

# *Improving Data Mining with New Hybrid Methods*

**Dan Steinberg**  
**N. Scott Cardell**



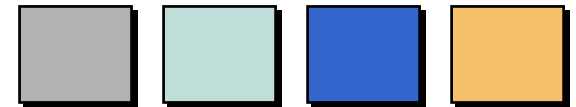
<http://www.salford-systems.com>

# Central Data Mining Tasks

## Generic task is “Pattern Recognition”

- **Classification** (assign object to class)
  - Binary:** Response yes/no?
  - Multi-class:** loan outcome: delinquent, bankrupt, paid
- **Class Probability** (assign a probability to each class outcome)
- **Regression** (model a continuous response)
  - ♦ dollars spent on a catalog
  - ♦ length of time customer will remain loyal
- **Clustering** (finding groups in data)
- **Association** (market basket analysis)

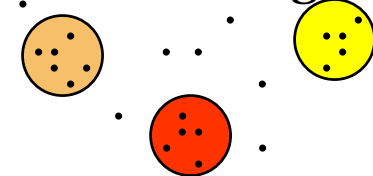
Classification: Which box?



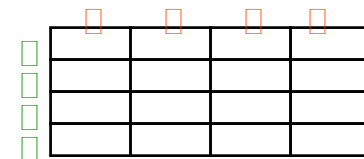
Regression



Clustering



Association




# *Tasks can be accomplished by a number of tools*

- **Classification and regression can be accomplished with**
  - **CART decision trees (other trees: CHAID, C4.5, & variations)**
  - **Statistical tools (Logit, Regression, Discriminant Analysis)**
  - **Neural Nets**
- **Clustering can be conducted using**
  - **Statistical tools (many alternatives available)**
  - **Self-organizing maps (Neural Net variant)**
- **Association**
  - **Various algorithms**
  - **CART decision trees (relatively new extension)**

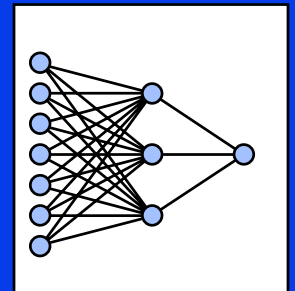
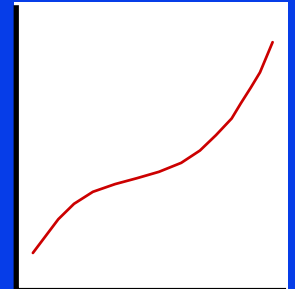
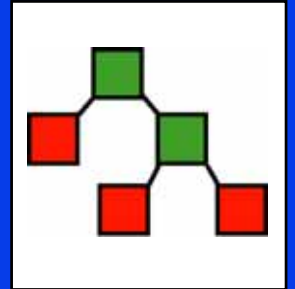
# *No one tool is best for all problems*

- Even for a single category of problem such as classification, no single tool is best
- Best tool depends on the nature of the data and the patterns we are trying to discover
  - Only common broad-scale structures
  - Multiple structures for different subsets of data
- Best tool depends on quality of the data
  - How prevalent are missing values?
  - How often are fields in error?
  - Are the target variables ever misrecorded? How frequently?
- Best tool depends on how long we can afford to wait for an answer
  - expert resources available or not

Age	Income	Spend	Credit
?	?	895	610
32	Bad Value	35	Bad Value
21	46200	?	490
16	21900	176	?
29	74300	Bad Value	530

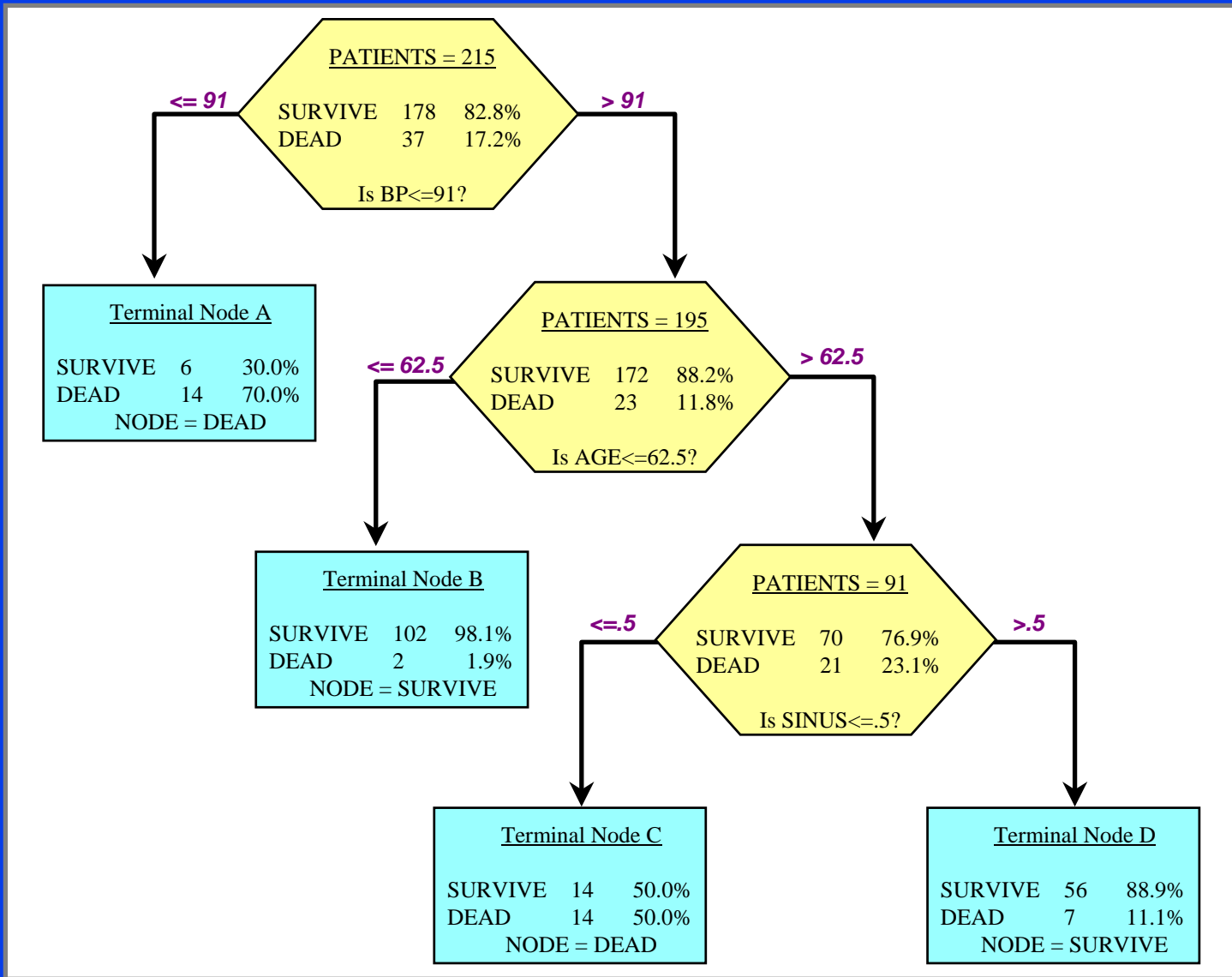
# *CART, LOGIT, Neural Nets:* *Three key tools*

- **CART - nontraditional decision tree methodology**
  - high degree of automation
  - communication via pictures
  - ability to handle arbitrarily complex data structures
  - relatively easy to use and understand even by non-statisticians
  - demonstrates remarkable accuracy in a broad range of contexts
- **LOGIT - traditional methodology relying on classical statistical principles**
  - requires expert to develop hand-crafted models
  - can often be understood only via simulation
  - ability to handle linear and smooth curvilinear data structures
  - demonstrates remarkable accuracy in a broad range of contexts
- **Neural Nets - possibly the best known data mining tool**
  - Can be thought of as a highly complex collection of Logits
  - Shares all characteristics of Logit model listed above for DM
  - Shares with CART ability to handle complex data and high accuracy



# *What is CART™? A brief overview*

- **Origins in research conducted at Berkeley & Stanford**
  - **Leo Breiman, University of California, Berkeley**
  - **Jerry Friedman, Stanford University**
  - **Charles J. Stone, University of California, Berkeley**
  - **Richard Olshen, Stanford University**
- **Monograph published in 1984 introduced a new decision tree procedure**
- **Provided detailed mathematical foundations for their approach**
  - **most remarkable: proofs for when CART gets the right answer; such proofs not available for any other decision tree**
- **Solved a number of problems plaguing other decision tree methods (CHAID, ID3)**
- **Very well known in biomedical and engineering arenas**
- **Only recently becoming known in IT, DM, and AI circles**



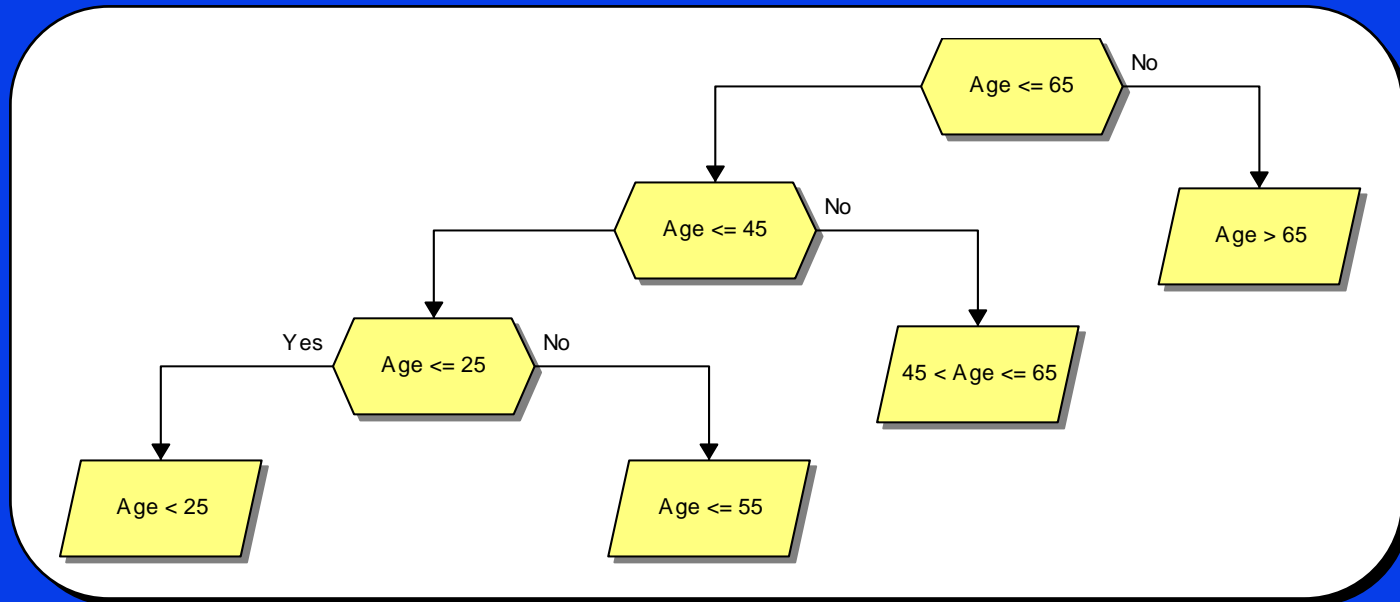
# *CART Decision Tree Innovations: How to grow a tree*

- **Grow a Tree by Splitting a database into partitions**
- **Alternative splitting rules -- Gini, Twoing, Entropy**
  - ◆ **these alternative rules do make a difference**
  - ◆ **in contrast to conventional wisdom you need a variety of splitting rules to get best results**
  - ◆ **Gini -- new with CART is usually best for yes/no outcomes**
  - ◆ **Twoing - new in CART -- similar to entropy but more flexible because it has a tuning parameter**
    - **excellent for multi-class outcomes**
    - **which of 5 segments does a record belong to**
    - **which of 19 minivans will a customer purchase**
    - **twoing excellent for hard to classify problems**
      - \* **problems where accuracy for all methods will be low**
      - \* **inherently difficult problems or low signal/noise ratio**
  - ◆ **Entropy- popular in Machine Learning literature**

# ***CART: Grow a tree via binary partitioning***

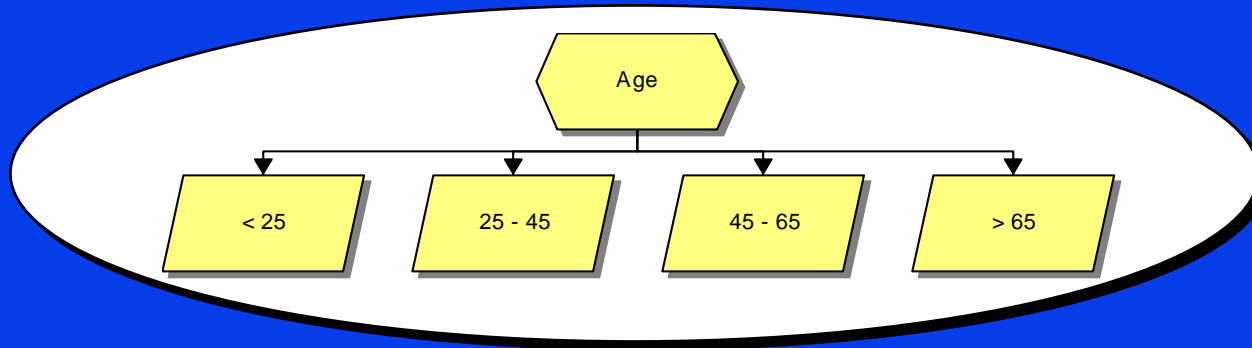
- **Strictly binary splits are guaranteed to be better!**
- **A binary splitting procedure can always reproduce a multi-way split**
- **A binary splitting procedure will only partially partition on a database field if another sequence is better**

# *Binary Split Revelation of a multi-way split*



**If the multi-way split is best, binary split method will find it**  
**If it is not best, binary method will move to another variable**

# *Multi-way split*



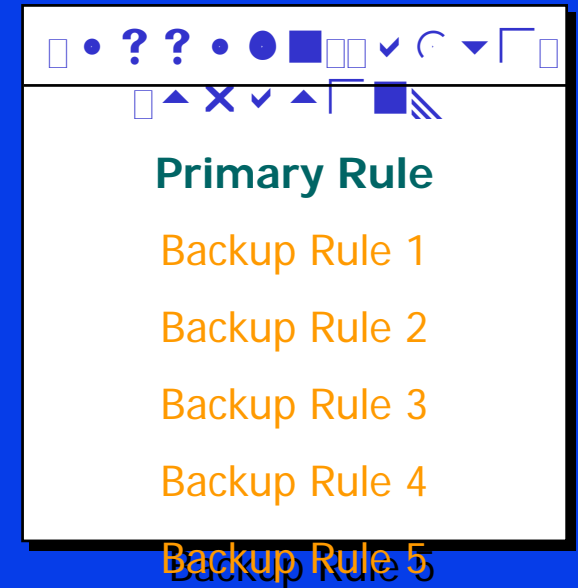
- **Split made all at once could be too hasty**
- **Even if age group *is* different, other variables might be even more valuable after the Age > 65, Age < 65 split**
- **The database is fragmented rapidly**
- **Even with 500,000 records, 5 consecutive 4-way splits leave about 2,000 records per partition**
- **Binary splits are more patient, giving a better chance to find important structure**

# *CART stopping rule innovation: Don't stop!*

- **CART trees are grown overly large deliberately**
- **Why? Impossible to find a stopping rule that works in all circumstances**
- **Only way to be sure all structure is discovered is to drill too deep and then back up**
- **CART introduced the “grow and then prune back” methodology**
- **In data mining with large databases test data available to ascertain precise size of tree use for best accuracy**

# *Final CART innovations: Missing value handling and costs*

- **CART does not require any special handling of data to deal with missing values**
- **Sophisticated surrogate splitting rule methodology finds backup rules to use when primary rule encounters a missing value**
- **Multiple backup rules found; if one fails, another will be triggered automatically**
- **Tests show that when data are missing at random even 25% missing rates have minimal effect on CART accuracy**
- **Costs of misclassification: allow for certain errors to be more serious than others**
- **CART will grow tree designed to avoid most serious mistakes**
- **Other DM tools cannot distinguish between important and unimportant errors and treat all errors as equally bad**



# ***CART strengths vs. LOGIT and NNs***

- **CART selects variables automatically (unlike LOGIT or NNs)**
  - critical when database contains hundreds of fields
  - alternative is manual search which can take weeks or months
- **CART impervious to outliers (handle dirty data-- unlike LOGIT)**
  - most corporate databases contain fields with questionable values
  - if outliers need to be inspected, progress to modeling will be slow
- **CART unaffected by missing values (unlike LOGIT or NNs)**
  - CART accuracy only slightly reduced even with many missings
  - LOGIT and NNs require special handling of any record with missing fields
  - LOGIT and NNs can suffer severe accuracy loss
  - CART automatically finds backup stand-in for missing fields
- **CART can handle explicit costs of misclassification**
- **CART is very high speed: NNs can be glacially slow to train**
  - even on a database with 300,000 records and 500 fields, CART tree will grow in minutes rather than hours, days or months (NNs)

# *Further CART strengths shared with NNs vs LOGIT*

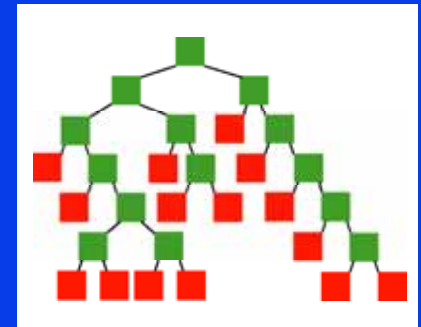
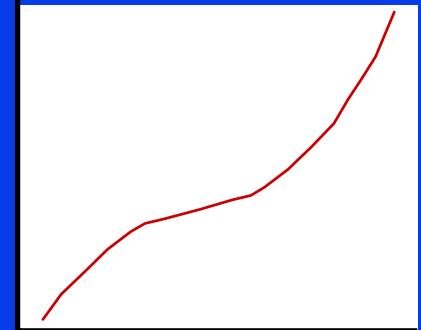
- **Model form is automatically determined by CART and NN**
  - Do not require data transforms such as log, square root
  - statistical models (LOGIT) require painstaking search for best “form”
- **Automatic interaction detection (model specification)**
  - both CART and NNs detect interactions automatically
- **Both CART and NNs can detect complex local data structure**
  - LOGIT not well suited for local sub-sample structure detection
  - can only be detected by laborious manual search
- **Both CART and NNs can converge (capture) virtually any data structure or pattern given sufficient data**
  - however, CART is always computationally feasible
  - in practice a NN may not be computationally feasible

# *Value of LOGIT in data mining*

- **Logit can be an excellent performer in classification**
- **In STATLOG project, variations of logistic discriminant analysis were most accurate in 5 out of 21 problems**
  - among top 5 performers in 12 out of 21 problems
  - NNs in top 5 performers 14 out of 21 problems
    - ♦ But NN was an outright winner only once in 21 problems
- **Provides a smooth, continuous predicted probability of class membership**
  - A small change in a predictor variable yields a small change in predicted probability
- **Effective capture of global features of data**
  - Main effects model reflects how probability responds to predictor  $x$  over entire range of predictor
  - Modest adaptability to local structure

# *CART, LOGIT and Neural Nets excel at different tasks*

- **LOGIT and Neural Nets easily capture and represent linear structure**
- **CART is notoriously weak at capturing strong linear structure**
  - **CART recognizes the structure but cannot represent it effectively**
  - **CART can produce a very large tree in an attempt to represent very simple relationships**
- **Neural Nets will capture complex structure but completely dependent on good variable selection**
- **Many non-linear structures can still be approximated with a linear structure, hence even incorrectly specified LOGIT can sometimes perform well**



# *Natural question: Can CART, Logit, and Neural Nets be combined?*

## CART

- **Automatic analysis**
- **Surrogates for missing values**
- **Unaffected by outliers**
- **Discontinuous response**
  - **Small change in  $x$  could lead to a large change in  $y$**
- **Coarse-grained**
  - **A 17-node tree can only predict 17 different probabilities**
- **High speed**

## LOGIT and Neural Nets

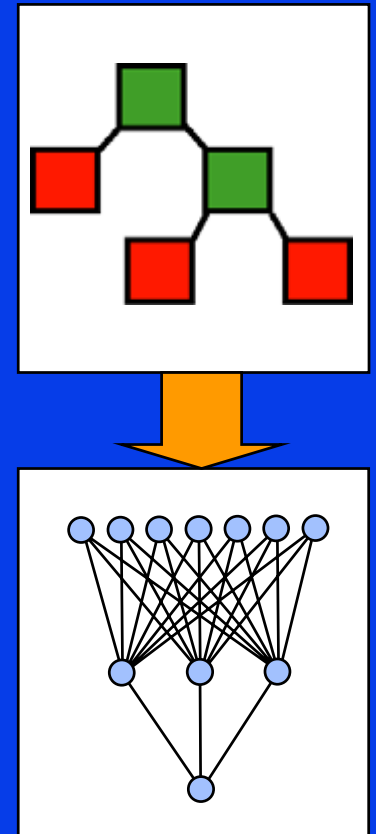
- **Requires experts to set up**
- **Deletes records or imputes missing values**
- **LOGIT Sensitive to outliers**
- **Continuous smooth response**
  - **Small change in  $x$  leads to a small change in  $y$**
- **Can have unique predicted probability for every record**
- **Low speed to infeasible if too many inputs**

# *A new approach: Use CART outputs as inputs to Logit and Neural Nets*

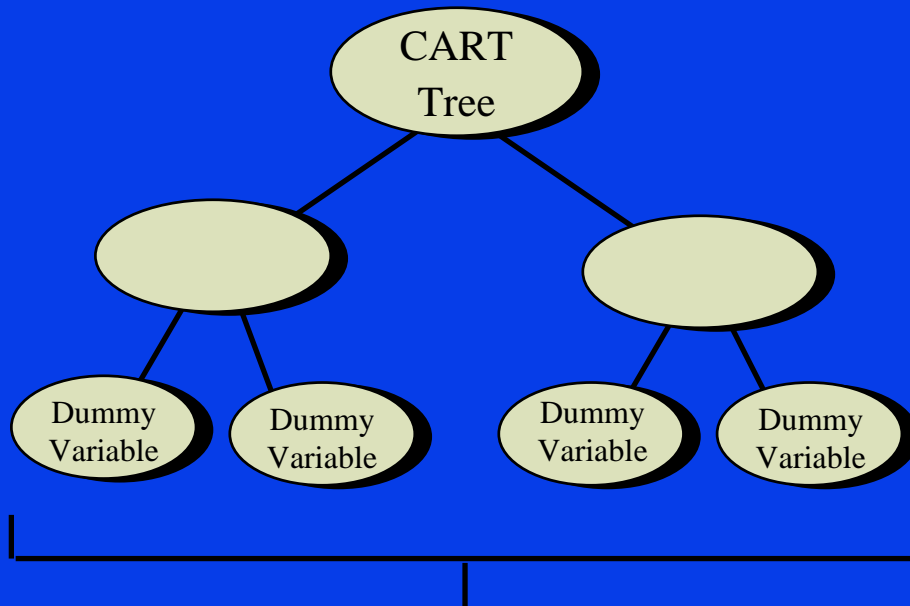
- **Let Logit and Neural Net start where CART stops - with the CART predictions**
- **Advantages:**
  - **CART gets a solution much faster**
  - **CART can work with dirty data**
  - **Gives a good idea of how accurately data can be modeled**
  - **CART Finds the database fields that are most promising**
  - **Missing values handled**
- **Let Logit and Neural Net attempt to improve on the CART tree by using all the information CART has already discovered**
  - **including which variables to use as inputs**
  - **would be very difficult to get to the CART outputs quickly if we began with a NN**

# *Types of CART output feeding NN*

- **CART outputs have two forms:**
  - probability of response (core output)
  - which node a record ends up in (as many outputs as there are nodes)
- **CART node indicators are far more flexible but use up more inputs**
- **CART also produces list of variables to use as additional inputs to NN**

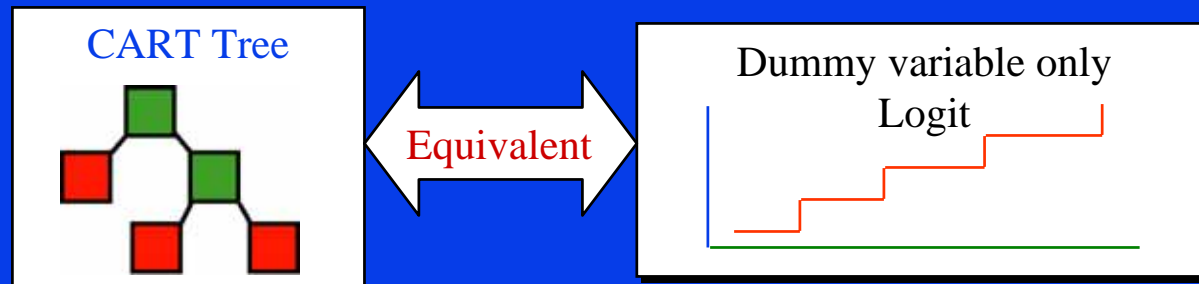


# *CART outputs for use in Hybrid: Terminal node indicators*



- **CART outputs are terminal node indicators**
- **Each associated with a probability of an outcome**
- **CART output could be the prediction (a single variable in y/n response)**
- **CART output could be complete set of indicator dummies (preferred)**

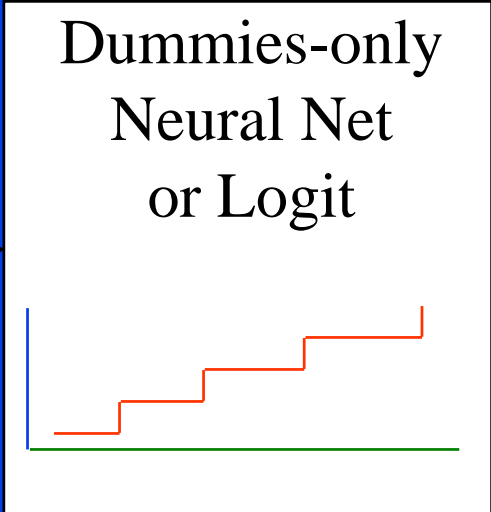
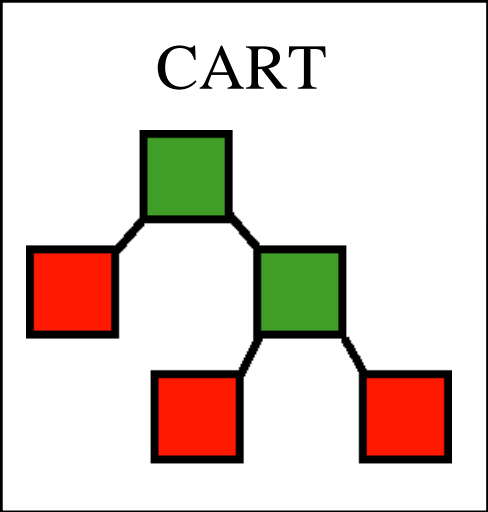
# *Logit or NN on terminal node dummies alone reproduces CART exactly*



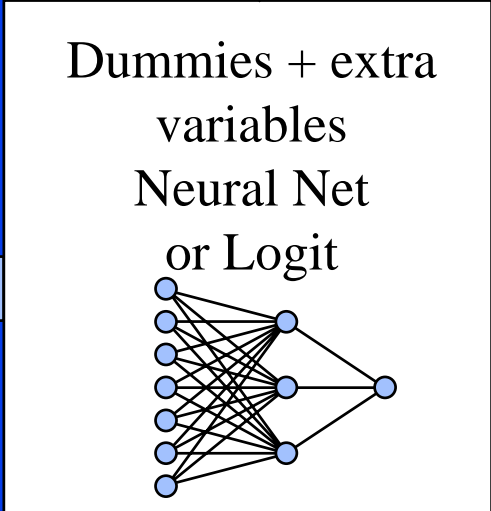
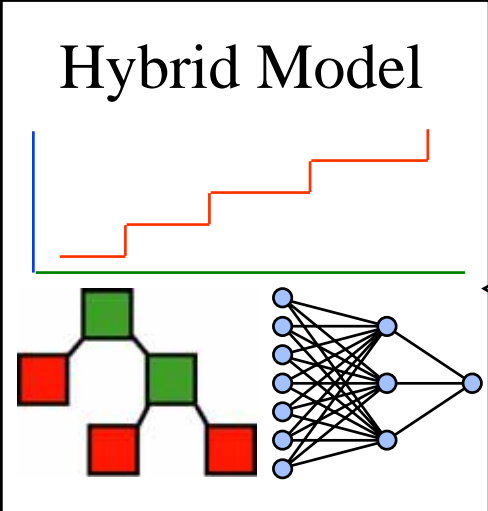
- **The logit model or NN fit to CART terminal node dummies converts the dummies into estimated probabilities**
- **is an exact representation of the CART model**
- **Each dummy represents the rules and interaction structure discovered by CART, albeit buried in a black box**
- **Excellent way to incorporate sampling weights**
- **Can also be used to recalibrate and update a CART tree**
- **Likelihood score for this model forms a baseline for further testing and model assessment**

## *Hybrid expands baseline by adding other inputs*

- **Add variables CART has identified as important predictors to the NN**
- **In recent article in PCAI Dwinnell used CART just for variable selection for an NN**
  - **obtained more accurate results in this two step method**
  - **NNs trained faster**
- **Our approach is more radical: use CART for variable selection AND feed outputs to NN**
- **Four real world examples to be reported below show improved results using this method**



Baseline  
for  
comparisons



# *Missing value handling in hybrid CART-Logit*

- **Simplest approach--ignore the problem!**
  - Drop all records with missing values on model variables
- **Assign CART-predicted probabilities to those cases**
- **Assign hybrid-predicted probabilities to all other cases**
- **More complicated approaches**
  - Missing value imputation
  - Dummy for missing value indicator plus nesting for non-missing
- **Complicated procedures not needed since CART tree will give good results anyway**

## *Hybrid CART-LOGIT:*

- **Variables added as main effects will capture effects common across all nodes**
- **Common effects are undetectable within terminal nodes because the signal to noise ratio within terminal nodes is too low**
- **Effects detected across terminal nodes are likely to be weak -- all strong effects already detected by CART**
- **Nevertheless, a collection of weak effects can be very significant**
- **Added variables can be tested as a group via likelihood ratio test**

# Logit formulas for CART and Hybrid models

- CART only**

$$y = \beta_0 + \beta_1 \text{NODE}_1 + \beta_2 \text{NODE}_2 + \dots + \beta_K \text{NODE}_K$$

Where  $\text{NODE}_i$  is a dummy variable for the  $i^{\text{th}}$  CART node.

**NOTE:** Every observation is assigned to a CART terminal node regardless of whether any predictor variables are missing.

- CART-Logit Hybrid**

$$y = \beta_0 + \beta_1 \text{NODE}_1 + \beta_2 \text{NODE}_2 + \dots + \beta_R \text{NODE}_R + \beta_{K+1} X_1 + \beta_{k+2} X_2 + \beta_{k+3} X_3 + \dots + \beta_R X_R$$

$$= \beta_0 + \sum_{i=1}^R \beta_i \text{NODE}_i + \sum_{R+1}^R \beta_i X_j$$

CART node dummies    Hybrid covariates

# *Does the Hybrid CART-Neural Net or CART-Logit work? Real World Results*

**These real-world examples all showed benefits-**

**Hybrid was always better than either component alone**

- **Direct Mail Response Model (Catalog)**
- **Direct Mail Response Model (Credit Card)**
- **Direct Mail Response Model (Insurance Product)**
- **Financial Markets: Mortgage Model**
- **Financial Markets: Loan delinquency**
- **Financial Markets: Predict consumer credit score threshold**
- **Retail Trade: Response to promotion**
- **Financial Markets: Fraud detection**

# *Catalog Sales Example*

- **Real world data mining project for major catalog retailer**
- **Goal was to improve targeting and obtain better response rates to mailing**
- **Complete history of all transactions for more than 5 years available for all previous purchasers**
- **Demographics, credit bureau, other spending data for both purchasers and non-purchasers**
- **Over 2,000 variables available for analysis**
- **Over 1 million names available for analysis**
- **Used CART, LOGIT, and Hybrid-CART-LOGIT models for targeting (modeled response (Y/N) only)**
- **Results were successful (goals met)**
  - **lift charts calculated for both raw response and \$ spent**

# ***HYBRID CART-NN Test Results***

- **Four real-world examples used (all Y/N response)**
  - All developed using training and test data to select best model
  - Models then compared on true holdout sample
- **Telecommunications: response to direct mail offer**
  - ◆ 87 attributes      150,000 training cases
- **Retail sales: response to promotion**
  - ◆ 6 attributes      300,000 training cases
- **Credit score: above or below threshold**
  - ◆ 21 attributes      300,000 training cases
- **Neural Nets run on dbProphet™ by Trajecta Inc of Austin TX**

# *CART-NN Hybrid Results*

## TELECOM

	TOP 23.19%	TOP 50.54%	TOP 77.18%	MAX KS
Pure CART	36.45	69.43	89.80	21.02
Pure dbProphet	37.70	70.47	91.38	22.62
CART/dbProphet	39.21	71.53	91.05	23.47

## RETAIL

	TOP 24.64%	TOP 50.13%	TOP 68.78%	MAX KS
Pure CART	34.81	62.60	77.94	18.32
Pure dbProphet	33.43	59.54	75.87	22.62
CART/dbProphet	33.76	60.99	76.00	23.47

## CREDIT

	TOP 19.04%	TOP 52.40%	TOP 72.69%	MAX KS
Pure CART	21.37	57.55	77.62	34.87
Pure dbProphet	21.54	57.74	77.93	36.66
CART/dbProphet	21.52	57.82	77.95	37.25

# *Why Does the Hybrid work?*

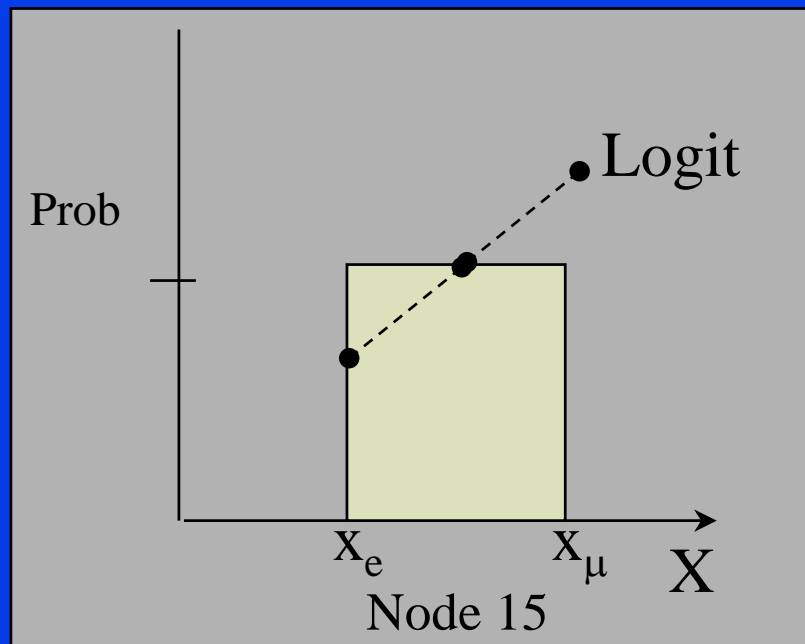
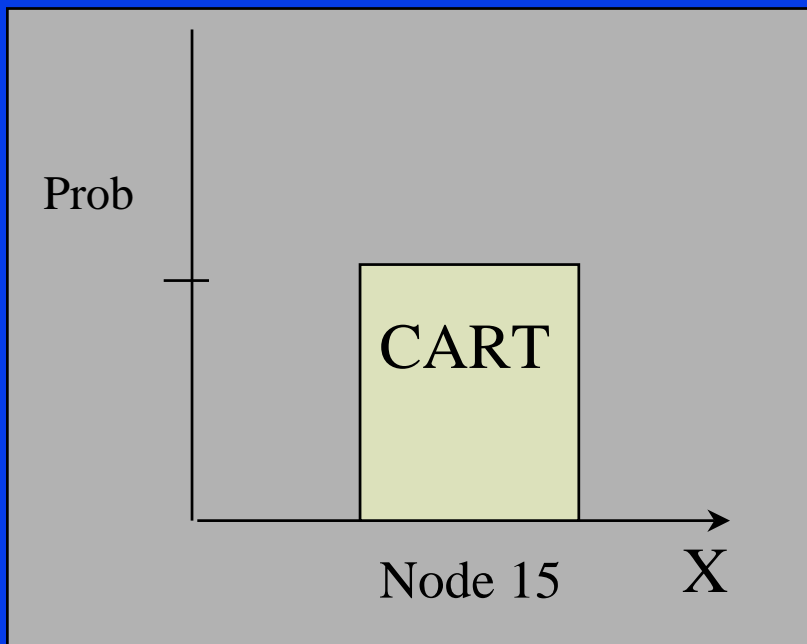
- **CART partitions data into relatively homogenous subgroups**
- **Each subgroup analyzed separately**
  - Ideal to detect segment structure
- **Pattern discovery becomes progressively more local**
- **Segments become smaller as CART tree grows**
- **Logit focus on entire data set**
  - Ideal to detect broad-scale effects and patterns
- **Neural Nets are a smooth fitting technique like Logit, but can adapt to local data structure like CART**
  - will detect cross-node commonalities

# *The Hybrid finds effects that are invisible to the decision tree*

- **By looking across nodes, Logit finds effects that CART cannot detect**
- **Because these effects are not terribly strong they are not picked up by CART as primary node splitters**
- **Once the sample is split by CART, these effects become progressively more difficult to detect as the subsamples become increasingly homogenous in the CART child nodes**
- **While these effects may not be the strongest individually, collectively they can add enormous predictive power to the model**
- **Result is a major enhancement in CART methodology**

# *How the Logit or NN improves CART*

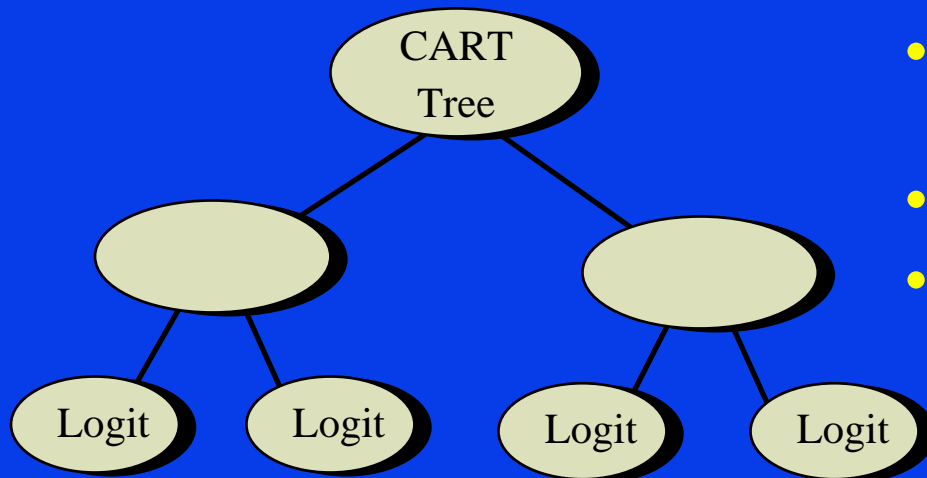
- CART assigns a single score (probability) to all cases arriving at a terminal node
- Logit and NN imposes a slope on the cases in the node, allowing smooth differentiation of within-node probabilities based on variables
- Note: slope possibly common across several or even all nodes



# *LOGIT and Neural Nets can also compensate for CART weaknesses*

- **CART sometimes produces a very coarse-grained response image**
  - Might produce only a small number of terminal nodes
  - “Score” is shared by all cases in a node
  - A 12 - terminal node tree produces only 12 distinct scores
- **Once CART finds a relatively rich group of responders it might stop splitting that node**
  - Identifying a large block of cases as responders could be a classification tree success
  - For targeting, we may want to distinguish the strongest responders from the merely strong responders (e.g., for targeting with a restricted budget, we might only mail to the strongest group)
  - For classification purposes, such a distinction is irrelevant
    - ♦ Both groups are responders so CART does not try to distinguish further

# *Previous Hybrid methods: Why they didn't work*



- **CART directed to grow a deliberately shallow tree**
- **Logit run in terminal nodes**
- **Results not successful**
  - **technique did not catch on**

**Technique of method combination did not play to the strengths of each method**

# *CART trees progressively truncate the variance of $x$ and $y$*

- **Goal of CART is to split data into homogeneous subsets**
  - the farther down the tree we go the less variability in the dependent variable
- **CART splits send cases with  $x \leq c$  to left and  $x > c$  to right**
  - Thus, the variance in predictor variables is also drastically reduced
  - If  $X$  is normally distributed and the cut point is at mean, variance in child nodes is reduced by about 64%; for subsequent mean splits the variance reduction will always be greater than 50%
  - Thus, two splits on a variable could reduce variance to less than 25% of full sample variance
    - ◆ reduction will also apply to other correlated predictors

# *Running Logit, NN, or any model in CART terminal nodes is hopeless*

- **By the time CART has declared a node terminal, the information remaining in the node is insufficient to support further statistical analysis**
- **The sample size is drastically reduced**
  - **The x and y variables have substantially reduced variance**
- **In a well-developed CART tree no parametric model should be supportable within terminal nodes**
- **Some analysts have suggested estimating logits or other parametric models earlier in the tree, after just a few splits**
  - **Same drawbacks as terminal node models but less extreme**
  - **At best provides a mechanism for finding switching regressions-- not terribly successful in practice**

# *Key to successful hybrid: Run Logit or Neural Net in the root node*

- **Need to run the Logit/Neural Net on all the data, thereby capitalizing on strength in detecting global structure**
- **Implemented as follows:**
  - **Run CART, assigning every case to a terminal node**
    - Assignment possible even for cases with many missing values
    - Even a case with all missing data can be assigned a terminal node
  - **Terminal node assignment reported by categorical variable with as many levels as terminal nodes**
  - **Feed this categorical variable in the form of terminal node dummies to a logit model**

# *Does the CART-Logit Hybrid work?*

## *Formal Experiments*

- **Real-world examples are valuable case studies but constitute a small sample**
- **We conducted extensive experiments on artificial data sets**
- **Each experiment run 100 times with different random number seeds**
- **Allows a more accurate assessment of possible benefits and flaws of the hybrid methodology**
- **Results show the hybrid method does work**
- **For speed reasons only CART and Logit run**

# *Monte Carlo model assessment*

- **Assume known probabilistic data generation processes**
  - the “truth”; try several different versions of truth
- **Draw random samples of various sizes (2,000 - 100,000)**
  - Repeat random experiment many times (100 replications)
- **Repetition for each scenario of what is truth**
- **Assess models on basis of**
  - Fit to data
  - Performance -- e.g. profit yielded if model guides policy
- **Look at both training data and holdout samples**
- **Comparison of training and holdout data can be used to measure overfitting**
  - The more overfit, the better performance will be on training data and the worse on Holdout data.

# *Specific Monte Carlo experiments*

- **True data-generating process is**
  - **Simple Logit- one variable**
  - **CART process- one variable, highly non-linear**
  - **Hybrid process**
  - **Logit- several variables (Possibly missing)**
  - **Hybrid- several variables (Possibly missing)**
  - **Highly non-linear smooth function (not Logit)**
  - **Complex Logit with informative missingness**

# *Summary of results*

- **In smaller samples, Logit performs well even when Logit is not the true model**
  - **Simpler model reduces risk of over-fitting**
- **In larger samples, Hybrid model dominates out-of-sample performance**
  - **Even when Logit is true model, Hybrid is almost as good**
  - **Hybrid models based on mediocre CART trees perform only slightly worse than hybrid models based on the best available CART tree**
- **In large datasets with high frequencies of missings, Hybrid outperforms other models regardless of which model is true**
- **CART and Hybrid manage problems with missing values quite well whereas logit performance collapses**
- **CART tree grown on one-half the training sample (LEARN data), the other half (TEST data) used to select the optimal tree**
- **Hybrid and CART overfit on LEARN data but not TEST data**
- **The CART TEST sample can be used to get measures of out-of-sample performance**

# *Log-likelihood results N=10,000*

Learn/Test Sample				
Experiment	CART	Logit	Hybrid	Truth
1	-12384.8	-11980	-11972.2	-11981
2	-12671.4	-13861.8	-12670.9	-12658.7
3	-11931.5	-11589.3	-11563.5	-11543.4
4	-12025.6	-11633.4	-11497.4	-10789.1
4 w/dummies	-11979.6	-11633.4	-11484.5	-10789.1
5	-11822.1	-11548.4	-11274.8	-10277.8
5 w/dummies	-11728	-11548.4	-11230.8	-10277.8
6	-13127.9	-13788.5	-13085.9	-12267.8
6 w/dummies	-13069.2	-13788.7	-13046.3	-12268
7	-11868.7	-11507.5	-11283.5	-10374.7
7 w/dummies	-11625.4	-11507.5	-11217	-10374.7

Holdout Sample				
Experiment	CART	Logit	Hybrid	Truth
1	-12431.4	-11999.2	-12010.6	-11997.9
2	-12710.1	-13864.2	-12710.5	-12659.9
3	-12003.1	-11585.8	-11594.8	-11537.6
4	-12259.2	-11645.1	-11648.5	-10794.8
4 w/dummies	-12238.8	-11645.1	-11654.1	-10794.8
5	-12028.4	-11547	-11417.5	-10280.6
5 w/dummies	-11970.3	-11547	-11401.1	-10280.6
6	-13388.9	-13796.2	-13357.4	-12251.5
6 w/dummies	-13368.6	-13796.5	-13350.3	-12251.8
7	-12114.7	-11505.6	-11461.6	-10370.6
7 w/dummies	-11792.5	-11505.6	-11400	-10370.6

# Performance on the holdout sample 20,000 observations, 100 replications

Experiment	CART	Logit	Hybrid	Truth
1	0.198	0.239	0.2386	0.239
2	0.1791	0.0003	0.1791	0.1819
3	0.2357	0.2619	0.2612	0.263
4	0.2241	0.2592	0.2601	0.302
4 w/dummies	0.226	same	0.26	same
5	0.2282	0.2536	0.2621	0.3114
5 w/dummies	0.232	same	0.2635	same
6	0.1114	0.0285	0.1209	0.2069
6 w/dummies	0.1167	same	0.1217	same
7	0.2152	0.247	0.251	0.2966
7 w/dummies	0.2323	same	0.2541	same

- **Table 12**
  - **Actual gains**

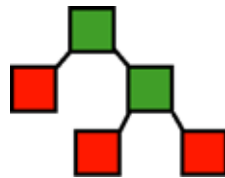
Experiment	CART	Logit	Hybrid	Truth
1	0.1987	0.2395	0.2391	0.2395
2	0.1793	0.0004	0.1794	0.1818
3	0.2354	0.2614	0.2608	0.2626
4	0.2243	0.2589	0.2599	0.3019
4 w/dummies	0.2265	same	0.2601	same
5	0.2282	0.2533	0.2622	0.3113
5 w/dummies	0.2321	same	0.2635	same
6	0.1118	0.0296	0.1213	0.2068
6 w/dummies	0.1171	same	0.1223	same
7	0.2159	0.2471	0.2514	0.2968
7 w/dummies	0.2316	same	0.2544	same

- **Table 13**
  - **Expected gains**

# *Do we need hybrid?*

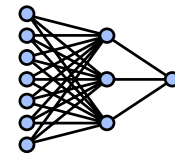
## CART

- **Fast**
- **Robust**
- **Easy to use**



## Neural Nets

- **Slow**
- **Prefer clean data**
- **Need expert guidance**



- **Can always start with CART and learn quickly**
- **Knowledge discovered with CART can be used to improve Neural Nets**
- **CART model applies to all records even with many missing data fields**
- **Hybrid allows for easy updates (as do NNs)**

# *References*

- **Breiman, L., J. Friedman, and R. Olshen, and C. Stone (1994), *Classification and Regression Trees*, Pacific Grove: Wadsworth.**
- **Friedman, J. H. (1991a), *Multivariate Adaptive Regression Splines (with discussion)*, *Annals of Statistics*, 19, 1-141 (March).**
- **Michie, D., D. J. Spiegelhalter, and C. C. Taylor, eds (1994), *Machine Learning, Neural and Statistical Classification*, London: Ellis Horwood Ltd.**
- **Steinberg, D. and Colla, P. L., (1995), *CART: Tree-Structured Nonparametric Data Analysis*, San Diego, CA: Salford Systems.**