Deloitte.

CASUALTY ACTUARIAL SOCIETY CAS 1914 ®

Non-Linear Modeling Techniques

CART, MARS, Neural Nets

James Guszcza, FCAS, MAAA

CAS Predictive Modeling Seminar
Boston
October, 2006

Audit.Tax.Consulting.Financial Advisory.

Topics

Overview

Classification and Regression Trees

Multivariate Adaptive Regression Splines

Neural Networks

Deloitte.

Overview

Definitions

Overview of Techniques

Audit.Tax.Consulting.Financial Advisory.

Semantics: Data Mining vs Predictive Modeling

- Data Mining
  - KDD: Knowledge Discovery in Databases
  - EDA: Exploratory Data Analysis
  - Open-ended
  - "cast the net wide"
  - "Let the data speak for itself"
  - CART, MARS, stepwise procedures, unsupervised procedures,…

- Predictive Modeling
  - Build a model tailored to achieve a pre-specified goal
  - Build on:
    - Results of data mining
    - **Domain expertise! (actuarial & insurance knowledge)**

Actuarial science needs data mining…

… but data mining *also* needs actuarial science

Philosophy of the Inventors of CART

"Our philosophy in data analysis is to look at the data from a number of different viewpoints. Tree structured regression offers an interesting alternative for looking at regression type problems. It has sometimes given clues to data structure not apparent from a linear regression analysis. Like any tool, its greatest benefit lies in its intelligent and sensible application."

"Binary Trees give an interesting and often illuminating way of looking at the data in classification or regression problems. They should not be used to the exclusion of other methods. We do not claim that they are always better. They do add a flexible nonparametric tool to the data analyst's arsenal."

--Breiman, Friedman, Olshen, Stone

Classification and Regression Trees (1984)

Some Definitions

- Target Variable Y
  - What we are trying to predict.
    - Profitability (loss ratio, LTV), Retention, ...

- Predictive Variables $\{X_1, X_2, \ldots, X_N\}$
  - "Covariates" used to make predictions.
    - Policy Age, Credit, #vehicles….

- Predictive Model $Y = f(X_1, X_2, \ldots, X_N)$

- Common model form: $Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_N X_N$

- We will explore 3 alternate model forms today.
    - CART
    - MARS
    - Neural Nets

Supervised vs Unsupervised Learning

- Supervised Learning
  - Attempt to predict *Y* in terms of ***X***.
    - Ordinary Least Squares (OLS) regression
    - Generalized Linear Models (GLM)
    - Classification and Regression Trees (CART)
    - Multivariate Adaptive Regression Splines (MARS)
    - Neural Nets
    - Generalized Additive Models (GAM)
    - Support Vector Machines
    - ...

- Unsupervised Learning
  - Attempt to find interesting patterns amongst the ***X***.
  - No outcome ("target") variable
    - Clustering
    - Principal Components / Factor Analysis
    - ...

Classification vs Regression

- Classification Problems
  - Goal is to segment the observations of interest into 2 or more distinct categories.
    - Average, above average, below average profitability customers
    - Likely fraudsters vs likely non-fraudsters
    - Likely defectors vs likely non-defectors

- Regression Problems
  - Goal is to predict a continuous amount.
    - Dollars of loss for a policy
    - Ultimate size of claim
    - Years of retention
    - Days out of work

Parametric vs Non-Parametric

- Parametric Statistics
  - Involves the specification of a probabilistic model governing the process that generated the data.
    - e.g. Poisson Regression: assume that the process generating the number of claims for a given policy is a Poisson process.
    - The mean (“λ”) of each policy’s Poisson distribution is a linear combination of various attributes.

- Non-Parametric Statistics
  - No probability model specified
  - Purely “algebraic”
    - e.g. classification trees use brute-force computing to create relatively “pure” segments of a dataset.
    - No assumptions made about the target variable.

Deloitte.

CART

Classification And Regression Trees

Audit.Tax.Consulting.Financial Advisory.

The Key Idea: Recursive Partitioning

- Take all of your data.
- Consider *all* possible values of *all* variables.
- Select the variable/value **(*X*=*t_1*)** that produces the greatest "separation" in the target.
  - **(*X*=*t_1*)** is called a "split".
- If $X < t_1$ then send the data to the "left"; otherwise, send data point to the "right".
- Now repeat same process on these two "nodes".
  - You get a "tree"
  - Note: CART only uses *binary* splits.

Simple Insurance Example

- Suppose you have 3 variables:

# vehicles:	$\{1,2,3...10^+\}$
Age category:	$\{1,2,3...6\}$
Liability-only:	$\{0,1\}$

- At each iteration, CART tests all 15 splits.

  (#veh<2), (#veh<3),..., (#veh<10)

  (age<2),..., (age<6)

  (lia<1)

Select split resulting in greatest increase in *purity*.

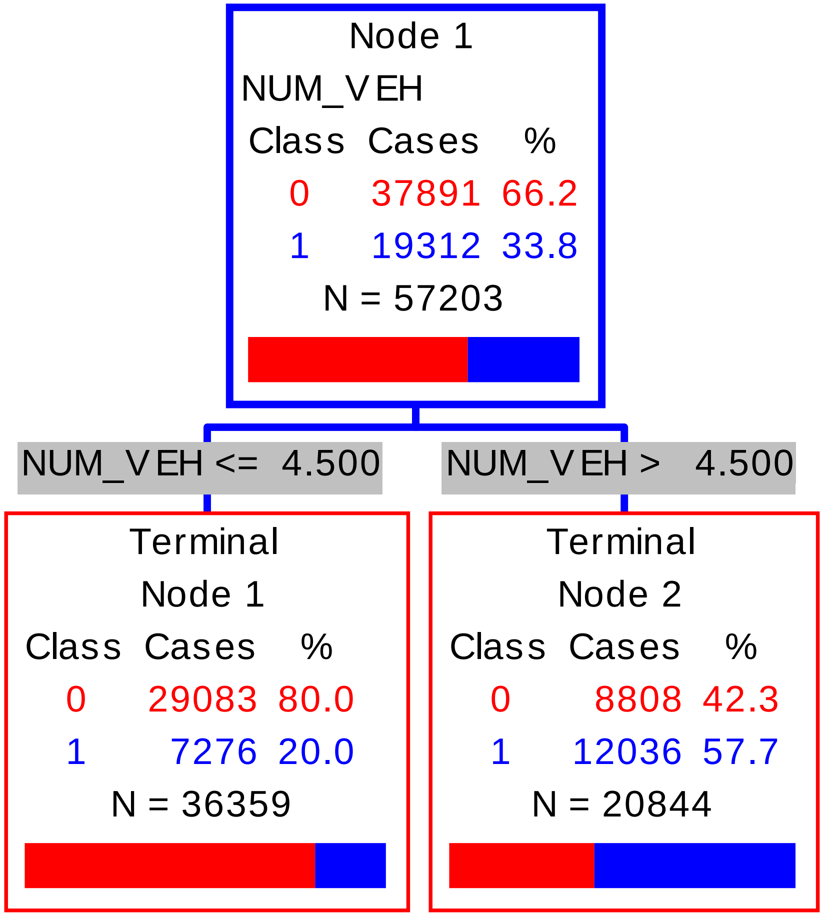
- Perfect purity: each split has either all claims or all no-claims.
- Perfect impurity: each split has same proportion of claims as overall population.

Predict Likelihood of a Claim

- Commercial Auto Dataset
  - 57,000 policies
  - **34%** claim frequency

- Classification Tree using Gini splitting rule

- First split:
  - Policies with ≥5 vehicles have **58%** claim frequency
  - Else **20%**
  - Big increase in purity



Node 1
NUM_VEH

Class	Cases	%
0	37891	66.2
1	19312	33.8

N = 57203

NUM_VEH <= 4.500

Terminal
Node 1

Class	Cases	%
0	29083	80.0
1	7276	20.0

N = 36359

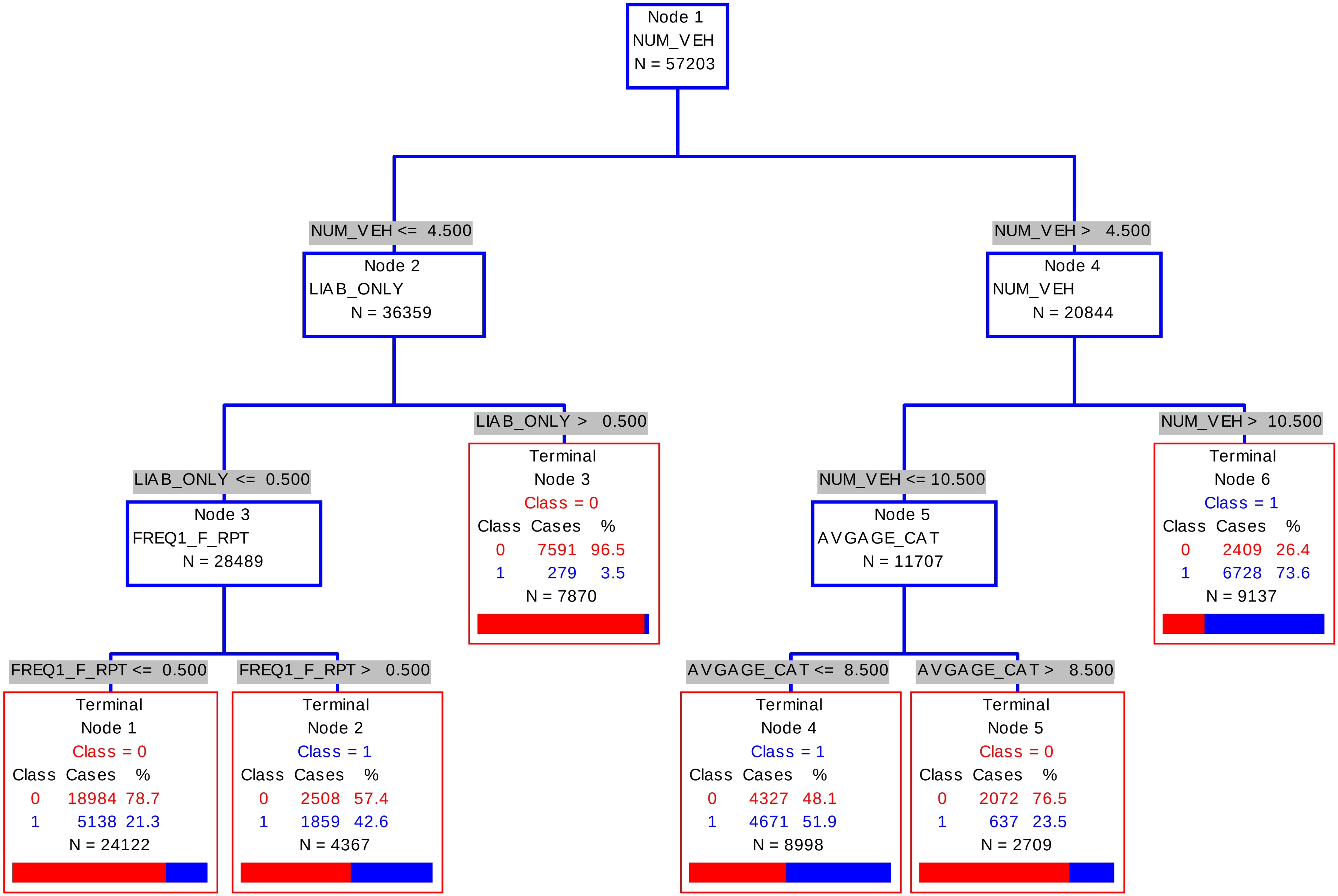
NUM_VEH > 4.500

Terminal
Node 2

Class	Cases	%
0	8808	42.3
1	12036	57.7

N = 20844

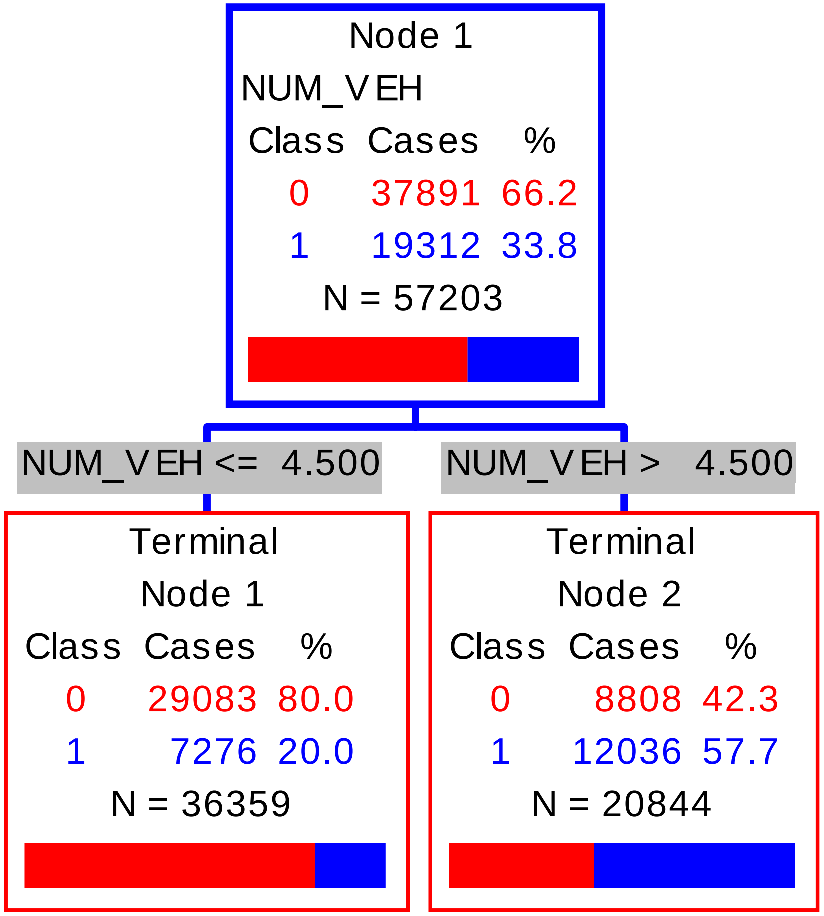
Growing The Tree



Observations (Shaking the Tree)

- First split (# vehicles) is rather obvious
  - More exposure ➔ more claims

- But it confirms that CART is doing something reasonable.
  - Also: the choice of splitting value 5 (not 4 or 6) is non-obvious.
  - This suggests a way of optimally “binning” continuous variables into a small number of groups.



Node 1
NUM_VEH

Class	Cases	%
0	37891	66.2
1	19312	33.8

N = 57203

NUM_VEH <= 4.500

Terminal Node 1

Class	Cases	%
0	29083	80.0
1	7276	20.0

N = 36359

NUM_VEH > 4.500

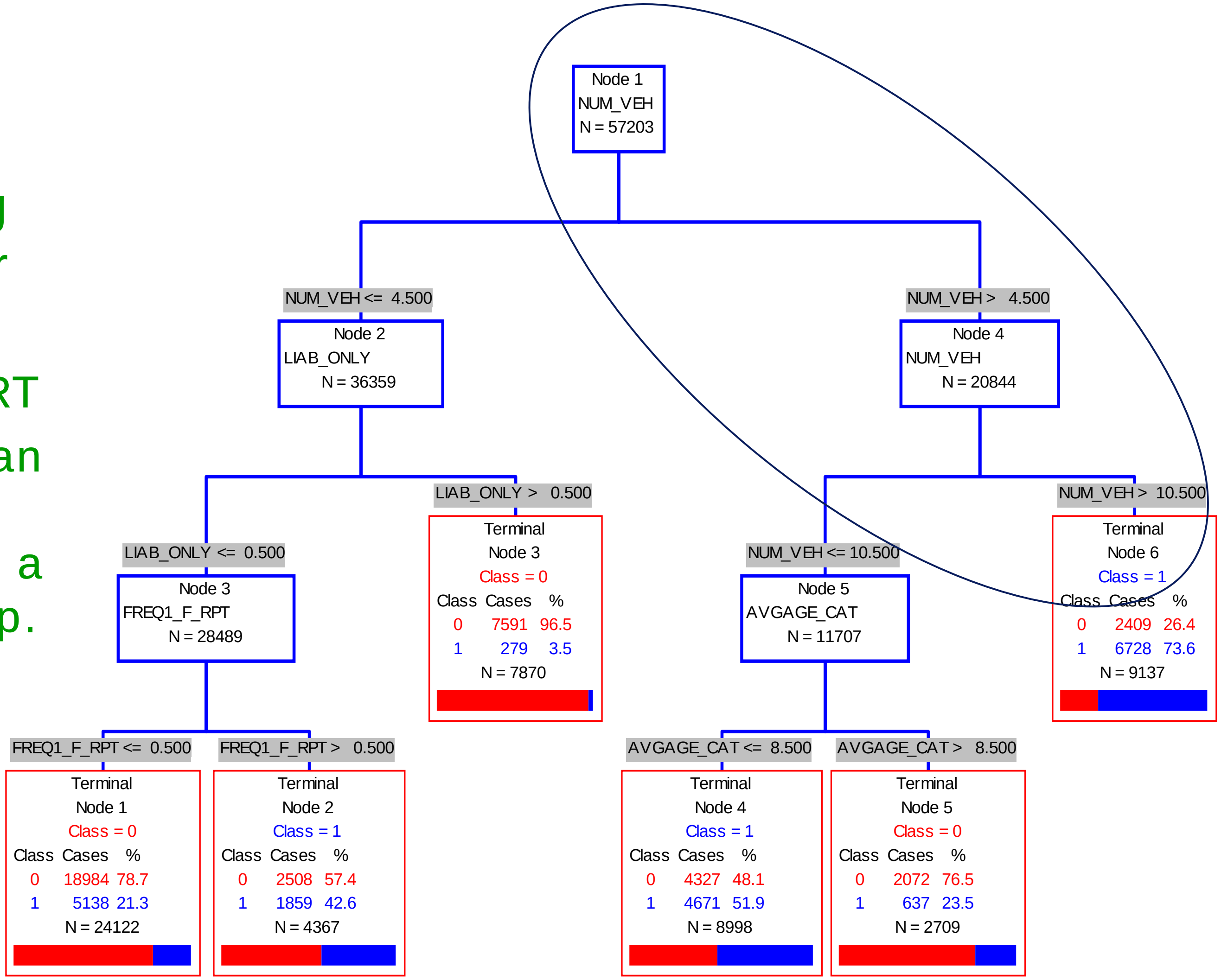
Terminal Node 2

Class	Cases	%
0	8808	42.3
1	12036	57.7

N = 20844

CART and Linear Structure

- Notice Right-hand side of the tree...
- CART is struggling to capture a linear relationship
  - Weakness of CART
  - The best CART can do is a step approximation of a linear relationship.



Node 1
NUM_VEH
N = 57203

NUM_VEH <= 4.500

Node 2
LIAB_ONLY
N = 36359

NUM_VEH > 4.500

Node 4
NUM_VEH
N = 20844

LIAB_ONLY <= 0.500

Node 3
FREQ1_F_RPT
N = 28489

LIAB_ONLY > 0.500

Terminal Node 3
Class = 0

Class	Cases	%
0	7591	96.5
1	279	3.5

N = 7870

NUM_VEH <= 10.500

Node 5
AVGAGE_CAT
N = 11707

NUM_VEH > 10.500

Terminal Node 6
Class = 1

Class	Cases	%
0	2409	26.4
1	6728	73.6

N = 9137

FREQ1_F_RPT <= 0.500

Terminal Node 1
Class = 0

Class	Cases	%
0	18984	78.7
1	5138	21.3

N = 24122

FREQ1_F_RPT > 0.500

Terminal Node 2
Class = 1

Class	Cases	%
0	2508	57.4
1	1859	42.6

N = 4367

AVGAGE_CAT <= 8.500

Terminal Node 4
Class = 1

Class	Cases	%
0	4327	48.1
1	4671	51.9

N = 8998

AVGAGE_CAT > 8.500

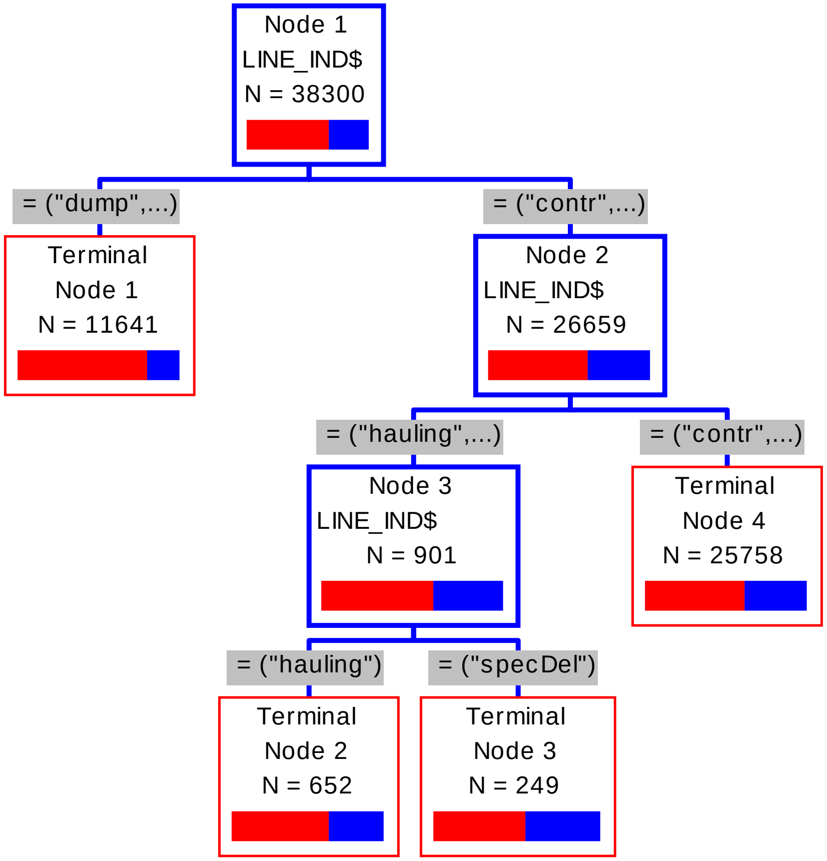
Terminal Node 5
Class = 0

Class	Cases	%
0	2072	76.5
1	637	23.5

N = 2709

High-Dimensional Predictors

- Categorical predictors: CART considers every possible subset of categories
  - Nice feature
  - Very handy way to group massively categorical predictors into a small # of groups



- Left (fewer claims): dump, farm, no truck

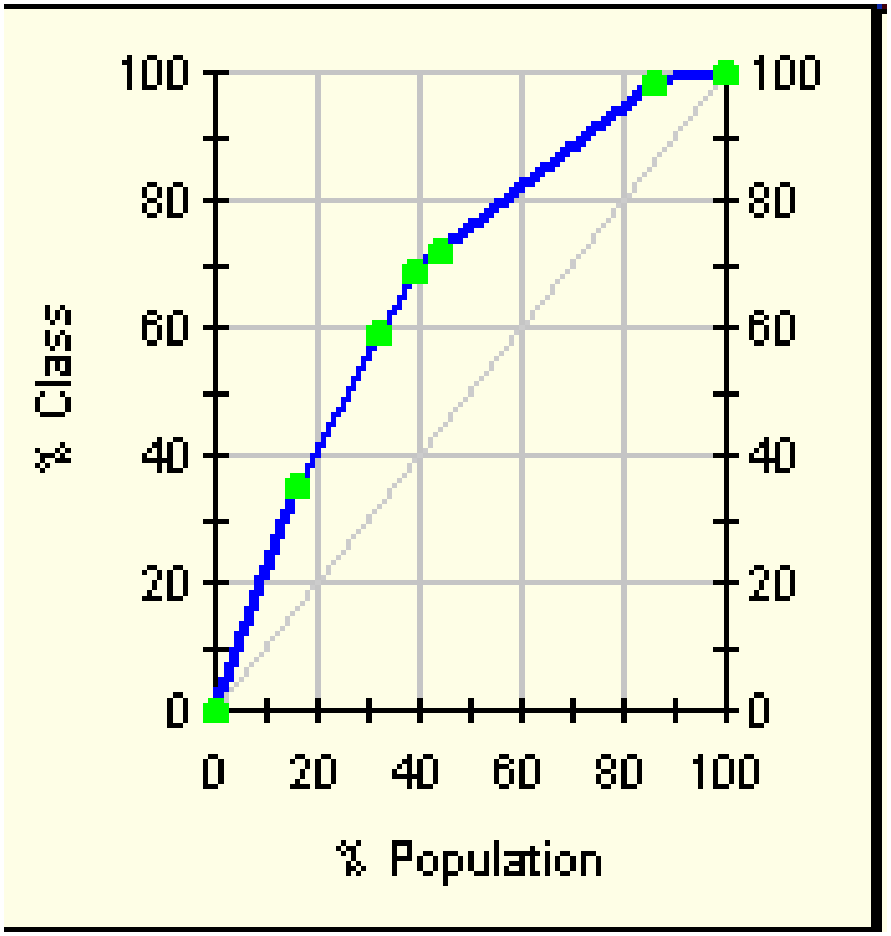
- Right (more claims): contractor, hauling, food delivery, special delivery, waste, other

Gains Chart: Measuring Success

<u>From left to right</u>:

- Node 6: 16% of policies, 35% of claims.
- Node 4: add'l 16% of policies, 24% of claims.
- Node 2: add'l 8% of policies, 10% of claims.
- ..etc.

- The steeper the gains chart, the stronger the model.
  - Analogous to a lift curve.
  - Desirable to use out-of-sample data.



Regression Trees

- Tree-based modeling for ***continuous* target variable.**
    - most intuitively appropriate method for loss ratio analysis
- Find split that produces greatest separation in

$$\sum[\mathbf{y} - \mathbf{E}(\mathbf{y})]^2$$

- i.e.: find nodes with minimal *within variance.*
    - and therefore greatest *between variance*
    - like credibility theory
- Every record in a node is assigned the same yhat.

    ➔ model is a *step function*

Classification Trees

- Tree-based modeling for ***discrete* target variable.**
- In contrast with regression trees, various measures of *purity* are used.
- Common measures of purity:
  - Gini: **$p(1-p)$**
  - Entropy: **$-\Sigma p \log p$**
  - Max entropy/Gini when p=.5
  - Min entropy/Gini when p=0 or 1
- Intuition: an ideal retention model would produce nodes that contain either defectors only or non-defectors only.
  - completely pure nodes

Cross-Validation and the Optimal Tree

- Essential to CART is a procedure for finding the optimal sized tree.
- CART does *not* use “stopping rules” to determine when to stop splitting the data into finer and finer nodes of the tree.
- Rather CART builds the tree all the way out; then systematically prunes the tree back to an optimal size.
- Cross-validation (error analysis on out-of-sample data) is used to determine the optimal tree size.
- Analogous idea used in MARS.

CART advantages

- Nonparametric (no probabilistic assumptions)
- Automatically performs variable selection
- Uses any combination of continuous/discrete variables
  - Very nice feature: ability to automatically bin massively categorical variables into a few categories.
    - zip code, business class, make/model…
- Discovers “interactions” among variables
  - Good for “rules” search
  - Hybrid GLM-CART models

CART advantages

- CART handles missing values automatically
  - Using “surrogate splits”
- Invariant to monotonic transformations of predictive variable
- Not sensitive to outliers in *predictive* variables
  - Unlike regression
- Great way to explore, visualize data

CART Disadvantages

- The model is a *step function*, not a continuous score
    - So if a tree has 10 nodes, $y^{\wedge}$ can only take on 10 possible values.
- Might take a large tree to get good lift
    - But then hard to interpret
    - Data gets chopped thinner at each split
- Instability of model structure
    - Correlated variables ➔ random data fluctuations could result in entirely different trees.
- CART does a poor job of modeling *linear structure*
    - MARS will be an improvement on this front

Uses of CART

- Building predictive models
  - Alternative to GLMs, neural nets, etc
  - Hybrid models

- Exploratory Data Analysis
  - A different view of the data (tree-structured, not linear etc).
  - You can build a tree on nearly any data set with minimal data preparation.
  - Which variables are selected first?
  - Interactions among variables
  - Take note of cases where CART keeps re-splitting the same variable (suggests linear relationship)

- Variable Selection
  - CART can rank variables
  - Alternative to stepwise regression

Deloitte.

MARS

Multivariate Adaptive Regression Splines

Audit.Tax.Consulting.Financial Advisory.

MARS – Basic Concepts

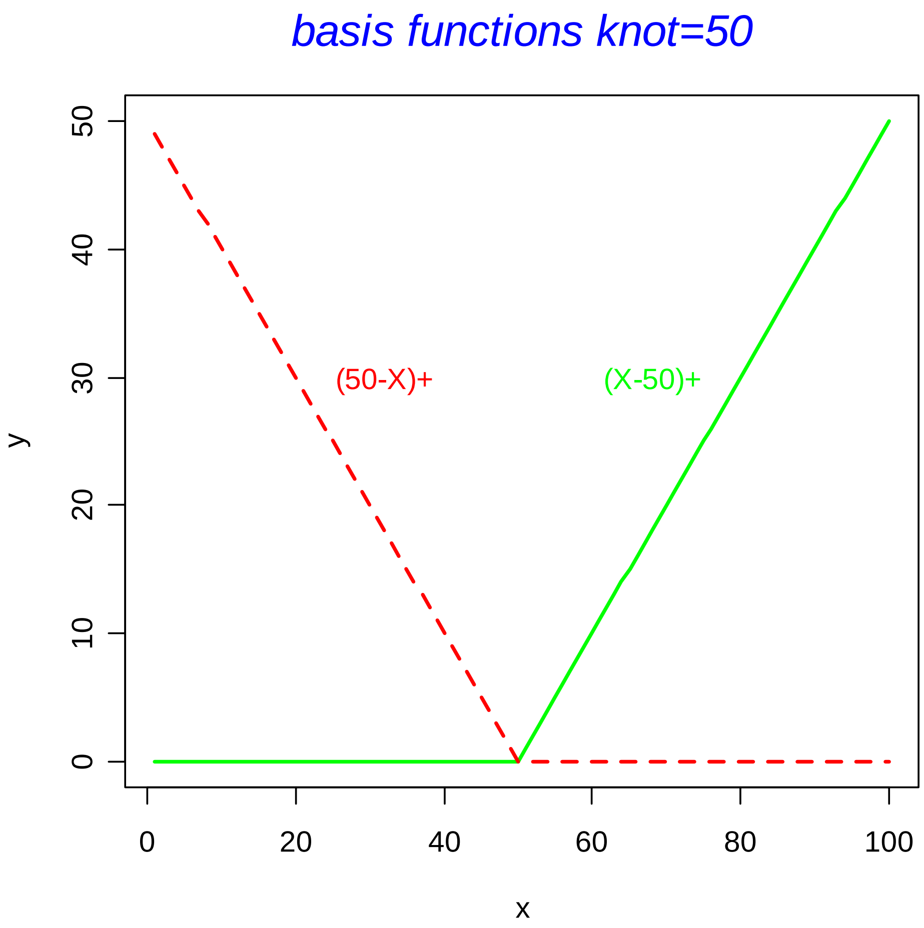
- Mars can be viewed as a generalized stepwise regression routine.
- First transform variables via “basis functions”.
- Next perform stepwise procedure on the transformed variables.
- One can also include interactions between the transformed variables in the stepwise procedure.
- Build the model out
- Prune back using cross-validation
    - (similar idea used in CART)

Basis Functions

- For each predictive variable, MARS considers a series of “basis functions”.

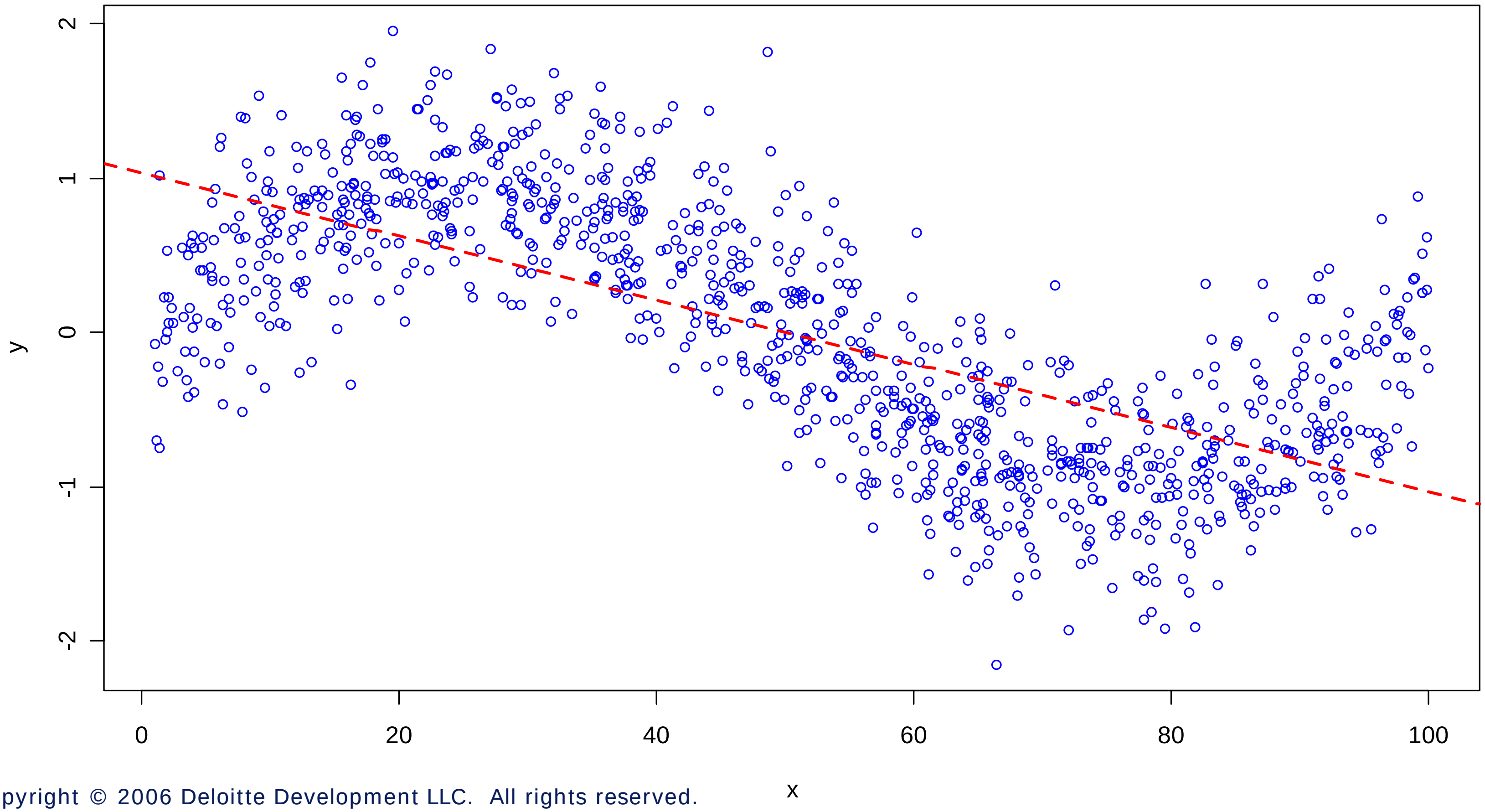
- For each value *x* of a variable *X*, we create two basis functions anchored at *x*.
    - “hockey stick functions”
    - E.g. if there are 100 possible values we create 200 basis functions

- Strategically selecting basis functions allows us to model non-linear relationships.



Simple Example: Fitting a Non-Linear Pattern

- In this example, *y* is a non-linear function of *x*.
- The regression line (red) fails to fit the pattern.
- Let's see what MARS does.

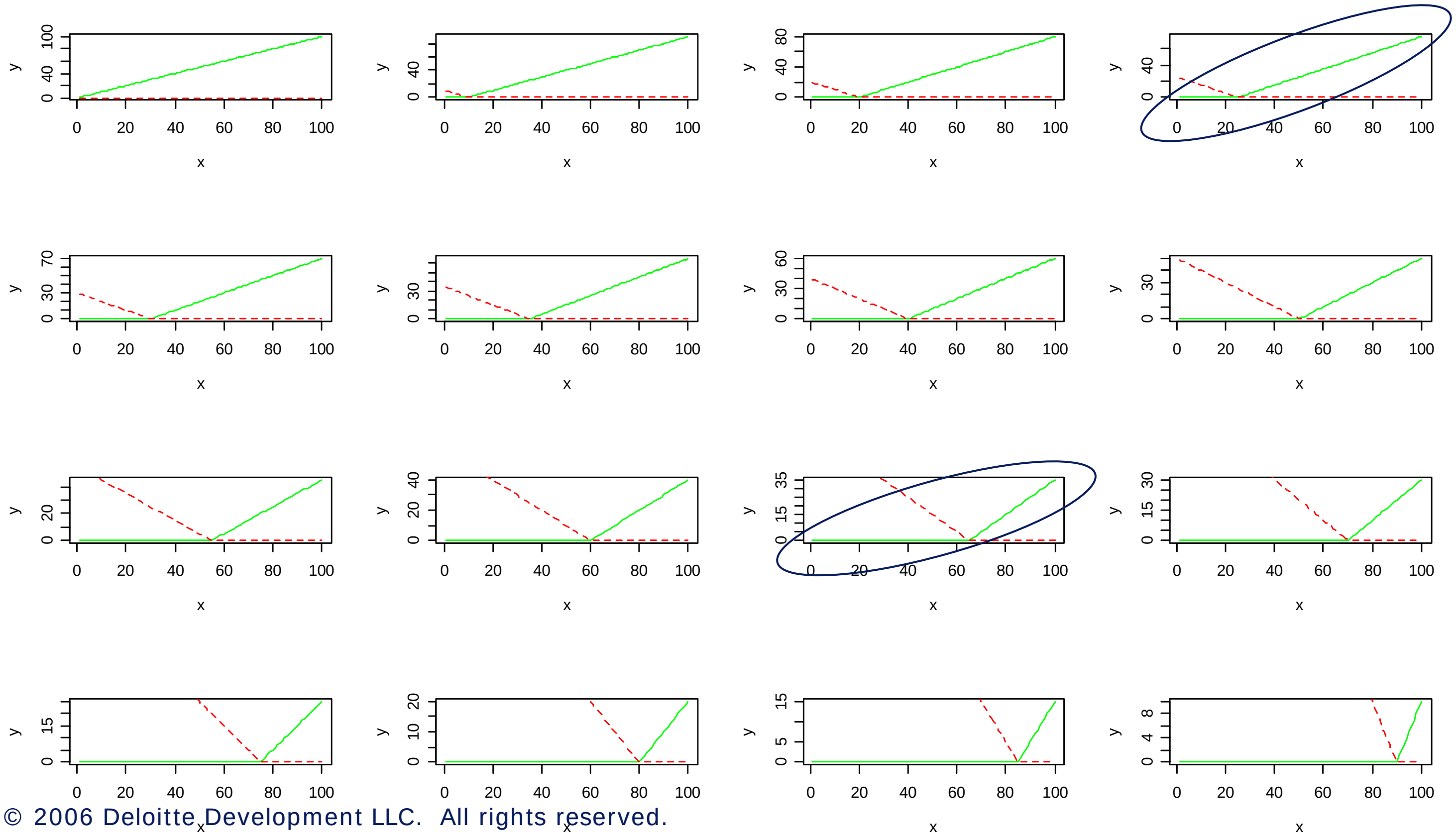


Simple Example: Fitting a Non-Linear Pattern

1. MARS generates a series of 200 basis functions.

   $(X\text{-}k)_+$ and $(k\text{-}X)_+$ for k ε {1,2,...,100}

2. Does a stepwise search to choose significant basis functions.
3. “Prunes back” the set of selected basis function to produce the optimal model.



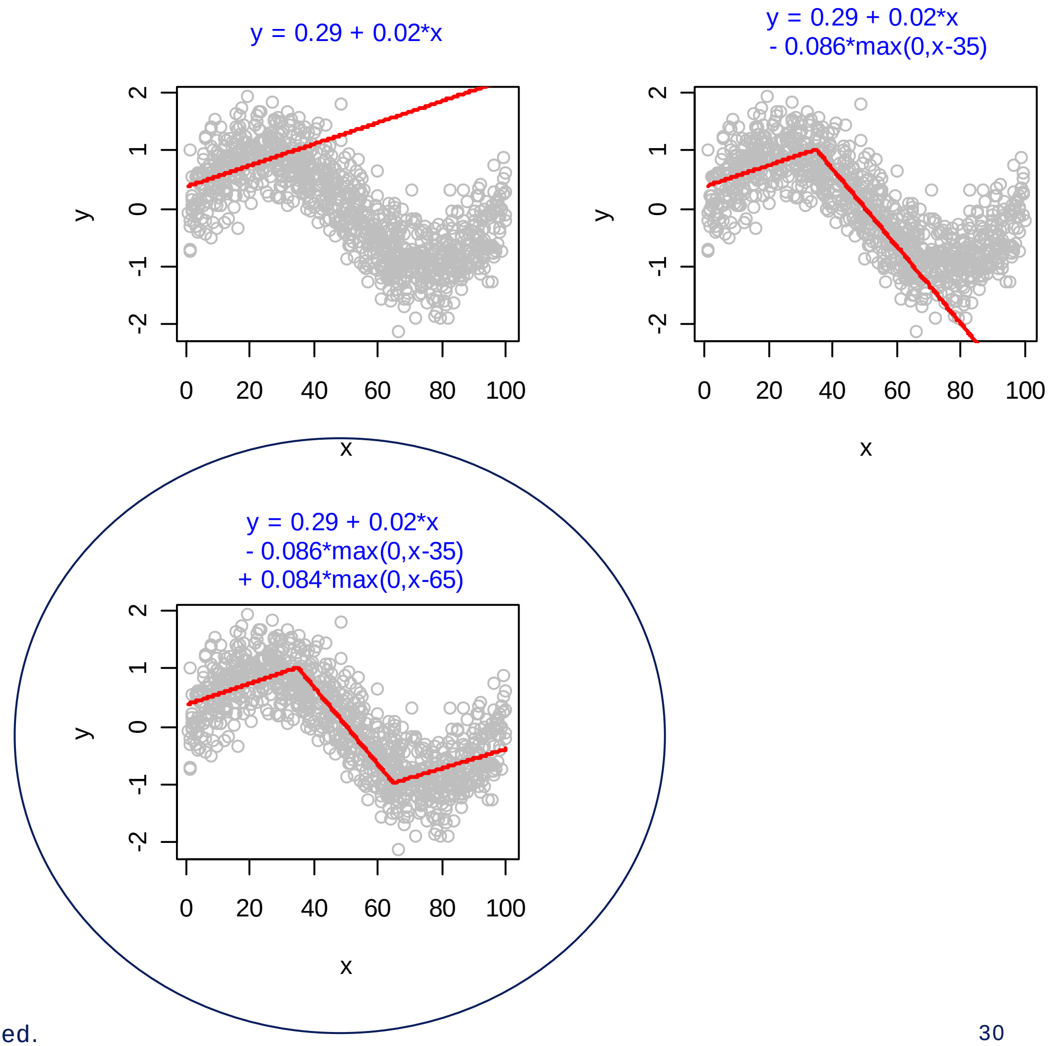
MARS Result

- After the forward stepwise search and pruning back the final MARS model is:

- $\hat{y} = 0.29 + 0.02*x$
  $- \; 0.086*\max(0,x-35)$
  $+ \; 0.084*\max(0,x-65)$

- $R^2_{reg} \;\; = 0.69$
- $R^2_{MARS} = 0.85$

- Only one candidate predictive variable (*x*), so no interaction term search in this case.



Multivariate MARS

- Suppose that at stage *n* of our forward selection procedure, the model has the form:

- $y^{\wedge} = \beta_0 h_0 + \beta_1 h_1 + \beta_2 h_2 + \ldots + \beta_p h_p$
  - Where h_i is a product of one or more basis functions
  - e.g.: $h_3=(X_2-43)_+ * (87-X_5)_+$
  - or: $h_5=(X_1-70)_+$
  - or: $h_0=1$

- At stage *n*+1, MARS considers adding terms of the form:

$$h_m*(X_\mathrm{j}-k)_+ \quad ; \quad h_m*(k-X_\mathrm{j})_+$$

  - For all candidate X_j and knots *k*
  - Note that h_0 is the identity function: $h_0=1$
  - Therefore the “interaction” with h_0 is in fact a stand-alone term

Interaction Terms in MARS

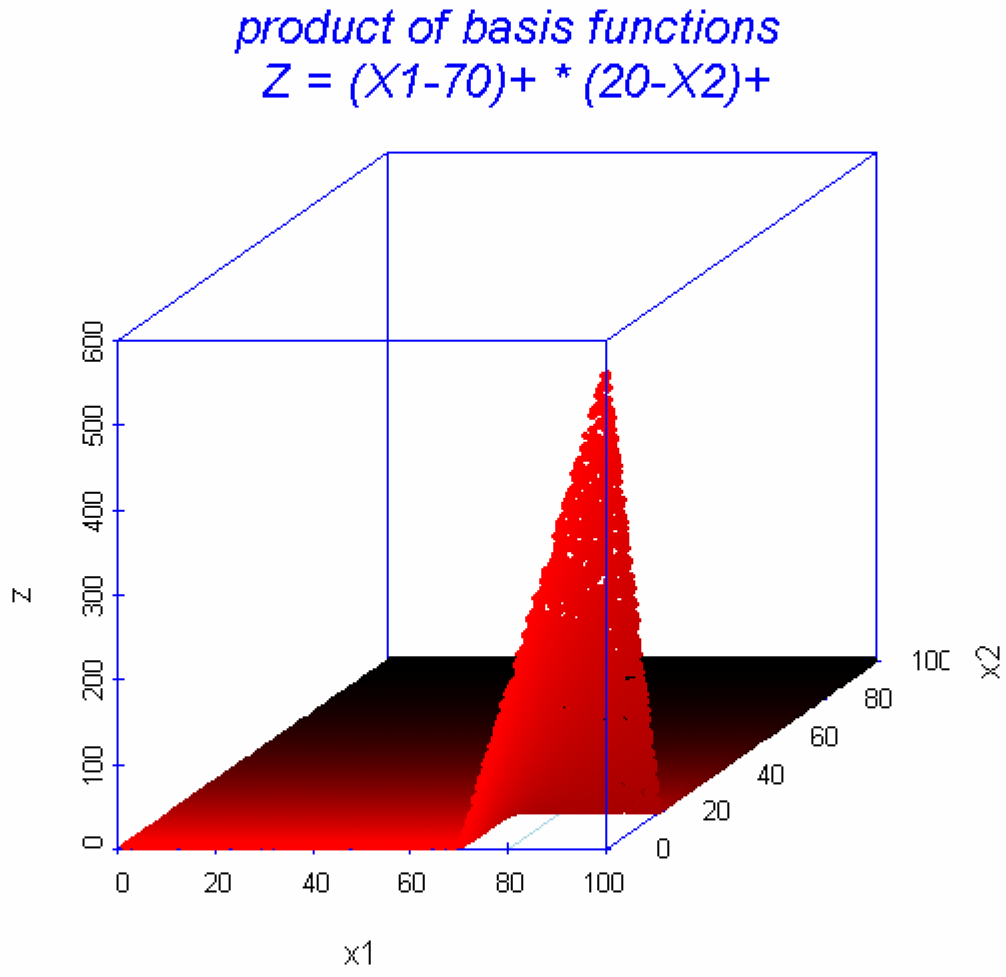
- As an example, suppose $h_5=(X_1\text{-}70)_+$.

- One of the interaction terms MARS will consider is:

  $(X_1\text{-}70)_+ * (20\text{-}X_2)_+$

- Unlike traditional interaction terms used in regression, these can be zero over much of the feature space.

- Multidimensional regression surface can be constructed "locally".

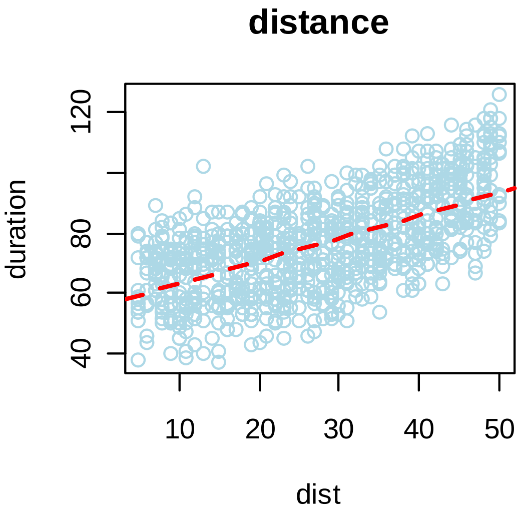

product of basis functions
Z = (X1-70)+ * (20-X2)+

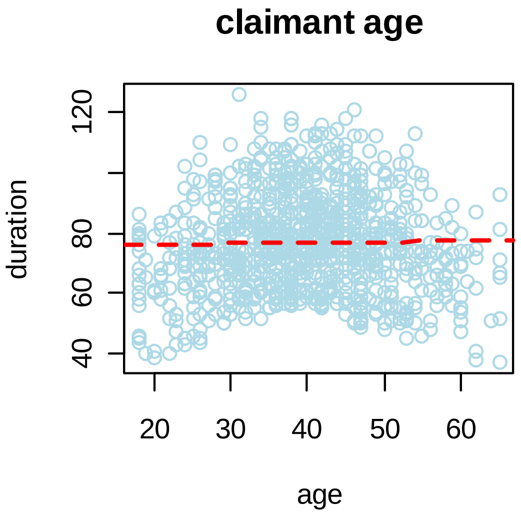
Multivariate Example – Predicting Claim Duration

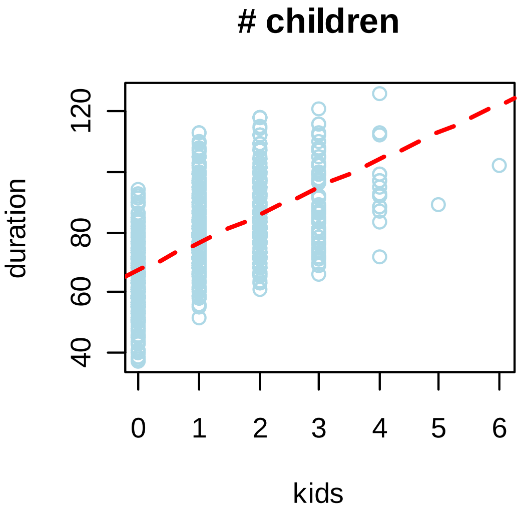
- 1000 observations
  - Target: claim duration
    - Workers comp
    - Lower back claims
  - Predictors:
    - Miles driven to work
    - Claimant age
    - # children

- Note non-linear relationships
  - Red line: 1-way regression fits

- Also – possibility of variable interactions.


distance


claimant age


# children

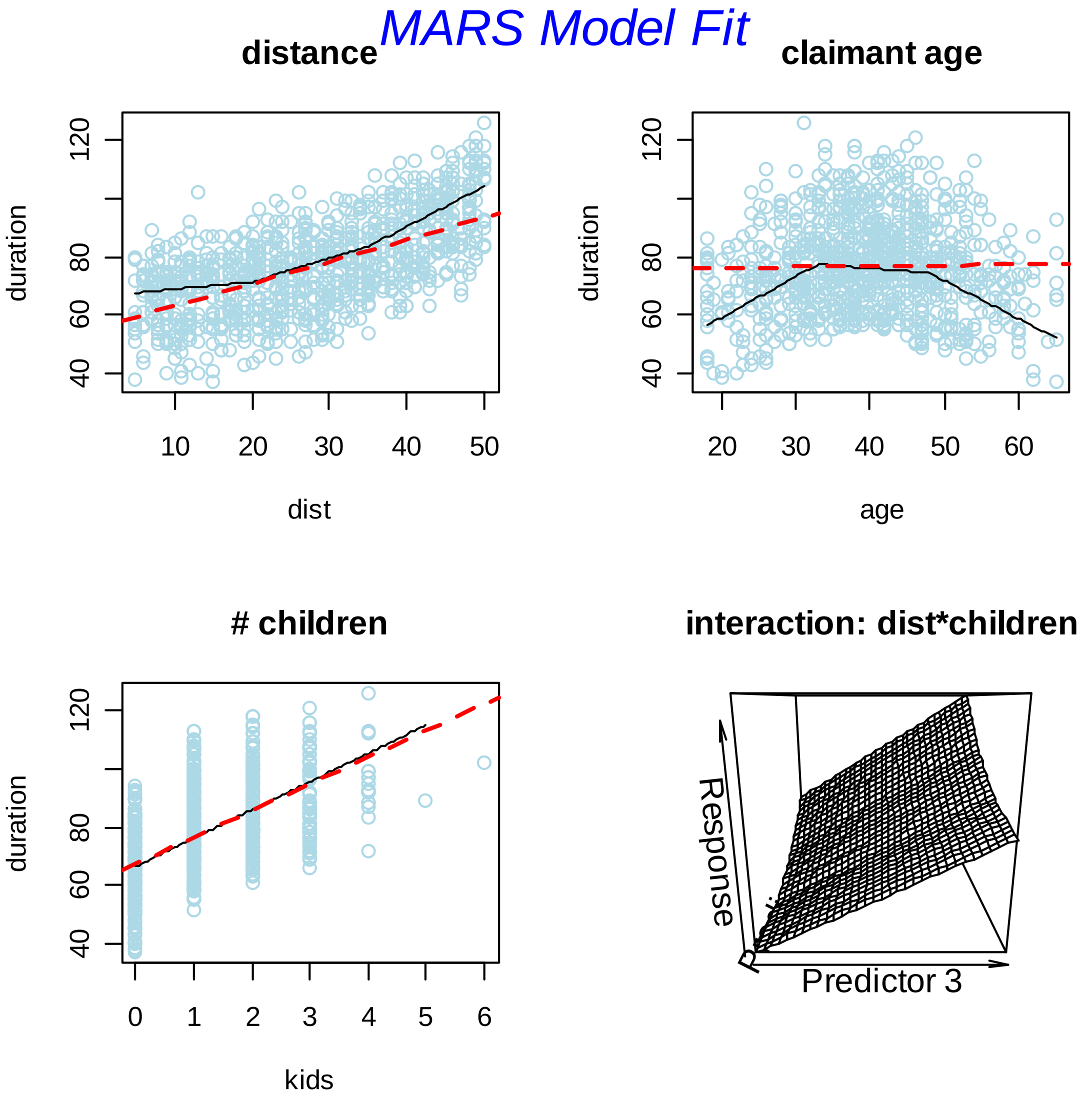
Model Fit

- Fit MARS model on 600 data points (randomly selected)

- MARS model:

$$
\begin{aligned}
y^{\wedge} = 13.37 &+ 0.19 * (dist) \\
&+ 8.26 * (kids) \\
&+ 1.4 \ \ * (age) \\
&+ 0.56 * (dist-20)_{+} \\
&+ 0.56 * (dist-35)_{+} \\
&- 1.59 * (age-33)_{+} \\
&- 1.1 \ \ * (age-48)_{+} \\
&+ .05 \ \ * (dist)*(kids)
\end{aligned}
$$

- Note MARS selected an interaction term.



Model Diagnostics

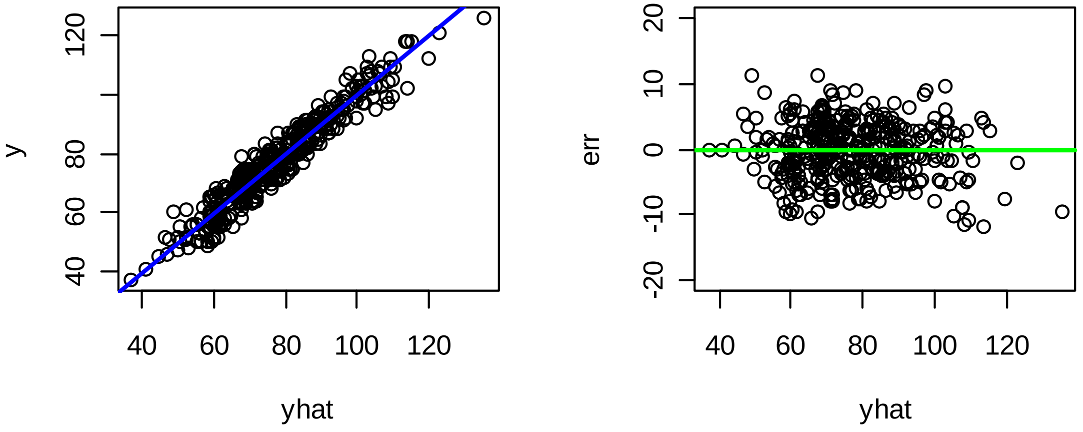
- Diagnostic plots on 400 validation data points (Not used to fit model).

- R^2_{MARS} = 0.93
- R^2_{reg} = 0.77

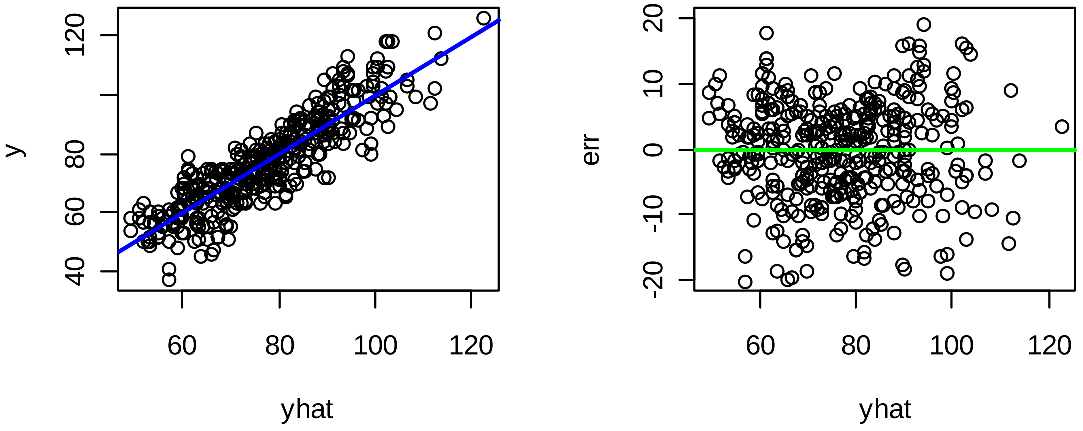
- Regression model (with no interaction) contains 4 parameters.
- MARS model contains 9 parameters.

- Significantly better fit, but still interpretable.

MARS diagnostic plots (validation data)



Regression diagnostic plots (validation data)



Deloitte.

Neural Networks

ANN: Artificial Neural Networks

Audit.Tax.Consulting.Financial Advisory.

Background

- Neural Networks originated in the Artificial Intelligence (AI) community as a way of modeling the workings of the human brain.

- But in hindsight they can be viewed in much more modest terms.

- They are a way of generalizing logistic regression.

- The basic *idea* is fairly simple…
    - Goal of this presentation

- … but the actual *models* produced are complex “black boxes”.

Neural Net Motivation

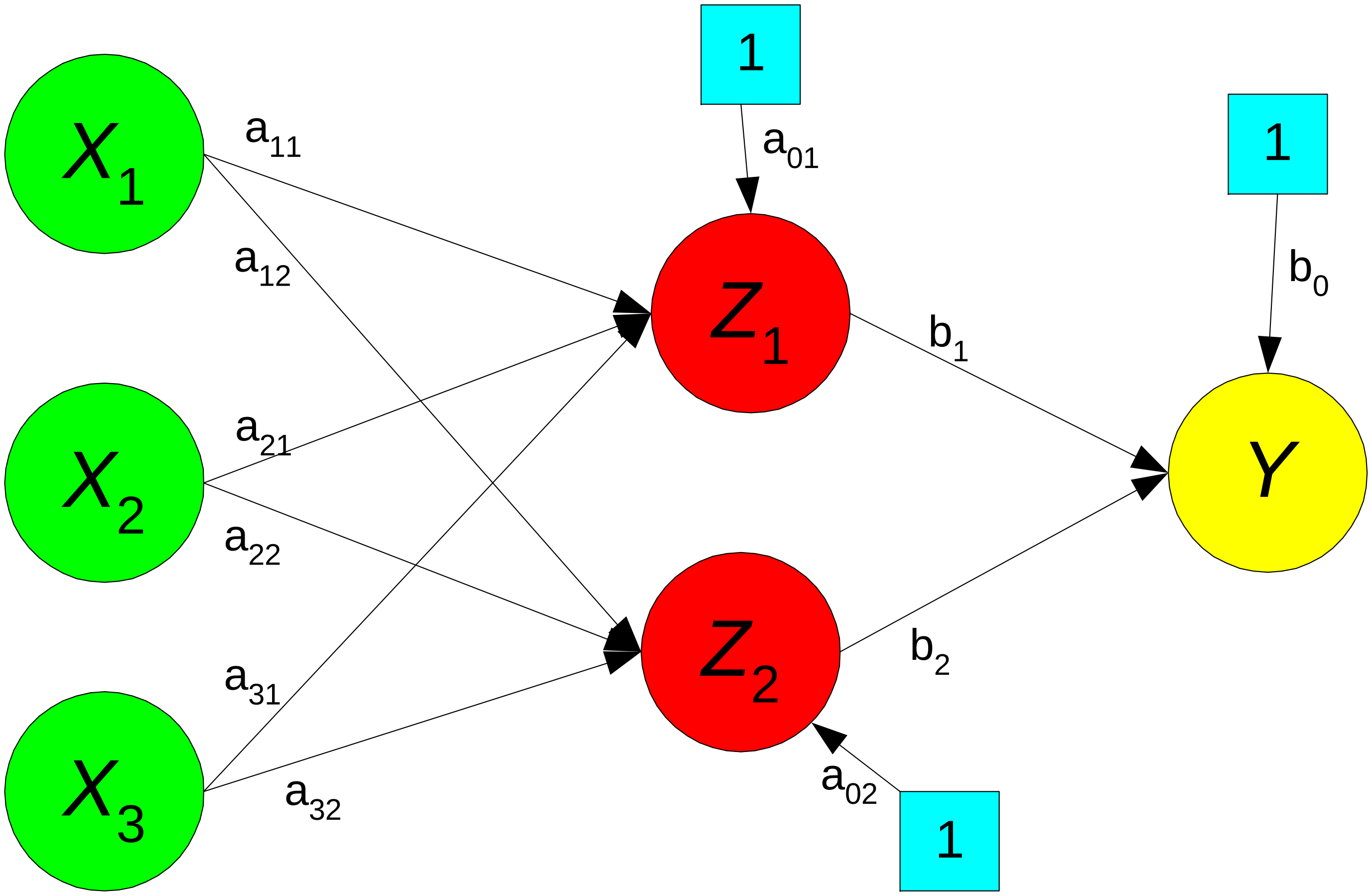
- Let X_1, X_2, X_3 be three predictive variables
  - claimant age, distance to work, #children

- Let Y be the target variable
  - claim duration

- A NNET model is a complicated, non-linear, function φ such that:

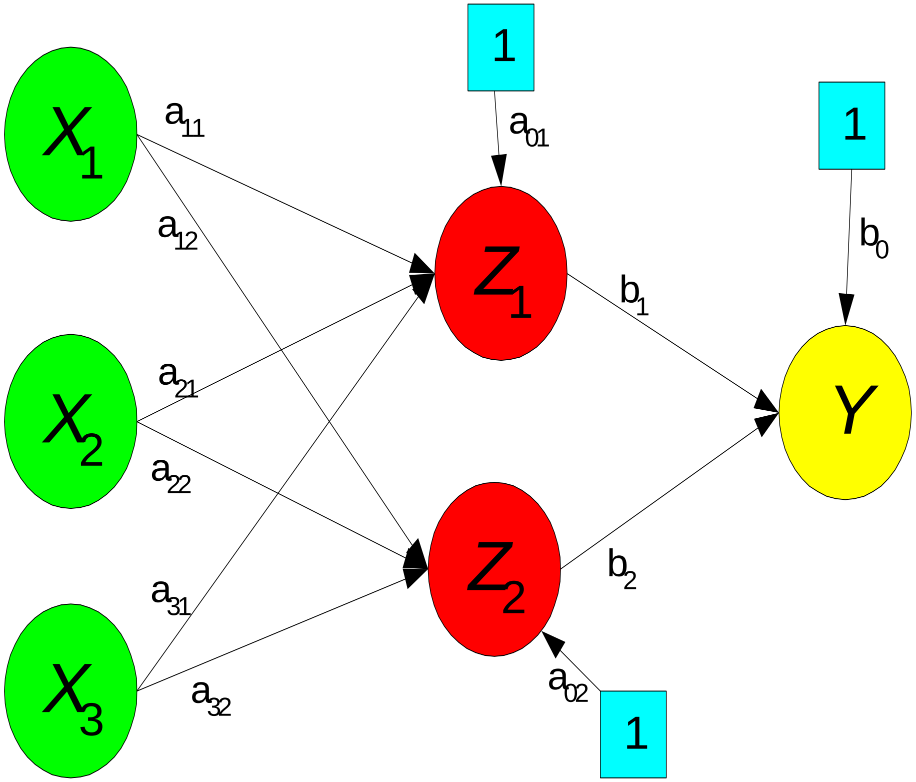
$$\varphi(X_1, X_2, X_3) \approx Y$$

In visual terms…



NNET lingo

- Green: “input layer”
- Red: “hidden layer”
- Yellow: “output layer”
- The {*a*, *b*} numbers are “weights” to be estimated.
- The network *architecture* and the *weights* constitute the model.

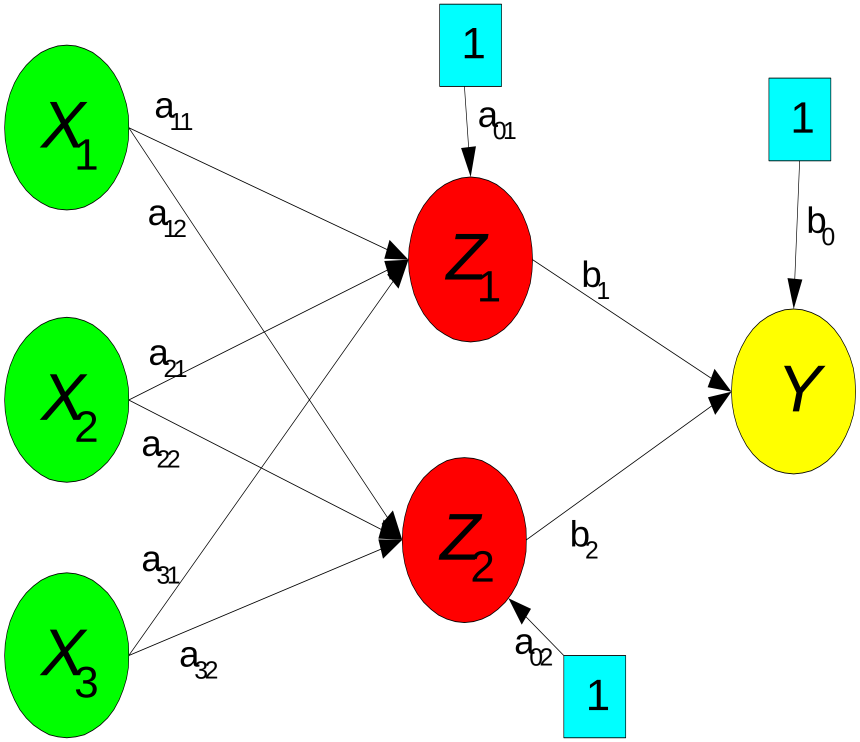


In more detail...

$$Z_1 = \frac{1}{1+e^{a_{01}+b_{11}x_1+b_{21}x_2+b_{31}x_3}}$$

$$Z_2 = \frac{1}{1+e^{a_{02}+b_{12}x_1+b_{22}x_2+b_{32}x_3}}$$

$$Y = \frac{1}{1+e^{b_0+b_1z_1+b_2z_2}}$$



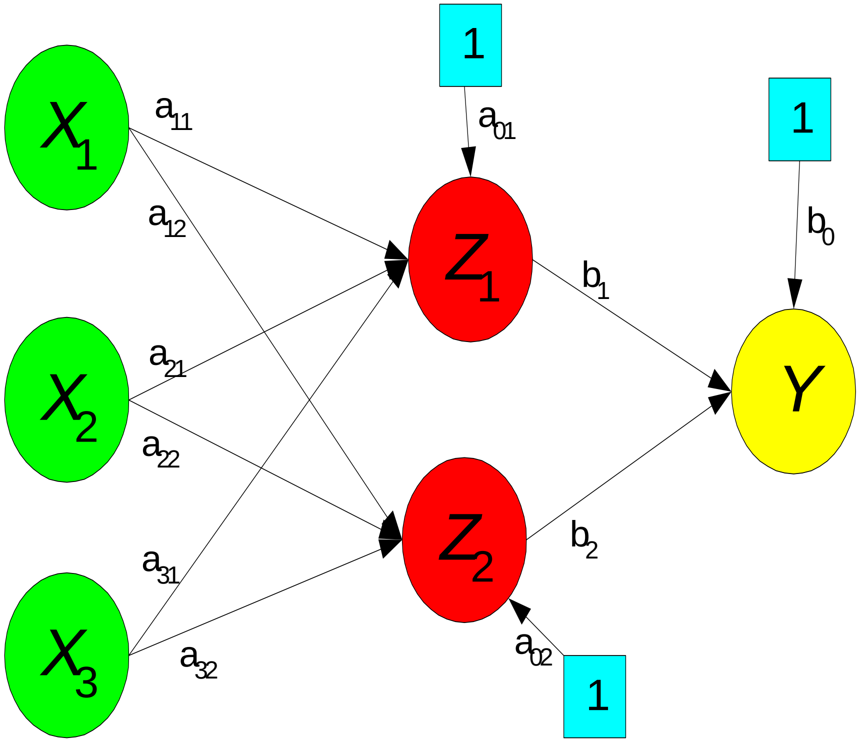
- The NNET model results from substituting the expressions for Z_1 and Z_2 in the expression for Y.

In more detail…

$$Z_1 = \frac{1}{1 + e^{a_{01} + b_{11}x_1 + b_{21}x_2 + b_{31}x_3}}$$

$$Z_2 = \frac{1}{1 + e^{a_{02} + b_{12}x_1 + b_{22}x_2 + b_{32}x_3}}$$

$$Y = \frac{1}{1 + e^{b_0 + b_1 z_1 + b_2 z_2}}$$



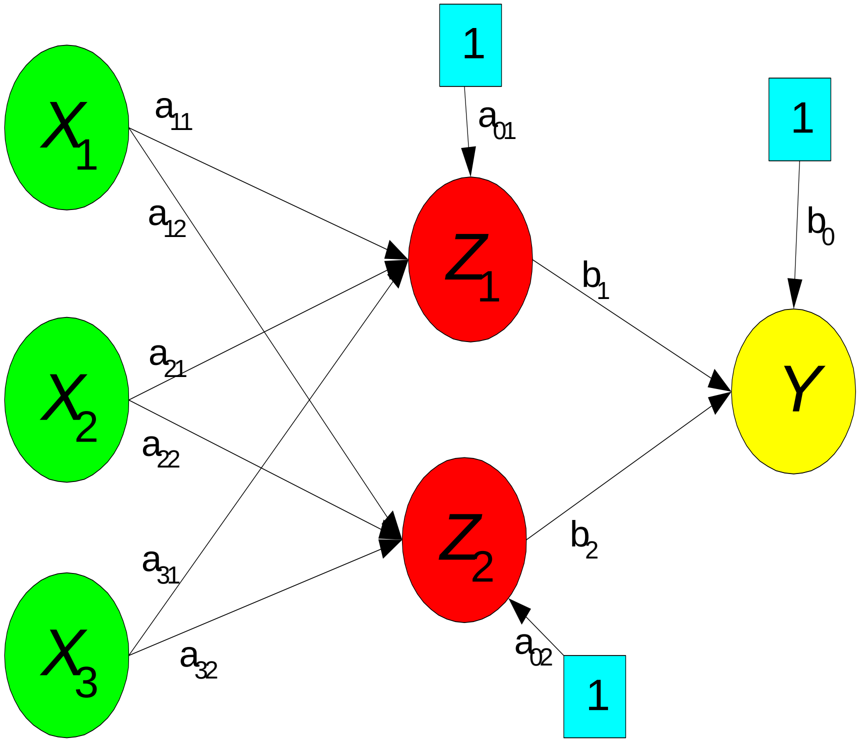
- Notice that the expression for *Y* has the form of a logistic regression.
- Similarly with Z_1, Z_2.

In more detail...

$$Z_1 = \frac{1}{1+e^{a_{01}+b_{11}x_1+b_{21}x_2+b_{31}x_3}}$$

$$Z_2 = \frac{1}{1+e^{a_{02}+b_{12}x_1+b_{22}x_2+b_{32}x_3}}$$

$$Y = \frac{1}{1+e^{b_0+b_1z_1+b_2z_2}}$$



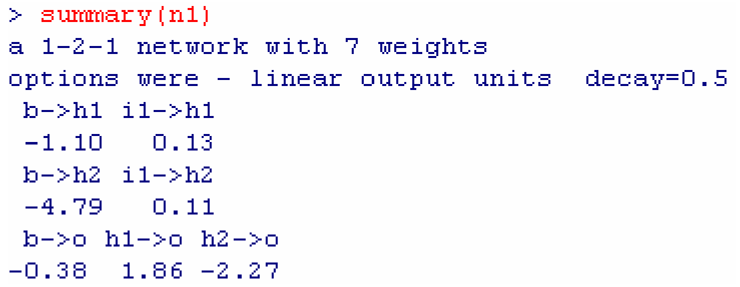
- You can therefore think of a NNET as a set of logistic regressions embedded in another logistic regression.

Universal Approximators

- The essential idea: by layering several logistic regressions in this way…
- …we can model *any* functional form
  - no matter how many non-linearities or interactions between variables X_1, X_2,…
  - by varying # of nodes and training cycles only
- NNETs (even with only one hidden layer) are "universal function approximators".
  - If you add enough complexity to the NNET you can capture any functional form.
- Big disadvantage: while the basic idea is simple, any particular model is hard to explain.

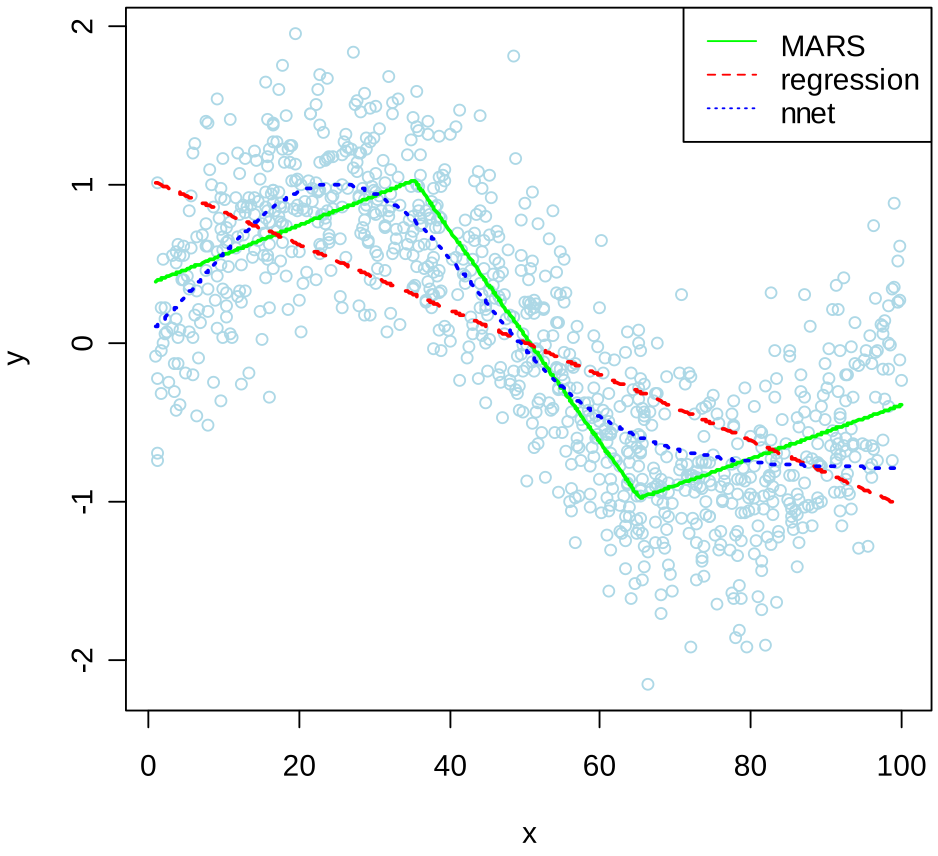
Sine Curve Example Redux

- Fit NNET with 1 hidden layer containing 2 nodes.
  - Regression: 2 parameters, $R^2=0.69$
  - MARS: 4 parameters, $R^2=0.85$
  - NNET: 7 parameters, $R^2=0.83$
- The NNET does well, but how to interpret the parameters?



```
> summary(n1)
a 1-2-1 network with 7 weights
options were - linear output units  decay=0.5
 b->h1 i1->h1
 -1.10   0.13
 b->h2 i1->h2
 -4.79   0.11
 b->o h1->o h2->o
-0.38  1.86 -2.27
```

non-linear modeling problem

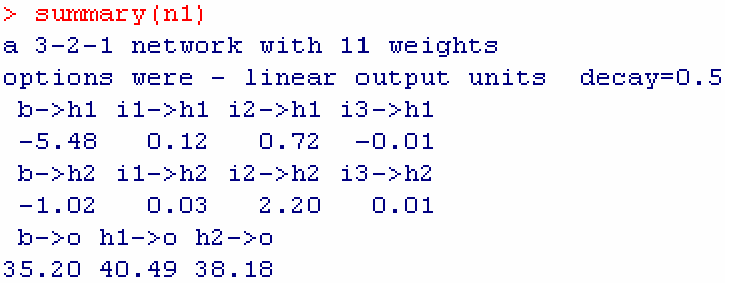


Neural Net Fit to Claims Data

- Fit NNET with 2 nodes in the hidden layer

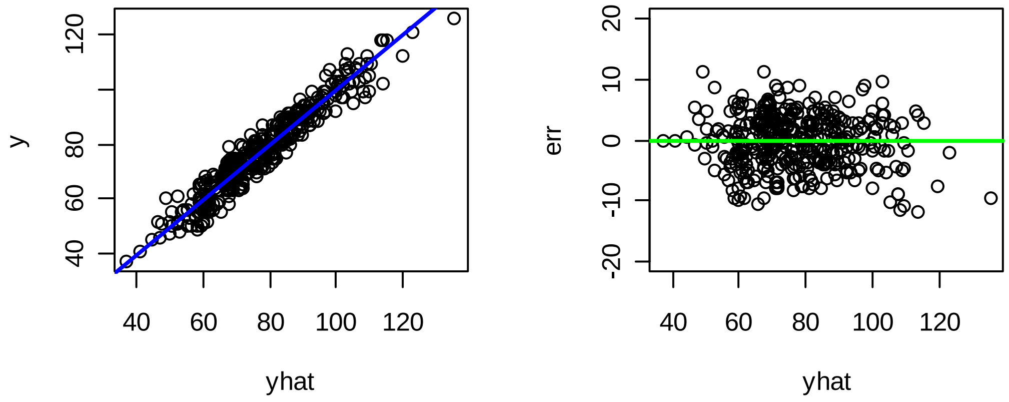
- R^2_{MARS} = 0.93
  - 9 parameters
- R^2_{nnet} = 0.84
  - 11 parameters

- Worse fit than MARS, but already hard to interpret.

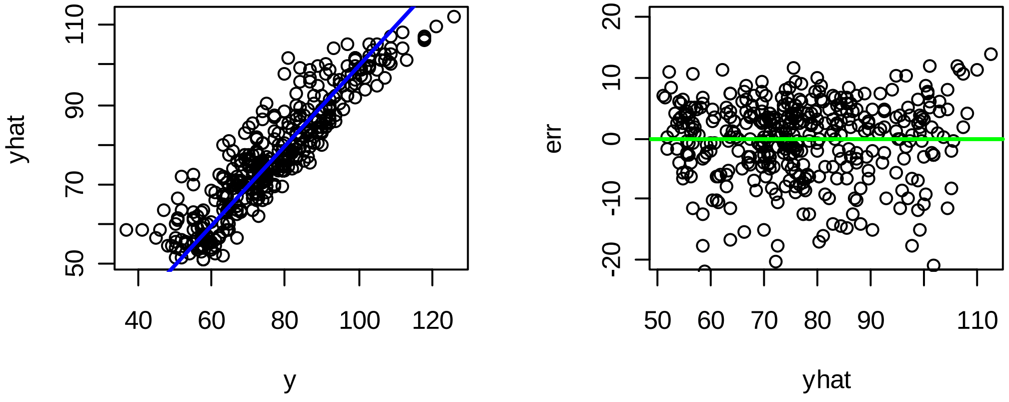


```
> summary(n1)
a 3-2-1 network with 11 weights
options were - linear output units  decay=0.5
 b->h1 i1->h1 i2->h1 i3->h1
 -5.48   0.12   0.72  -0.01
 b->h2 i1->h2 i2->h2 i3->h2
 -1.02   0.03   2.20   0.01
 b->o h1->o h2->o
35.20 40.49 38.18
```

MARS diagnostic plots (validation data)



Neural Net - 1 hidden layer, 2 nodes



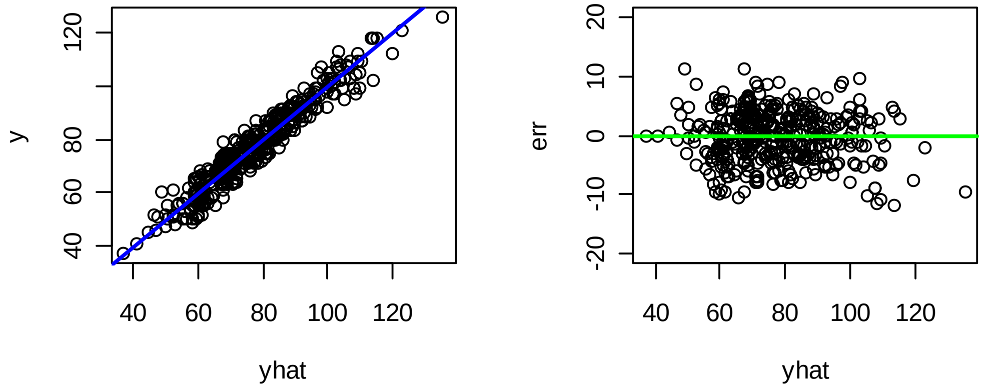
Neural Net Fit to Claims Data

- Now use 3 nodes

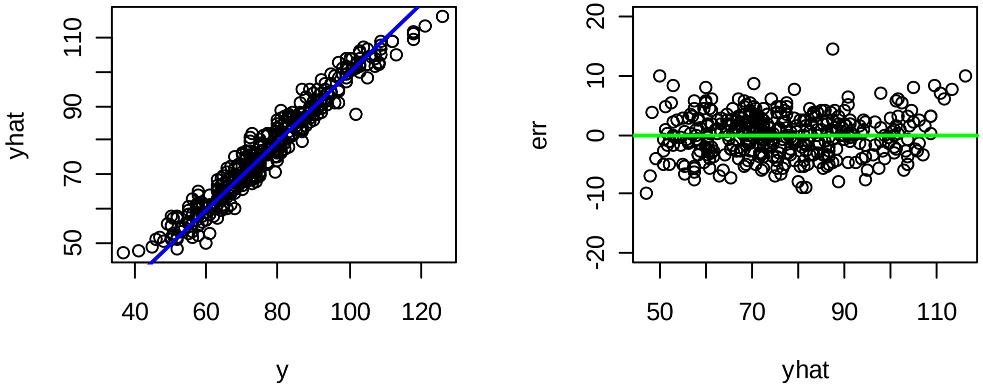
- $R^2_{MARS} = 0.93$
  - 9 parameters
- $R^2_{nnet} = 0.95$
  - 16 parameters

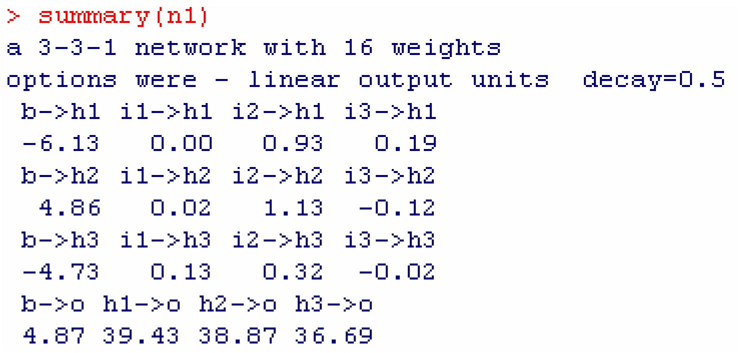
- Now the fit is comparable.

MARS diagnostic plots (validation data)



Neural Net - 1 hidden layer, 3 nodes





```
> summary(n1)
a 3-3-1 network with 16 weights
options were - linear output units  decay=0.5
 b->h1 i1->h1 i2->h1 i3->h1
 -6.13   0.00   0.93   0.19
 b->h2 i1->h2 i2->h2 i3->h2
  4.86   0.02   1.13  -0.12
 b->h3 i1->h3 i2->h3 i3->h3
 -4.73   0.13   0.32  -0.02
 b->o h1->o h2->o h3->o
 4.87 39.43 38.87 36.69
```

References

- For Beginners:

  Data Mining Techniques

  --Michael Berry & Gordon Linhoff

- For Mavens:

  The Elements of Statistical Learning

  -- Jerome Friedman, Trevor Hastie, Robert Tibshirani
  2001, Springer Verlag

  Classification and Regression Trees

  --Robert Breiman, Jerome Friedman, Robert Olshen, Charles Stone
  1984, Wadsworth

  Pattern Recognition and Neural Networks

  --Brian Ripley
  1996, Cambridge University Press