Introduction to Data Mining with CART


Dan Steinberg
Mykhaylo Golovnya
support@salford-systems.com

August 12, 2015
In 1984 Berkeley and Stanford statisticians announced a remarkable new classification tool which:

- Could separate relevant from irrelevant predictors
- Did not require any kind of variable transforms (logs, square roots)
- Impervious to outliers and missing values
- Could yield relatively simple and easy to comprehend models
- Required little to no supervision by the analyst
- Was frequently more accurate than traditional logistic regression or discriminant analysis, and other parametric tools
Ship Recognition Problem

Profile

Detect

Classify
CART was slow to gain widespread recognition in late 1980s and early 1990s for several reasons

- Monograph introducing CART is challenging to read
  - Brilliant book overflowing with insights into tree growing methodology but fairly technical and brief
- Method was not expounded in any textbooks
- Originally taught only in advanced graduate statistics classes at a handful of leading Universities
- Original standalone software came with slender documentation, and output was not self-explanatory
- Method was such a radical departure from conventional statistics
Rising interest in data mining over the past decade

- Availability of huge data sets requiring analysis
- Need to automate or accelerate and improve analysis process

Advantages of CART over other tree methods

- Handling of missing values
- Assistance in interpretation of results (surrogates)
- Performance advantages: speed, accuracy, scalability

New software and documentation make techniques accessible to end users

Word of mouth generated by early adopters

Easy to use professional software available from Salford Systems
The algorithm can be summarized as:

- For each current data region, consider all possible orthogonal splits (based on one variable) into 2 sub-regions
- The best split is defined as the one having the smallest MSE after fitting a constant in each sub-region (regression) or the smallest resulting class impurity (classification)
- Proceed recursively until all structure in the training set has been completely exhausted \( \rightarrow \) largest tree is produced
- Create a sequence of nested sub-trees with different amount of localization (tree pruning)
- Pick the best tree based on the performance on a test set or cross-validated

One can view CART tree as a set of dynamically constructed orthogonal nearest neighbor boxes of varying sizes guided by the response variable (homogeneity of response within each box)
Splitting rule controls the tree growing stage

At each node, a finite pool of all possible splits exists

- The pool is independent of the target variable, it is based solely on the observations (and corresponding values of predictors) in the node

The task of the splitting rule is to rank all splits in the pool according to their improvements

- The winner will become the main splitter
- The top runner-ups will become competitors

It is possible to construct different splitting rules emphasizing various approaches to the definition of the “best split”
CART is best illustrated with a famous example – the UCSD Heart Disease study

- Given the diagnosis of a heart attack based on
  - Chest pain, Indicative EKGs, Elevation of enzymes typically released by damaged heart muscle, etc.
- Predict who is at risk of a 2nd heart attack and early death within 30 days
- Prediction will determine treatment program (intensive care or not)

For each patient about 100 variables were available, including:

- Demographics, medical history, lab results
- 19 noninvasive variables were used in the analysis
  - Age, gender, blood pressure, heart rate, etc.

CART discovered a very useful model utilizing only 3 final variables!
Example of a CLASSIFICATION tree:

- Dependent variable is categorical (SURVIVE, DIE).
- The model structure is inherently hierarchical and cannot be represented by an equivalent logistic regression equation.
- Each terminal node describes a segment in the population.

**PATIENTS = 215**

Is BP<=91?

- **Terminal Node A**
  - SURVIVE: 6 (30.0%)
  - DEAD: 14 (70.0%)
  - NODE = DEAD

- **PATIENTS = 195**
  - Is AGE<=62.5?
    - **Terminal Node B**
      - SURVIVE: 102 (98.1%)
      - DEAD: 2 (1.9%)
      - NODE = SURVIVE
    - **Terminal Node C**
      - SURVIVE: 14 (50.0%)
      - DEAD: 14 (50.0%)
      - NODE = DEAD

- **PATIENTS = 91**
  - Is SINUS<=.5?
    - **Terminal Node D**
      - SURVIVE: 56 (88.9%)
      - DEAD: 7 (11.1%)
      - NODE = SURVIVE

All internal splits are binary.
Rules can be extracted to describe each terminal node.
Terminal node class assignment is determined by the distribution of the target in the node itself.
The tree effectively compresses the decision logic.
General Workflow

Stage 1
Historical Data
Build a Sequence of Nested Trees

Stage 2
Test
Monitor Performance
Best

Stage 3
Validate
Confirm Findings

Learn
Question is of the form: is statement TRUE?

Is continuous variable $X \leq c$ ?

Does categorical variable D take on levels i, j, or k?
  ✦ e.g. Is geographic region 1, 2, 4, or 7?

Standard split:
  ✦ If answer to question is YES a case goes left; otherwise it goes right
  ✦ This is the form of all primary splits

Question is formulated so that only two answers possible
  ✦ Called binary partitioning
  ✦ In CART the YES answer always goes left
Classification is determined by following a case’s path down the tree

- Terminal nodes are associated with a single class
  - Any case arriving at a terminal node is assigned to that class

- The implied classification threshold is constant across all nodes and depends on the currently set priors, costs, and observation weights (discussed later)
  - Other algorithms often use only the majority rule or a limited set of fixed rules for node class assignment thus being less flexible

- In standard classification, tree assignment is not probabilistic
  - Another type of CART tree, the class probability tree does report distributions of class membership in nodes (discussed later)

- With large data sets we can take the empirical distribution in a terminal node to represent the distribution of classes
Regression trees result to piece-wise constant models (multi-dimensional staircase) on an orthogonal partition of the data space

- Thus usually not the best possible performer in terms of conventional regression loss functions

Only a very limited number of controls is available to influence the modeling process

- Priors and costs are no longer applicable
- There are two splitting rules: LS and LAD

Very powerful in capturing high-order interactions but somewhat weak in explaining simple main effects
Split Improvement

\[ \sum_{i \in P} (y_i - \bar{y}_P)^2 \]

\[ \sum_{i \in L} (y_i - \bar{y}_L)^2 \]

\[ \sum_{i \in R} (y_i - \bar{y}_R)^2 \]

- Parent node \[ SSE_P = \sum_{i \in P} (y_i - \bar{y}_P)^2 \]
- Left child \[ SSE_L = \sum_{i \in L} (y_i - \bar{y}_L)^2 \]
- Right child \[ SSE_R = \sum_{i \in R} (y_i - \bar{y}_R)^2 \]
- Improvement \[ \Delta = SSE_P - SSE_L - SSE_R \]
- One can show \[ \Delta = \frac{N_P}{N_LN_R} (\bar{y}_L - \bar{y}_R)^2 \]

Find the split with the largest improvement by conducting exhaustive search over all splits in the parent node.
Concerns the housing values in Boston area


Combined information from 10 separate governmental and educational sources to produce this data set

506 census tracts in City of Boston for the year 1970

- **Goal**: study relationship between quality of life variables and property values
- **MV**: median value of owner-occupied homes in tract ($1,000's)
- **CRIM**: per capita crime rates
- **NOX**: concentration of nitric oxides (pp 10 million)
- **AGE**: percent built before 1940
- **DIS**: weighted distance to centers of employment
- **RM**: average number of rooms per house
- **LSTAT**: % lower status of the population
- **RAD**: accessibility to radial highways
- **CHAS**: borders Charles River (0/1)
- **INDUS**: percent non-retail business
- **TAX**: property tax rate per $10,000
- **PT**: pupil teacher ratio
The distribution of the target variable (in thousands $)

Clear manifestation of the inflation over the past 40 years
The data violates all conventional modeling assumptions

Clearly some non-normal distributions and non-linear relationships
Improvement is defined in terms of the greatest reduction in the sum of squared errors when a single constant prediction is replaced by two separate constants on each side.
Regression Tree Model

- All cases in the given node are assigned the same predicted response – the node average of the original target
- Nodes are color-coded according to the predicted response
- We have a convenient segmentation of the population according to the average response levels
The Best and the Worst Segments

Terminal Node 10

Node Statistics

- Min: 21.90000
- Q1: 42.67500
- Mean: 45.09667
- Median: 46.35000
- Q3: 50.00000
- Max: 50.00000
- Cases: 30

Rules for terminal node 10:

```c
if
{
  RM > 7.437
}
{
  terminalNode = -10;
  mean = 45.0967
}
```

Terminal Node 8

Node Statistics

- Min: 5.00000
- Q1: 8.40000
- Mean: 10.70392
- Median: 10.50000
- Q3: 13.00000
- Max: 17.80000
- Cases: 51

Rules for terminal node 8:

```c
if
{
  RM <= 6.941 &&
  LSTAT > 14.4 &&
  CRIM > 6.99237 &&
  NOX >= 0.675
}
{
  terminalNode = -8;
  mean = 10.7039
}
```
Some researchers suggested using 3-way, 4-way, etc. splits at each node (univariate)

There are important theoretical and practical reasons why we feel strongly against such approaches:

- Exhaustive search for a binary split has linear algorithm complexity, the alternatives have much higher complexity
- Choosing the best binary split is “less ambiguous” than the suggested alternatives
- Most importantly, binary splits result to the smallest rate of data fragmentation thus allowing a larger set of predictors to enter a model
- Any multi-way split can be replaced by a sequence of binary splits

There are numerous practical illustrations on the validity of the above claims
Sample node details for a CART split are shown above.

- **Improvement** measures class separation between two sides, the main splitter by definition has the largest calculated improvement.

- **Competitors** are next best splits in the given node ordered by improvement values.

- **Surrogates** are splits most resembling (case by case) the main split.
  - Association measures the degree of resemblance.
  - Looking for competitors is free, finding surrogates is expensive.
  - It is possible to have an empty list of surrogates.

- Variable FIT is a perfect surrogate (and competitor) for CLASSES.
Competitors and Surrogates Are Different

For symmetry considerations, both splits must result to identical improvements.

Yet one split can’t be substituted for another.

Hence Split X and Split Y are perfect competitors (same improvement) but very poor surrogates (low association).

The reverse does not hold – a perfect surrogate is always a perfect competitor.

- Three nominal classes
- Equally present
- Equal priors, unit costs, etc.
The Utility of Surrogates

- A major part of internal CART machinery is centered around the surrogates and missing value handling.
- Suppose that the main splitter is INCOME and comes from a survey form.
- People with very low or very high income are likely not to report it—informative missingness in both tails of the income distribution.
- Regardless of the split value, the nature of partitioning here is to separate low income on the left from high income on the right—hence, it is unreasonable to treat missing INCOME as a separate unique value being sent to one chosen side (usual way to handle such case).
- Using a surrogate helps to resolve ambiguity and redistribute missing income records between the two sides.
- For example, all low education subjects will join the low income side while all high education subjects will join the high income side.
- One way to look at this is utilizing local rank-correlation between INCOME and other related features to redistribute missing INCOME records between BOTH sides of the split.
Tree Interpretation and Use

- Practical decision tool
  - Trees like this are used for real world decision making
    - Rules for physicians
    - Decision tool for a nationwide team of salesmen needing to classify potential customers

- Selection of the most important prognostic variables
  - Variable screen for parametric model building
  - Example data set had 100 variables
  - Use results to find variables to include in a logistic regression

- Insight — in our example AGE is not relevant if BP is not high
  - Suggests which interactions will be important

- **Missing Value Imputation** using “rotating target” approach
CART advantages:

- One of the fastest data mining algorithms available
- Requires minimal supervision and produces easy to understand models
- Focuses on finding interactions and signal discontinuities
- Important variables are automatically identified
- Handles missing values via surrogate splits
  - A surrogate split is an alternative decision rule supporting the main rule by exploiting local rank-correlation in a node
- Invariant to monotone transformations of predictors

CART disadvantages:

- Model structure is fundamentally different from conventional modeling paradigms – may confuse reviewers and classical modelers
- Has limited number of positions to accommodate available predictors – ineffective at presenting global linear structure (but great for interactions)
- Produces coarse-grained piece-wise constant response surfaces
Real world study early 1990s

Fixed line service provider, offering a new mobile phone service, wants to identify customers most likely to accept new mobile offer

Data set based on limited market trial

- 830 households from 5 different cities offered a mobile phone package
- All offered identical package but pricing was varied at random
  - Handset prices ranged from low to high values
  - Per minute prices ranged separately from low to high rate rates
- Household asked to make yes or no decision on offer (15.2% accepted the offer)

Our goal was to answer two key questions

- Who to make offers to?
- How to price?

Variables collected included

- CITY, SEX – location and gender information
- TELEBILL – average monthly phone bill
- HANDPRIC – handset price offered
- RESPONSE – whether or not subscribe for wireless service (about 15%)
10-node CART tree was built on the cell phone dataset introduced above.

The root Node 1 displays details of TARGET variable in the training data:
- 15.2% of the 830 households accepted the marketing offer.

CART tried all available predictors one at a time and found out that partitioning the set of subjects based on the Handset Price variable is most effective at separating responders from non-responders at this point:
- Those offered the phone with a price > 130 contain only 9.9% responders.
- Those offered a lower price < 130 respond at 21.9%.

The process of splitting continues recursively until the largest tree is grown.

Subsequent tree pruning eliminates least important branches and creates a sequence of nested trees – candidate models.
The red nodes indicate good responders while the blue nodes indicate poor responders.

Observations with high values on a split variable always go right while those with low values go left.

Terminal nodes are numbered left to right and provide the following useful insights:

- **Node 1:** young prospects having very small phone bill, living in specific cities are likely to respond to an offer with a cheap handset.
- **Node 5:** mature prospects having small phone bill, living in specific cities (opposite of Node 1) are likely to respond to an offer with a cheap handset.
- **Nodes 6 and 8:** prospects with large phone bill are likely to respond as long as the handset is cheap.
- **Node 10:** “high-tech” prospects (having a pager) with large phone bill are likely to respond to even offers with expensive handset.

//Rules for terminal node 10
if
{
    
    PAGER == 1
    
    HANDPRIC > 130 &
    
    TELEBILC > 50

    
}
{ terminalNode = 10;
    class = 1;
    probClass1 = 0.590909;
    probClass2 = 0.409091;
Variable Importance and Predictive Accuracy

A number of variables were identified as important

- Note the presence of surrogates not seen on the main tree diagram previously

Prediction Success table reports classification accuracy on the test sample

- Top decile (10% of the population with the highest scores) captures 40% of the responders (lift of 4)
The following procedure is used by default to calculate variable importance in CART:

- List all main splitters as well as all surrogates along with the corresponding improvements.
- Aggregate the improvements by variable.
- Sort the result descending and scale such that the largest value is 100%.

Variable importance is based on the current tree – a different tree will have different variable importance values.

- Larger trees will tend to include more variables.

Variable importance reveals a deeper structure in stark contrast to a casual look at the main splitters.
The task of finding a segment rich in a certain outcome of interest often arises in practice

- Fraud detection: important to identify conditions leading to fraud with high certainty
- Direct marketing: important to identify segments very likely to respond when the campaign cost is high
- Drug research: important to search for most active compounds

All such tasks are generally known as Hotspot Detection

Many different approaches exist, most notably Jerry Friedman’s PRIM algorithm that proceeds by building multidimensional boxes “encasing” hot spots

In what follows we demonstrate that a similar solution can be obtained effectively with CART trees using the mechanism of priors
In CART the within node probability of class $j$ is calculated as 

$$p_i(t) = \left( \frac{n_i}{N_i} \right) \frac{\pi_i}{p_t}$$

- $\pi_i$ = prior probability of class $j$
- $n_i$ = node count of class $j$
- $N_i$ = sample count of class $j$
- $p_t$ = overall probability of node $t$

Internal class assignment rule is always the same: 
assign node to the class with the highest $p_i(t)$

For PRIORS DATA this simplifies to the simple node majority rule: 

$$p_i(t) \propto n_i$$

For PRIORS EQUAL this simplifies to class assignment with the highest lift with respect to the root node presence: 

$$p_i(t) \propto \frac{n_i}{N_i}$$

More generally, the smaller the prior the greater the requirement on the node class count – hence, reducing prior probability has a tendency to produce richer nodes in the outcome of interest.
General Rules

- As prior on the given class decreases \( \downarrow \)
- The class assignment threshold increases. \( \uparrow \)
- Node richness goes up, \( \uparrow \)
- But class accuracy goes down \( \downarrow \)
- PRIORS EQUAL uses the root node class ratio as the class assignment threshold – hence, most favorable conditions to build a tree
- PRIORS DATA uses the majority rule as the class assignment threshold – hence, difficult modeling conditions on unbalanced classes
- In reality, a proper combination of priors can be found experimentally
- Eventually, when priors are too extreme, CART will refuse to build a tree
  - Often the hottest spot is a single node in the tree built with the most extreme priors with which CART will still build a tree
  - Comparing hotspots in successive trees can be informative, particularly in moderately-sized data sets
We have a mixture of two overlapping classes

The vertical lines show root node splits for different sets of priors. (The left child is classified as red, the right child is classified as blue)

Varying priors provides effective control over the tradeoff between class purity and class accuracy
The powerful mechanism of priors is at the core of the tree building mechanism.

Here we report the results of an experiment with prior on responders varying from 0.05 to 0.95 in increments of 0.05.

The resulting CART models “sweep” the modeling space enforcing different sensitivity-specificity tradeoff and varying node richness.
Recall that hot spots are areas of data very rich in the event of interest, even though they could only cover a small fraction of the targeted group.

The varying-priors collection of runs introduced above gives perfect raw material in the search for hot spots:

- Simply look at all terminal nodes across all trees in the collection and identify the highest response segments.
- Also want to have such segments as large as possible.
- Once identified, the rules leading to such segments (nodes) are easily available.

The graph on the left reports all nodes according to their target coverage and lift.

The blue curve connects the nodes most likely to be a hot spot.
Our next experiment (variable shaving) runs as follows:

- Build a CART model with the full set of predictors
- Check the variable importance, remove the least important variable and rebuild CART model
- Repeat previous step until all variables have been removed

Six-variable model has the best performance so far

Alternative shaving techniques include:

- Proceed by removing the most important variable – useful in removal of model “hijackers” – variables looking very strong on the train data but failing on the test data (e.g. ID variable)
- Set up nested looping to remove redundant variables from the inner positions on the variable importance list
Many predictive models can benefit from Salford Systems patent on “Structured Trees”

Trees constrained in how they are grown to reflect decision support requirements

- Variables allowed/disallowed depending on a level in a tree
- Variable allowed/disallowed depending on a node size

In mobile phone example: want tree to first segment on customer characteristics and then complete using price variables

- Price variables are under the control of the company
- Customer characteristics are beyond company control

The top part of the tree partitions the universe of prospects into different segments

Then the bottom branching within each segment suggests the best pricing strategy to consider
Various areas of research were spawned by CART
We report on some of the most interesting and well developed approaches
Hybrid models
  - Combining CART with Linear and Logistic Regression
  - Combining CART with Neural Nets
Linear combination splits
Committees of trees
  - Bagging
  - Arcing
  - Random Forest
Stochastic Gradient Boosting (MART a.k.a. TreeNet)
Rule Fit and Path Finder
Download a current version from our website [http://www.salford-systems.com](http://www.salford-systems.com)

Version will run without a license key for 10-days

Request a license key from [unlock@salford-systems.com](mailto:unlock@salford-systems.com)

Request configuration to meet your needs
- Data handling capacity
- Data mining engines made available


