that is, one having branching factor $B = 2$ throughout, as shown here. By convention the “yes” branch is on the left, the “no” branch on the right. This binary tree contains the same information and implements the same classification as that in Fig. 6.1. From Richard O. Duda, Peter E. Hart, and David G. Stork, eds. Copyright © 2001 by John Wiley & Sons, Inc.

Node Impurity – Occam’s Razor

- The fundamental principle underlying tree creation is that of simplicity: we prefer simple, compact tree with few nodes
- <http://math.ucr.edu/home/baez/physics/occam.html>
- Occam’s (or Ockham’s) razor is a principle attributed to the 14th century logician and Franciscan friar, William of Occam. Ockham was the village in the English county of Surrey where he was born.
- The principle states that “Entities should not be multiplied unnecessarily.”
- “when you have two competing theories which make exactly the same predictions, the one that is simpler is the better.”
- Stephen Hawking explains in A Brief History of Time: “We could still imagine that there is a set of laws that determines events completely for some supernatural being, who could observe the present state of the universe without disturbing it. However, such models of the universe are not of much interest to us mortals. It seems better to employ the principle known as Occam’s razor and cut out all the features of the theory which cannot be observed.”
- Everything should be made as simple as possible, but not simpler

Property Query and Impurity Measurement

- ◆ We seek a property query T at each node N that makes the data reach the immediate descendent nodes as pure as possible
- ◆ We want $i(N)$ to be 0 if all the patterns reach the node bear the same category label
- ◆ Entropy impurity (information impurity)

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

$P(\omega_j)$ is the fraction of patterns at node N that are in category ω_j

Other Impurity Measurements

AICIP
RESEARCH

- ◆ Variance impurity (2-category case)

$$i(N) = P(\omega_1)P(\omega_2)$$

- ◆ Gini impurity

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

- ◆ Misclassification impurity

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

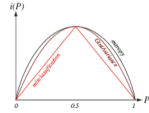


FIGURE 8.4. For the two-category case, the impurity functions peak at equal class frequencies and the variance and the Gini impurity functions are identical. The entropy, variance, Gini, and misclassification impurities given by Eqs. 1–4, respectively have been adjusted in scale and offset to facilitate comparison here; each 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.

Choose the Property Test?

AICIP
RESEARCH

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

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

- N_L, N_R : left and right descendent nodes
- $i(N_L), i(N_R)$: impurities
- P_L : fraction of patterns at node N that will go to N_L
- ◆ Solve for extrema (local extrema)

Example

AICIP
RESEARCH

- ◆ Node N :

- 90 patterns in ω_1
- 10 patterns in ω_2

- ◆ Split candidate:


- 70 ω_1 patterns & 0 ω_2 patterns to the right
- 20 ω_1 patterns & 10 ω_2 patterns to the left

**AICIP
RESEARCH**

When to Stop Splitting?

- ◆ Two extreme scenarios
 - Overfitting (each leaf is one sample)
 - High error rate
- ◆ Approaches
 - Validation and cross-validation
 - 90% of the data set as training data
 - 10% of the data set as validation data
 - Use threshold
 - Unbalanced tree
 - Hard to choose threshold
 - Minimum description length (MDL)

$$MDL = \alpha * size + \sum_{leaf\ nodes} i(N)$$
 - $i(N)$ measures the uncertainty of the training data
 - Size of the tree measures the complexity of the classifier itself


 THE UNIVERSITY OF TENNESSEE
13

**AICIP
RESEARCH**

When to Stop Splitting? (cont')

- ◆ Use stopping criterion based on the statistical significance of the reduction of impurity
 - Use chi-square statistic
 - Whether the candidate split differs significantly from a random split


$$\chi^2 = \sum_{i=1}^2 \frac{(n_{it} - n_{ie})^2}{n_{ie}}$$

 THE UNIVERSITY OF TENNESSEE
14

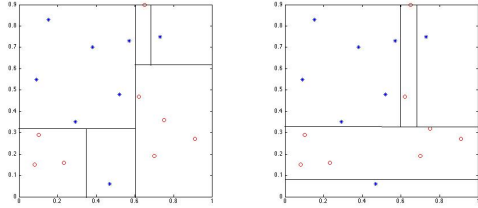
**AICIP
RESEARCH**

Pruning

- ◆ Another way to stop splitting
- ◆ Horizon effect
 - Lack of sufficient look ahead
- ◆ Let the tree fully grow, i.e. beyond any putative horizon, then all pairs of neighboring leaf nodes are considered for elimination

 THE UNIVERSITY OF TENNESSEE
15

Instability



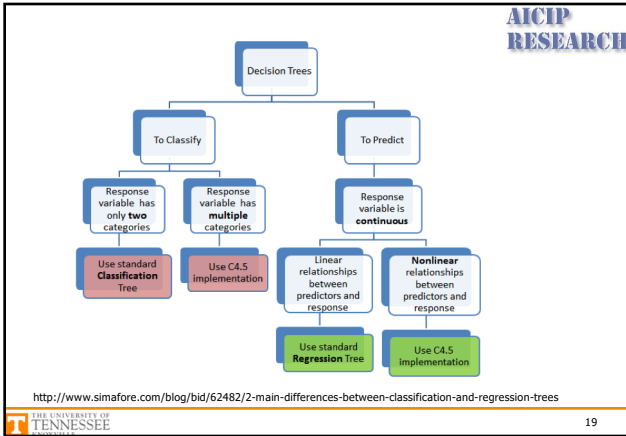
Other Methods

- Quinlan's ID3
 - C4.5 (successor and refinement of ID3)
- <http://www.rulequest.com/Personal/>

MATLAB Routine

- classregtree
- Classification tree vs. Regression tree
 - If the target variable is categorical or numeric

<http://www.mathworks.com/help/stats/classregtree.html>



AICIP RESEARCH

Random Forest

- Potential issue with decision trees
- Prof. Leo Breiman
- Ensemble learning methods
 - Bagging (**B**ootstrap **a**ggregating): Proposed by Breiman in 1994 to improve the classification by combining classifications of randomly generated training sets
 - Random forest: bagging + random selection of features at each node to determine a split

THE UNIVERSITY OF TENNESSEE 20

AICIP RESEARCH

MATLAB Implementation

- `B = TreeBagger(nTrees, train_x, train_y);`
- `pred = B.predict(test_x);`

THE UNIVERSITY OF TENNESSEE 21

Reference

- [CART] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
- [Bagging] L. Breiman, "Bagging predictors," *Machine Learning*, 24(2):123-140, August 1996. (citation: 16,393)
- [RF] L. Breiman, "Random forests," *Machine Learning*, 45(1):5-32, October 2001.
