Classification and Regression Trees (CART)

Chris McConnell
Carnegie Mellon University

September 1989

Roadmap

- The concept induction problem
- History
- What CART is and how it works
- Comparison to NN approaches

Concept Induction

- Induce concepts from examples so that expert knowledge can be more easily incorporated into systems.
- The input is a collection of examples.
- The output is a procedure for classifying new examples into classes.
- Examples include: ship classification, terrain analysis, medical diagnosis, text retrieval, etc.

Input Characteristics

- Classes or a distance metric for assigning classes are supplied.
- Classes may be in multiple overlapping regions.
- Feature values can be numeric, categorical or missing.
- The distribution of examples may not reflect the distribution of data to be classified.

Output Characteristics

- **Generalize** over the examples
- **Parsimonious** and should not overfit the data.
- **Fast** at classifying new examples
- **Interpretable** so that its limitations and mistakes can be examined and understood
- **Costs** associated with making mistakes should be incorporated

History

- Traditional classification procedures like discriminant analysis allow one region per class.
- Clustering techniques have problems with the sparsity of examples in high dimensional spaces.
- Early classification procedures assumed that perfect classification was possible, attributes were over a small unordered set of discrete values and were always present. (HLS, ID3.)
- Once noisy and incomplete data sets were allowed, there was a problem with overfitting the data. Stopping rules did not work.
- CART is a recursive partitioning algorithm that grows maximal trees and then uses statistical techniques to prune the maximal tree to obtain an optimal one.

Concept Induction With CART - 4, 5

CART algorithm

- Start with a root that has all cases in training set.
- Loop until there are too few cases, only one class or identical features.
  - Search through the binary splits of node until an optimal split s_{opt} is found.
  - Split the examples in node into τ_l and τ_r. For examples that are missing A, use the splits on other attributes that best reflect the original split.
- Create a nested sequence of sub-trees at breakpoints of $R_\alpha(T) = \hat{R}(T) + \alpha\mathcal{T}$ as α increases. $\hat{R}(T)$ is the total error found in the leaves of tree T,
- Use an error estimation technique to select the optimal tree.

Splitting Nodes

- An attribute A is split by $A \leq r$ where r is between adjacent unique values for numeric attributes and by $A \in D$ where D is a subset of possible catagorical values for catagorical attributes.
- With C classes, $p(j|\tau)$ as the proportion of class j cases at node τ, and impurity function ϕ, the impurity of node τ is $\iota(\tau) = \phi(p(1|\tau), ..., p(C|\tau))$.
- The change in impurity for a split s that sends p_R cases right and p_L cases left is $\Delta\iota(s, \tau) = \iota(\tau) - p_R\iota(\tau_R) - p_L\iota(\tau_L)$
- The Gini index of diversity is:

$$\sum_{j=1}^{C} \sum_{k=1}^{C} p(k|\tau)p(j|\tau), j \neq k$$

Selecting an Optimal Tree

- If you have enough training data, hold some of it out and use it to estimate $R^*(T)$ for sub-trees.
- Otherwise, grow Z cross validation trees each using $(1 - Z)/Z$ examples and the remaining $1/Z$ of the data to estimate error rate.
- Create sub-trees for each value of original tree's α levels and find the average error of the corresponding sub-tree in the cross-validation trees.
- Select the smallest tree with the minimal error.

Example Dataset

Priors: A 0.8, B: 0.1, C: 0.1

Example	Class	N1	N2	C1	C2
0	A	***	2.5	D	Y
1	B	1.5	2.0	E	N
2	C	1.0	2.5	***	Y
3	B	1.0	2.0	E	N
4	A	1.0	2.0	F	Y

Example Splits

```
Node #1=#<Node 0 C1 (E)>
 Feature
  Value:  A-<=  B-<=  C-<=   Left   Parent Split-ra   Mirror
          A->   B->   C->    Right  Delta  No-split

N1
  1.0     1     1     1      0.24   0.34   3          T
          0     1     0      0.00   0.10
  1.5     1     2     1      0.30   0.34
          0     0     0      0.00   0.04   *
N2
  2.0     1     2     0      0.16   0.34   2          NIL
          1     0     1      0.16   0.02
  2.5     2     2     1      0.34   0.34
          0     0     0      0.0    0.00   *
C1
  (D)     1     0     0      0.0    0.34
          1     2     0      0.16   0.18
  (D E)   1     2     0      0.16   0.34
          1     0     0      0.0    0.18
  (D F)   2     0     0      0.0    0.34
          0     2     0      0.0    0.34
  (E)     0     2     0      0.0    0.34   0          NIL
          2     0     0      0.0    0.34
  (E F)   1     2     0      0.16   0.34
          1     0     0      0.0    0.18
  (F)     1     0     0      0.0    0.34
          1     2     0      0.16   0.18
C2
  (Y)     2     0     1      0.18   0.34   1          T
          0     2     0      0.0    0.16
  (N)     0     2     0      0.0    0.34
          2     0     1      0.18   0.16
```

Final tree

```
If C1 <- (E)
   C2 <> (Y)
   N2 <= 2.25
   N1 > 1.25 then (B)
else
   If N2 <= 2.25 then (A)
   else ((A 0.8) (C 0.2))
```

Example of Pruning

```
If PETAL-WIDTH <= 8.0 then (SETOSA)
else
   If PETAL-WIDTH <= 17.5 then
      If PETAL-LENGTH <= 49.5 then
         If PETAL-WIDTH <= 16.5 then (VERSICOLOR)
         else (VIRGINICA)
      else
         If PETAL-WIDTH <= 15.5 then (VIRGINICA)
         else
            If SEPAL-LENGTH <= 69.5 then (VERSICOLOR)
            else (VIRGINICA)
   else
      If PETAL-LENGTH <= 48.5 then
         If SEPAL-WIDTH <= 31.0 then (VIRGINICA)
         else (VERSICOLOR)
      else (VIRGINICA)

If PETAL-WIDTH <= 8.0 then (SETOSA)
else
   If PETAL-WIDTH <= 17.5 then ((VERSICOLOR 0.91) (VIRGINICA 0.09))
      else ((VIRIGINICA 0.98) (VERSICOLOR 0.02))
```

CART

- Advantages
  - Only one pass through the training set.
  - Allows the specification of prior probabilities and misclassification costs.
  - Easily interpretable decision trees.
  - Doesn't overfit the data.
  - Can assign probability of class membership.
- Disadvantages
  - Cannot handle non-linear combination of features.

Neural Nets

- Advantages
  - Non-linear combination of features.
  - Possibility of compact dynamic representations.
- Disadvantages
  - Overfits data.
  - The results are not very interpretable.
  - Cannot incorporate priors and costs.
  - Requires many (an unpredictable number of) passes through the data.
  - Size of the network is unknown.

Conclusion

CART is a better choice than a NN approach if classes can be distinguished by a linear combination of features.