SNS COLLEGE OF TECHNOLOGY

Coimbatore – 35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

An Introduction to Decision Tree

Learning Systems

- Learning systems consider
  - Solved cases - cases assigned to a class
- Information from the solved cases - general decision rules
- Rules - implemented in a model
- Model - applied to new cases
- Different types of models - present their results in various forms

- Linear discriminant model - mathematical equation ($p = ax_1 + bx_2 + cx_3 + dx_4 + ex_5$).

- Presentation comprehensibility

Data Classification and Prediction

- Data classification
  - classification
  - prediction
- Methods of classification
  - decision tree induction
  - Bayesian classification
  - backpropagation
  - association rule mining

Data Classification and Prediction

- Method creates model from a set of training data
  - individual data records (samples, objects, tuples)
  - records can each be described by its attributes
  - attributes arranged in a set of classes
  - supervised learning - each record is assigned a class label

Data Classification and Prediction

- Model form representations
  - mathematical formulae
  - classification rules
  - decision trees
- Model utility for data classification
  - degree of accuracy
  - predict unknown outcomes for a new (no-test) data set
  - classification - outcomes always discrete or nominal values
  - regression may contain continuous or ordered values

Description of Decision Rules or Trees

- Intuitive appeal for users
- Presentation Forms
  - “if, then” statements (decision rules)
  - graphically - decision trees

What They Look Like

- Works like a flow chart
- Looks like an upside down tree
- Nodes
  - appear as rectangles or circles
  - represent test or decision
- Lines or branches - represent outcome of a test
- Circles - terminal (leaf) nodes
- Top or starting node- root node
- Internal nodes - rectangles

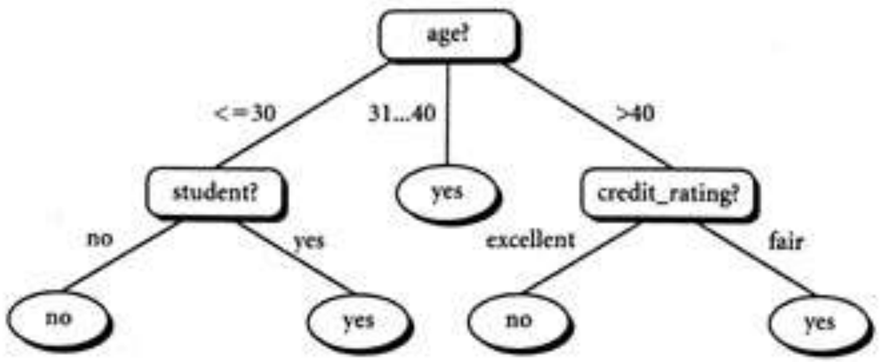

Figure 7.2 A decision tree for the concept *buys_computer*, indicating whether or not a customer at *AllElectronics* is likely to purchase a computer. Each internal (nonleaf) node represents a test on an attribute. Each leaf node represents a class (either *buys_computer = yes* or

An Example

- Bank - loan application
- Classify application
  - approved class
  - denied class
- Criteria - Target Class approved if 3 binary attributes have certain value:
  - (a) borrower has good credit history (credit rating in excess of some threshold)
  - (b) loan amount less than some percentage of collateral value (e.g., 80% home value)
  - (c) borrower has income to make payments on loan
- Possible scenarios = 3^2 = 8
  - If the parameters for splitting the nodes can be adjusted, the number of scenarios grows exponentially.

How They Work

- Decision rules - partition sample of data
- Terminal node (leaf) indicates the class assignment
- Tree partitions samples into mutually exclusive groups
- One group for each terminal node
- All paths
  - start at the root node
  - end at a leaf
- Each path represents a decision rule
  - joining (AND) of all the tests along that path
  - separate paths that result in the same class are disjunctions (ORs)
- All paths - mutually exclusive
  - for any one case - only one path will be followed
  - false decisions on the left branch
  - true decisions on the right branch

Disjunctive Normal Form

- Non-terminal node - model identifies an attribute to be tested
  - test splits attribute into mutually exclusive disjoint sets
  - splitting continues until a node - one class (terminal node or leaf)
- Structure - *disjunctive normal form*
  - limits form of a rule to conjunctions (adding) of terms
  - allows disjunction (or-ing) over a set of rules

Geometry

- Disjunctive normal form
- Fits shapes of decision boundaries between classes
- Classes formed by lines parallel to axes
- Result - rectangular shaped class regions

INDUCTION OF OBLIQUE DECISION TREES

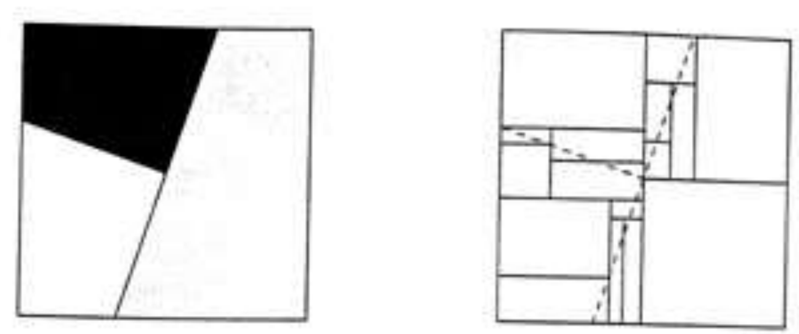

Figure 2: The left side shows a simple 2-D domain in which two oblique hyperplanes define the classes. The right side shows an approximation of the sort that an axis-parallel decision tree would have to create to model this domain.

Binary Trees

- Characteristics
  - two branches leave each non-terminal node
  - those two branches cover outcomes of the test
  - exactly one branch enters each non-root node
  - there are n terminal nodes
  - there are n-1 non-terminal nodes

Nonbinary Trees

- Characteristics
  - two or more branches leave each non-terminal node
  - those branches cover outcomes of the test
  - exactly one branch enters each non-root node
  - there are n terminal nodes
  - there are n-1 non-terminal nodes

Goal

- Dual goal - Develop tree that
  - is small
  - classifies and predicts class with accuracy
- Small size
  - a smaller tree more easily understood
  - smaller tree less susceptible to overfitting
  - large tree less information regarding classifying and predicting cases

Rule Induction

- Process of building the decision tree or ascertaining the decision rules
  - tree induction
  - rule induction
  - induction
- Decision tree algorithms
  - induce decision trees recursively
  - from the root (top) down - *greedy* approach
  - established basic algorithms include ID3 and C4.5

Discrete vs. Continuous Attributes

- Continuous variables attributes - problems for decision trees
  - increase computational complexity of the task
  - promote prediction inaccuracy
  - lead to overfitting of data
- Convert continuous variables into discrete intervals
  - “greater than or equal to” and “less than”
  - optimal solution for conversion
  - difficult to determine discrete intervals ideal
    - size
    - number

Making the Split

- Models induce a tree by recursively selecting and subdividing attributes
  - random selection - noisy variables
  - inefficient production of inaccurate trees
- Efficient models
  - examine each variable
  - determine which will improve accuracy of entire tree
  - problem - this approach decides best split without considering subsequent splits

Evaluating the Splits

Measures of impurity or its inverse, goodness reduce impurity or degree of randomness at each node popular measures include:

Entropy Function

- ☝ p_j log p_j
 j

Gini Index

1 - ☝ p^2_j
 j

Twoing Rule

k
(🖂T$_L$ 🖂/*n*) * (🖂T$_R$ 🖂/*n*) * (☝ 🖂L$_i$ 🖂T$_L$🖂 - R$_i$📫
🖂T$_R$🖂🖂)^2
i=1

Evaluating the Splits

Max Minority

$$\text{MinorityL} = \sum_{i=1, i \neq \max L_i}^{k} L_i$$

$$\text{MinorityR} = \sum_{i=1, i \neq \max R_i}^{k} R_i$$

Sum of Variances

$$\text{Max Minority} = \max(\text{MinorityL}, \text{MinorityR})$$

Sum Of Variances. The definition of this measure is:

$$\text{VarianceL} = \sum_{i=1}^{|T_L|} (Cat(T_{L_i}) - \sum_{j=1}^{|T_L|} Cat(T_{L_j})/|T_L|)^2$$

$$\text{VarianceR} = \sum_{i=1}^{|T_R|} (Cat(T_{R_i}) - \sum_{j=1}^{|T_R|} Cat(T_{R_j})/|T_R|)^2$$

$$\text{Sum of Variances} = \text{VarianceL} + \text{VarianceR}$$

Overfitting

- Error rate in predicting the correct class for new cases
  - overfitting of test data
  - very low apparent error rate
  - high actual error rate

Optimal Size

- Certain minimal size smaller tree
  - higher apparent error rate
  - lower actual error rate
- Goal
  - identify threshold
  - minimize actual error rate
  - achieve greatest predictive accuracy

Ending Tree Growth

- Grow the tree until
  - additional splitting produces no significant information gain
  - statistical test - a chi-squared test
  - problem - trees that are too small
  - only compares one split with the next descending split

Pruning

- Grow large tree
  - reduce its size by eliminating or pruning weak branches step by step
  - continue until minimum true error rate
- Pruning Methods
  - *reduced-error* pruning
  - divides samples into test set and training set
  - training set is used to produce the fully expanded tree
  - tree is then tested using the test set
  - weak branches are pruned
  - stop when no more improvement

Pruning

- Resampling
  - 5 - fold cross-validation
  - 80% cases used for training; remainder for testing
- *Weakest-link* or *cost-complexity* pruning
  - trim weakest link (produces the smallest increase in the apparent error rate)
  - method can be combined with resampling

Variations and Enhancements to Basic Decision Trees

- Multivariate or Oblique Trees
  - CART-LC - CART with Linear Combinations
  - LMDT - Linear Machine Decision Trees
  - SADT - Simulated Annealing of Decision Trees
  - OC1 - Oblique Classifier 1

Evaluating Decision Trees

- Method’s Appropriateness
- Data set or type
- Criteria
  - accuracy - predict class label for new data
  - scalability
    - performs model generation and prediction functions
    - large data sets
    - satisfactory speed
  - robustness
    - perform well despite noisy or missing data
  - intuitive appeal
    - results easily understood
    - promotes decision making

Decision Tree Limitations

- No backtracking
  - local optimal solution not global optimal solution
  - *lookahead* features may give us better trees
- Rectangular-shaped geometric regions
  - in two-dimensional space
    - regions bounded by lines parallel to the x- and y- axes
  - some linear relationships not parallel to the axes

Conclusions

- **Utility**
  - analyze classified data
  - produce
  - accurate and easily understood classification rules
  - with good predictive value

Improvements
  - Limitations being addressed
  - multivariate discrimination - oblique trees
  - data mining techniques

Bibliography

- *A System for Induction of Oblique Decision Trees*, Sreerama K. Murthy, Simon Kasif, Steven Salzberg, Journal of Artificial Intelligence Research 2 (1994) 1-32.
- *Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey*, Sreerama K. Murthy, Data Mining and Knowledge Discovery, 2. 345-389 (1998) Kluwer Academic Publishers.
- *Classification and Regression Trees*, Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone, 1984, Wadsworth Int. Group.
- *Computer Systems That Learn*, Sholom M. Weiss and Casimer A. Kulikowski, 1991, Morgan Kaufman.
- *Data Mining, Concepts and Techniques*, Jiawei Han and Micheline Kamber, 2001, Morgan Kaufman.
- *Introduction to Mathematical Techniques in Pattern Recognition*, Harry C. Andrews, 1972, Wiley-Interscience.
- *Machine Learning*, Tom M. Mitchell, 1997, McGraw-Hill.