**Lecture Slides for**
**INTRODUCTION TO**
**Machine Learning**

ETHEM ALPAYDIN
© The MIT Press, 2004
alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml

**CHAPTER 9: Decision Trees**

Tests at Internal Nodes, Predictions at Leaves

Divide and Conquer
- Internal decision nodes
  - Univariate: Uses a single attribute, $x_i$
    - Numeric $x_i$: Binary split: $x_i > w_m$
    - Discrete $x_i$: $n$-way split for $n$ possible values
  - Multivariate: Uses all attributes, $x$
- Leaves
  - Classification: Class labels, or proportions
  - Regression: Numeric; use average or local fit
- Learning is greedy; find the best split recursively (Breiman et al, 1984; Quinlan, 1986, 1993)

High level view:
- Want small tree fitting data for good generalization
- Occam’s razor (simplest hypothesis consistent with data is best)
- Distinguish real world from all possible situations vs. Distinguish real world from all simpler situations
- $N$ data points, $d$ real valued attributes, $k$ nodes:
  - $dN$ many different attribute < value tests
  - $(dN)^k$ many trees -- too many to search
  - Use greedy top-down search

Classification Trees (ID3, CART, C4.5)
- For node $m$, $N_m$ instances reach $m$, $N_m$ belong to $C_i$
  \[
  \hat{p}(C_i | x, m) = p_{m_i}^* = \frac{N_{m_i}}{N_m}
  \]
- Node $m$ is pure if $p_{m_i}$ is 0 or 1
- A measure of impurity is entropy
  \[
  I_m = -\sum_{m_i} p_{m_i}^* \log p_{m_i}^*
  \]
Best Split
- If node $m$ is pure, generate a leaf and stop, otherwise split and continue recursively
- Impurity after split: $N_{mj}$ of $N_m$ take branch $j$. $N_{mj}$ belong to $C_i$

$$p(C_i | x, m, j) = p_{mj} = \frac{N_{mj}}{N_m}$$

$$I_m = -\sum_{i} \frac{N_i}{N_m} \sum_{j} p_{mj} \log p_{mj}$$

- Find the variable and split with min impurity (among all variables -- and split positions for numeric variables)

Regression Trees
- Error at node $m$
  $$b_m(x) = \begin{cases} 1 & \text{if } x \in X_m : x \text{ reaches node } m \\ 0 & \text{otherwise} \end{cases}$$

$$E_m = \frac{1}{N_m} \sum (r - g_m) b_m(x)$$

- After splitting:
  $$b_m(x) = \begin{cases} 1 & \text{if } x \in X_m : x \text{ reaches node } m \text{ and branch } j \\ 0 & \text{otherwise} \end{cases}$$

$$E_m = \frac{1}{N_m} \sum (r - g_m)^2 b_m(x)$$

Pruning Trees
- Remove subtrees for better generalization (decrease variance)
  - Prepruning: Early stopping
  - Postpruning: Grow the whole tree then prune subtrees which overfit on the pruning set
- Prepruning is faster, postpruning is more accurate (requires a separate pruning set)

Rule Extraction from Trees
C4.5Rules (Quinlan, 1993)

- $x_1$: Age
- $x_2$: Years in job
- $x_3$: Gender
- $x_4$: Job-type

$$\begin{align*}
R_1: & \text{ IF } (\text{age} > 38.5 \text{ AND years-in-job} > 2.5) \text{ THEN } y = 0.8 \\
R_2: & \text{ IF } (\text{age} > 38.5 \text{ AND years-in-job} > 2.5) \text{ THEN } y = 0.6 \\
R_3: & \text{ IF } (\text{age} > 38.5 \text{ AND job-type} = \text{A}) \text{ THEN } y = 0.4 \\
R_4: & \text{ IF } (\text{age} > 38.5 \text{ AND job-type} = \text{B}) \text{ THEN } y = 0.3 \\
R_5: & \text{ IF } (\text{age} > 38.5 \text{ AND job-type} = \text{C}) \text{ THEN } y = 0.2 \\
\end{align*}$$
Learning Rules

- Rule induction is similar to tree induction but
  - tree induction is breadth-first,
  - rule induction is depth-first; one rule at a time
- Rule set contains rules; rules are conjunctions of terms
- Rule covers an example if all terms of the rule evaluate to true for the example
- Sequential covering: Generate rules one at a time until all positive examples are covered
- IREP (Fürnkranz and Widmer, 1994), Ripper (Cohen, 1995)

Multivariate Trees

Random Forests

- Pick small random subset of features to try at each node rather than exhaustive search
- Build many trees and predict with most frequent prediction
- Subset saves time, robust against missing data
- Ensemble reduces variance - don’t need pruning
- Ho ‘95, Brieman ‘01