C4.5 and Beyond

J. R. Quinlan
quinlan@rulequest.com

Source code for C4.5 is available from
http://rulequest.com/Personal/c4.5r8.tar.gz
• C4.5 follows CLS (Hunt et al, 1966) and ID3 (Quinlan, 1979)
  – generates decision tree classifiers
  – unlike CLS and ID3, can also generate ruleset classifiers

• Input:
  – set S of cases described by fixed attributes \{A_1, A_2, \ldots\},
    each belonging to one class in \{C_1, C_2, \ldots\}

• Output:
  – classifier that maps new case to class (prediction)

• Plan:
  – very broad-brush sketch of algorithms
  – key improvements in successor C5.0
  – two research problems
• **Decision trees** generated in two phases:
  – grow initial tree (divide and conquer)
  – prune to avoid overfitting

• Procedure $\text{tree}(S)$ to grow tree for set $S$ of cases:
  – if all cases in $S$ belong to same class or $|S|$ too small -
    leaf labelled with majority class in $S$
  – otherwise:
    • select test on single attribute $A_i$ with outcomes $o_1, o_2, o_3, \ldots$
    • partition $S$ into $S_1, S_2, S_3, \ldots$ according to outcomes
    • apply recursively to subsets, giving
• Criteria for evaluating tests:
  – information gain - biased towards tests with many outcomes
  – gain ratio: gain / information to determine test outcome (default)

• Format of test outcomes:
  – if attribute \( A_i \) is numeric: \( A_i \leq h, A_i > h \)
    • threshold \( h \) found by sorting \( S \) on values of \( A_i \)
      – at most \( |S|-1 \) possible values for \( h \) (usually far fewer)
      – choose \( h \) to maximise test criterion (gain or gain ratio)

  – if attribute \( A_i \) is nominal with values \( V_1, V_2, \ldots \) :
    • one outcome for each value \( V_i \) (default)
    • values partitioned into 2+ subsets (option)
• Suppose leaf has N cases, E errors (not from labelled class)
  – error rate on new cases usually higher than E / N
  – C4.5 estimates true error rate as approximate solution for p of

\[
\sum_{x=0}^{E} \binom{N}{x} p^x (1-p)^{N-x} = CF
\]

for user-specified parameter CF (default 0.25)

• Estimate error at each node of decision tree:
  – if leaf: N \times estimated error rate as above
  – otherwise, sum of estimated errors of subtrees

• Prune decision tree as follows:
  – consider each non-leaf node starting at bottom of tree
  – replace with leaf or one of subtrees if estimated error reduced

\[
\begin{align*}
U_{25\%}(0.6) &= 0.206 \\
U_{25\%}(0.1) &= 0.750 \\
U_{25\%}(0.9) &= 0.143 \\
U_{25\%}(1,16) &= 0.157
\end{align*}
\]

\[
\begin{align*}
6 \times 0.205 + 1 \times 0.750 + 9 \times 0.143 &= 3.273 \\
16 \times U_{25\%}(1,16) &= 16 \times 0.157 = 2.512
\end{align*}
\]
Example: Wisconsin breast cancer data

• Source: Dr William H. Wolberg, University of Wisconsin Hospitals
• 699 cases, two classes (2, 4)
• 9 numeric attributes:
  - Clump Thickness
  - Uniformity of Cell Size
  - Uniformity of Cell Shape
  - Marginal Adhesion
  - Single Epithelial Cell Size
  - Bare Nuclei
  - Bland Chromatin
  - Normal Nucleoli
  - Mitoses
C4.5 decision tree: (16 leaves, shown in boldface)

Uniformity of Cell Size <= 2 :
|   Bare Nuclei <= 3 :
|   |   Single Epithelial Cell Size <= 2 : 2 (380.4/1.4)
|   |   Single Epithelial Cell Size > 2 :
|   |   Uniformity of Cell Shape <= 2 : 2 (22.9/1.3)
|   |   Uniformity of Cell Shape > 2 : 4 (2.0/1.0)
|   Bare Nuclei > 3 :
|   |   Clump Thickness <= 3 : 2 (11.6/1.3)
|   |   Clump Thickness > 3 :
|   |   |   Bland Chromatin > 2 : 4 (8.1/1.3)
|   |   |   Bland Chromatin <= 2 :
|   |   |   |   Marginal Adhesion <= 3 : 4 (2.0/1.0)
|   |   |   |   Marginal Adhesion > 3 : 2 (2.0/1.0)

Uniformity of Cell Size > 2 :
|   Uniformity of Cell Shape <= 2 :
|   |   Clump Thickness <= 5 : 2 (19.0/2.5)
|   |   Clump Thickness > 5 : 4 (4.0/1.2)
|   Uniformity of Cell Shape > 2 :
|   |   Uniformity of Cell Size > 4 : 4 (177.0/7.3)
|   |   Uniformity of Cell Size <= 4 :
|   |   |   Bare Nuclei <= 2 :
|   |   |   |   Marginal Adhesion <= 3 : 2 (11.4/2.7)
|   |   |   |   Marginal Adhesion > 3 : 4 (3.0/1.1)
|   |   |   Bare Nuclei > 2 :
|   |   |   |   Clump Thickness > 6 : 4 (31.8/2.6)
|   |   |   |   Clump Thickness <= 6 :
|   |   |   |   |   Uniformity of Cell Size <= 3 : 4 (13.0/3.6)
|   |   |   |   |   Uniformity of Cell Size > 3 :
|   |   |   |   |   |   Marginal Adhesion <= 5 : 2 (5.8/2.3)
|   |   |   |   |   |   Marginal Adhesion > 5 : 4 (5.0/1.2)
• C4.5 trees differ from CART (Breiman, Friedman, Olshen, Stone) in several aspects, eg:

  – Tests:
    CART: always binary
    C4.5: any number of branches

  – Test selection criterion:
    CART: diversity index (Gini)
    C4.5: information-based criteria

  – Pruning:
    CART: cross-validated using cost-complexity model
    C4.5: single pass based on binomial confidence limits

  – Missing values (not discussed here):
    CART: surrogate tests to approximate outcome
    C4.5: case apportioned probabilistically among outcomes
• **C4.5 rulesets** formed from unpruned tree
  – each path from root to leaf gives possible simplified rule
    if A and B and C … then class X
    where A, B, C, … are conditions on path and X is class at leaf
  – use MDL (MML) to select subset of rules for each class
  – order class rulesets

• To simplify rule:
  – dropping a condition may increase coverage (N) and errors (E)
    • new estimated error may be lower or unchanged
  – if any such condition, eliminate the one giving lowest estimated error and repeat

• To classify case using ruleset:
  • check class rulesets in turn -- if case satisfies any rule in ruleset, assign case to that class
  • if no rulesets match, assign case to default class
C4.5 ruleset for Wisconsin data (8 rules vs 16 leaves)

Clump Thickness <= 3
Uniformity of Cell Size <= 2
->  class 2  [99.5%]

Clump Thickness <= 5
Uniformity of Cell Shape <= 2
->  class 2  [99.0%]

Uniformity of Cell Size <= 4
Marginal Adhesion <= 3
Bare Nuclei <= 2
->  class 2  [98.8%]

Clump Thickness > 6
Bare Nuclei > 2
->  class 4  [98.0%]

Uniformity of Cell Size > 4
->  class 4  [95.9%]

Uniformity of Cell Shape > 2
Marginal Adhesion > 3
->  class 4  [94.2%]

Uniformity of Cell Shape > 2
Bare Nuclei > 2
->  class 4  [93.6%]

Clump Thickness > 5
Uniformity of Cell Size > 2
->  class 4  [93.5%]

Default class: 2
C4.5 rulesets easier to understand, but much more computation
- one application: 10K cases require 1.4 secs (tree), 32 secs (ruleset)
- chart shows time multiplier as sample size increased to 100K
Changes introduced in See5/C5.0 (1997 - )

- **boosting** -- variant of Adaboost (Freund and Schapire)
- **variable misclassification costs** determined by predicted, true class
- **new data types** (eg: time, date, ordered nominal)
- **new value “N/A”** (not applicable) distinguished from “?” (missing)
- decision trees generally **smaller**
- **unordered rulesets** -- all relevant rules vote
  - improves both accuracy and interpretability
- mechanism to pre-select **most relevant attributes**
- **multi-threaded**: can take advantage of multiple CPUs or cores
- **greatly improved scalability**
  - eg: 100,000 training and unseen test cases, 40 attributes
    - trees: C5.0 five times faster than C4.5, tree 40% smaller
    - rules: C5.0 1,000 times faster than C4.5, 80% less memory
- more details at [http://rulequest.com/see5-comparison.html](http://rulequest.com/see5-comparison.html)
• Research issue: stable trees
  – resubstitution error rate generally much lower than leave-one-out cross-validated error rate
    eg: letter recognition dataset (20,000 cases, 26 classes)
    C4.5 error rate 4% (resubstitution), 11.7% (20,000-fold xval)
  – leaving out single case affects test selection!
  – stable tree implies resubstitution error = leave-one-out xval error
    • correct model size and higher predictive accuracy?

• Research issue: decomposing classifiers
  – ensemble classifiers (boosting, bagging, random forests, …) improve classification accuracy
  – can take, for instance, 3 bagged trees and generate single tree that is exactly equivalent (and very large)
  – can we go the other way: reconstruct given complex classifier as small ensemble of simple, voting classifiers while preserving predictive accuracy?
Greetings from Sydney, Australia