ACT Study report using Classification and Regression Tree (CART) Analysis

Dr. Robert M. Simmonds
US Army Logistics Management College
Systems Engineering Department
2401 Quarters Road. Fort Lee, VA 23801-1705
simmondr@lee.army.mil

I used a recursive partitioning algorithm for the CoIL-Challenge data. The software incorporating the recursive partitioning algorithm is a Classification and Regression Tree (CART) product produced by Salford Systems Inc., called CART. CART is a robust data-analysis tool that automatically searches for important patterns and relationships and quickly uncovers hidden structure even in highly complex data. The selected CART tree is then used to predict future outcomes based on new data. In this case, the prediction set provided on the CoIL-Challenge Web site. In addition, the patterns found by CART can be restated as rules of thumb and integrated into other software development efforts for automating the prediction process of new data from a user-friendly environment. The benefit to a CART analysis is the ability to explain the how the software arrived at the final structured tree. This feature is not found in other intelligent techniques such as neural networks. The version of CART used in developing this model ran on the Windows NT 4.0 platform.

The CART software provides several algorithms in partitioning data for categorical data. Given the classification problem outlined in the CoIL-Challenge data (i.e., "purchase" or "no purchase"), the Gini algorithm provided the best out put given several models developed during this exercise to match patterns among independent variables and the dependent variable (purchase). CART is a nonparametric procedure. The procedure does not require one to select variables in advance of the analysis. In fact, recursive partitioning has been used as a method of variable reduction. In this problem, I used CART to reduce the variables in the final model from 80 to twelve. The final twelve variables were: PPERSAUT, APERSAUT, PBRAND, PWAPART, AWAPART, ABBRAND, MINKGEM, MINKM30, MHKOOP, MHHUUR, MINK4575, AND MOPLLAAG. CART process cases with missing values for predictors by developing surrogate splitters at each split in the tree. Two important points about the CART software from Salford systems is its ability to identify an optimal tree, one providing the least amount of error between the learning data set and the test data set.

I selected the Gini algorithm from the CART techniques to use as the splitting rule for the CoIL-Challenge data set. I loaded the entire "ticdata2000" data set into CART and allowed the software to randomly select 35% of the 5,822 records as a test set. The remaining records were used as the learning set. Based on these records, I received the following:

TEST SAMPLE CLASSIFICATION TABLE

```
ACTUAL  PREDICTED CLASS                ACTUAL
CLASS          0            1          TOTAL
 --------------------------------------------------
          0     1006.000      858.000     1864.000
          1       25.000      101.000      126.000
 --------------------------------------------------
 PRED. TOT.       1031.000      959.000     1990.000
 CORRECT             0.540        0.802
 SUCCESS IND.       -0.397        0.738
 TOT. CORRECT        0.556
```

The Gini algorithm splits the data to find terminal nodes that reduce the impurity factor of the original data set. Reducing the impurity of the data at the terminal node is used to explain the pattern that exists in the data. Basically as we understand the relationships between the independent variables, we begin to see patterns appear that explain why some individuals purchase caravan insurance while others do not purchase the insurance policy.

The Gini algorithm is expressed as: I(t) = 1 - S, where S is the sum of the squared probabilities p(j|t) summed over all levels of the dependent variable. As an example, consider a data set where the dependent variable consists of three classifications such that the distirbution is 1/4, 1/4, 1/2. The Gini impurity index would be I(t) = 1 - (1/16 + 1/16 + 1/4) = 1-.374 = .626 and would be interpreted to have a high impurity index, an index close to one (1). As an index approaches zero, the indication is that a particular node has less impurity and as a result the pattern is better in explaining how the data came to be in this node. In building the CART model, I selected the Gini algorithm as the splitting rule, identified the cost of a miss as 1 1/2 times as costly as a false alarm. A miss identified as predicting "no purchaset" and there was actually a "purchase"; where a false alarm was predicting a "purchase" and there was "no purchase.

I have a full analysis available upon request that outlines the tree statistics as well as each node in the tree. The explanation of why one does or does not purchases the caravan insurance can be found in a sample rule based generated from the tree analysis. These rules only highlight some of the variable relationships in the overall tree.

```
/*Terminal Node 1*/
if
(
    MINKM30 <= 2.5 &&
    PBRAND <= 2.5 &&
    MHHUUR <= 0.5 &&
    PPERSAUT <= 2
)
{
    terminalNode = -1;
    class = 1;
}

/*Terminal Node 2*/
if
(
    MINKM30 <= 2.5 &&
    PBRAND <= 2.5 &&
    MHHUUR <= 0.5 &&
    PPERSAUT > 2 &&
    PPERSAUT <= 5.5
)
{
    terminalNode = -2;
    class = 0;
}

/*Terminal Node 3*/
if
(
    PPERSAUT <= 5.5 &&
    MINKM30 <= 2.5 &&
    PBRAND <= 2.5 &&
    MHHUUR > 0.5
)
{
    terminalNode = -3;
    class = 0;
}

/*Terminal Node 4*/
if
```

```
(
    PPERSAUT <= 5.5 &&
    PBRAND > 2.5 &&
    PBRAND <= 4.5 &&
    MHHUUR <= 0.5 &&
    MINKM30 <= 0.5
)
{
    terminalNode = -4;
    class = 0;
}

/*Terminal Node 5*/
if
(
    PPERSAUT <= 5.5 &&
    PBRAND > 2.5 &&
    PBRAND <= 4.5 &&
    MHHUUR <= 0.5 &&
    MINKM30 > 0.5 &&
    MINKM30 <= 2.5
)
{
    terminalNode = -5;
    class = 1;
```