# ANTITRUST Notice

The Casualty Actuarial Society is committed to adhering strictly to the letter and spirit of the antitrust laws. Seminars conducted under the auspices of the CAS are designed solely to provide a forum for the expression of various points of view on topics described in the programs or agendas for such meetings.

Under no circumstances shall CAS seminars be used as a means for competing companies or firms to reach any understanding – expressed or implied – that restricts competition or in any way impairs the ability of members to exercise independent business judgment regarding matters affecting competition.

It is the responsibility of all seminar participants to be aware of antitrust regulations, to prevent any written or verbal discussions that appear to violate these laws, and to adhere in every respect to the CAS antitrust compliance policy.

# Machine Learning

Insurance Applications and Recent Advances

Prof. Paul Beinat
Research and Development, EagleEye Analytics and
Centre For Quantum Computation and Intelligent Systems
University of Technology Sydney

## Value Proposition – What's Different

*Statistical Science*
2001, Vol. 16, No. 3, 199–231
**Statistical Modeling: The Two Cultures**
**Leo Breiman**

"*Abstract.* There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown.

The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems.

Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools."

**Leo Breiman** (January 27, 1928 – July 5, 2005) was a distinguished statistician at the University of California, Berkeley. He was the recipient of numerous honors and awards, and was a member of the United States National Academy of Science.
Breiman's work bridged the gap between statisticians and computer scientists, particularly in the field of machine learning.

## Machine Learning

- Why do it?
- Types of problem classes
- Data issues
- Methods
- Insurance Applications
  - Induction
  - Continuous Induction
  - Multivariate Fusion

## Why do Machine Learning?

- Database resources are largely going to waste
  - MIS, DSS are not machine learning
- Increasing rate of data collection
- Knowledge Based Systems
  - Domain Experts have limited knowledge
  - Knowledge Acquisition is a slow process

## Machine Learning Myths

- Machine learning tools need no guidance.
- Machine learning requires no data analysis skill.
- Machine learning eliminates the need to understand your business and your data.
- Machine learning tools are “different” from statistics.

## Types of Problems

- Classification
  - Fraud detection, loan approval, etc.
- Regression
  - Insurance rating, stock price prediction, etc.
- Clustering (Pattern Detection)
  - Customer clustering, time series, etc
- Mixtures

## Data

- Data Types
- Reliability
  - Missing data
  - Skewed data
  - Noisy Data
- Variability
  - Data changes over time
- Sufficiency
  - Sample size
- Testing implications
  - Validation data

## Classification Learning

- Supervised Learning, Inductive Learning
- Dataset contains examples of input attributes and their corresponding classification

## How Does It Work?

- Examine every input attribute
  - Find the best split that can be made
- Select the best attribute, with the best split
- Build a branch for it and assign the appropriate subset of the data
- Determine if any branch is a leaf node
- Send that subset back to the start

## Limitations

- No guaranteed way to find optimal solution
- Induction must compromise
  - Specificity
  - Simplicity
- OR
  - Accuracy
  - Description length

## Good and Bad of Induction

- Many input attributes
- Easy to understand results
- Relatively fast to run
- Relatively easy to use
- Several algorithms to choose from
  - IDS, C4.5, CART, O-Btree

- No continuous non-linear effects
- No optimal way of partitioning numeric attributes
- Poor performance on noisy data
- No algorithm selection rules found
- Regression uncommon

## Knn - K Nearest Neighbor

- Take k nearest instances to make estimate
- Issues
  - How many is k
  - How far away can neighbors be
  - CBR

## How Many in k

- Static - determined at development time
- Dynamic
  - Statistically based
  - Heuristically based
  - Composites
    - With/without Gaussian biases
    - With/without directional imperatives

## How Far Away for Neighbours

- Euclidian distance
  - Difficult with categorical data
  - Development of biases for axes
- Heuristic distance
  - Encodes domain knowledge
- Non-linear
  - Interaction between variables

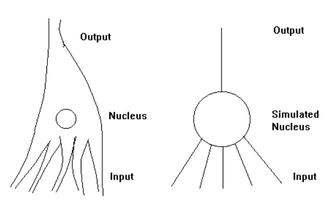
## Classification

- Easy when all neighbors agree
- Problems
  - Sparsity of data locally
  - Neighbors not in agreement
    - Probabilistic democratic classification
    - Weighted probabilities base on distance
  - Edges - all neighbors on one side

## Regression

- Central estimate based on neighbours
- Estimate biased by distance
- Certainty
  - Sample sufficiency
  - Distribution
- Edges
  - Incorporate multi-dimensional linear trends
- Care with skewed distributions

## Neurons - Real v Artificial



## Artificial Neural Networks

- Fundamentals
- Training
- Problems with NN’s
- NN Variants

## Fundamentals

- Each Neuron
  - A Number of Inputs
    - Each Input Has a Weight Associated With It
    - The Weight Moderates Sensitivity To That Input
  - One Output
    - The Output Can be Connected to Many Neurons
    - As Input
  - A Mathematical Function Inside
    - Controls Firing

## Training

- Initialization
  - Weights Must be Randomized
- Learning Function
  - Back propagation
  - Weight Space and Error Function
- Data
  - Training Set
  - Test Set
  - OverFitting

## Using Neural Nets

- Set learning parameters
  - Learning rate, momentum
  - Tolerance
- Train
- Test
- Iterate and tune

## Problems?

- Local Minima!
  - Shaking
  - Genetic Algorithms
- OverFitting
  - Needs validation data to control
- Topology
  - Brittleness to Change (Sine Function NN)

## More Problems

- Black Box
  - No Insight
  - No Expert Validation
- No Domain Knowledge
  - No Ability to Use the Obvious

## Adaptive Networks

- Cascade Correlation
  - Start With Inputs Connected to Outputs - Train
  - Add One Hidden Node - Set Orthogonal to Existing Nodes - Retrain Other Nodes
  - Go Back to Step 2
  - Stop When Adding a Node Does Not Improve Fit
  - Can Produce Mapping with Abrupt Transitions

## Optimal Brain Damage

- Start With Overly Populated NN – Train
- Select Hidden Layer Node with Least
- Sensitivity to Input Nodes
- Delete that Node - Re-Train
- Repeat
- Stop When Next Iteration Does Not Improve Fit

## Radial Basis Functions

- Hidden Units Use Gaussian Activation
- Activation Governed by Euclidian Distance Between Weight and Input Vectors
- Hidden Units Represent Clusters in Data
- Gradient Descent Learning
- Input Attribute Numbers Cause Geometric Explosion of Hidden Units

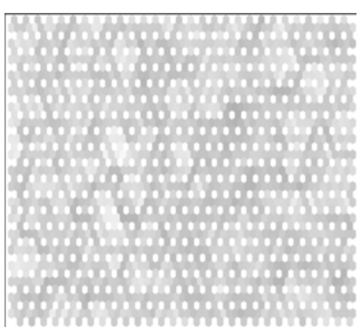
## Kohonen Self Organising Maps

- Unsupervised, Winner Take All Learning (No Output Values)
- Clustering Problem
- Units Arranged in 2 Dimensional Lattice
- Weight Vector Same Dimensionality as Input Vectors
- Randomize Weight Vectors Initially

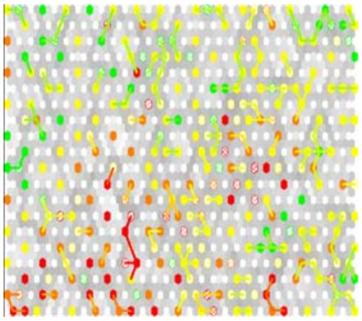
## SOMs

- For Each Input Vector
  - Determine Which Unit Wins
  - Change Its Weight Vector Towards Input Vector (by Learning Rate)
  - Change All Its Neighbors (Limited by Neighborhood Size) Towards Input Vector
- As Training Proceeds
  - Reduce Neighborhood Size
  - Reduce Learning Rate

## SOM Network View



## SOM Data View



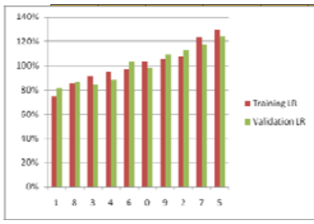
## GLM Related Results

- Methodology
  - Use machine learning inductive techniques to search the residuals of the GLMs
  - Medium size auto portfolio
  - Pure premiums set by consulting actuaries using GLM tools on 4 years of data
  - Pure premiums derived from ALL of the data
  - We divide data into training and validation
  - Search for difference between pure premiums and claims signal
  - Test on validation data

## Results

- Methodology
  - Use only policy attributes
  - One year for training, subsequent year for validation
  - Derive point estimates for segments of the portfolio using a minimum data requirement (number of claims)
  - Derive continuous estimates
  - Test accuracy of estimates on validation data

## Consulting Actuaries A Results




| Training LR | Validation LR |
|---|---|
| 75% | 82% |
| 85% | 87% |
| 91% | 84% |
| 95% | 88% |
| 97% | 104% |
| 103% | 98% |
| 105% | 109% |
| 107% | 113% |
| 124% | 117% |
| 129% | 124% |

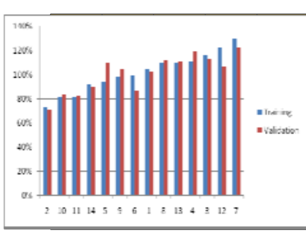
## How well does it do?

- Correlation 0.93
- Lift
  - Using exposure weighted standard deviation of loss ratios
  - Training 16.4%
  - Validation 14.5%

| | Deviance | Squared error | Chi squared error |
|---|---|---|---|
| GLM Premiums | 34.15388 | 1463.838 | 5.47708 |
| Estimated Premiums | 15.02064 | 243.8955 | 0.946702 |

- Result
  - Consistent and large signal present in GLM residuals
  - Better fitting premiums

## Consulting Actuaries B Training Results




| Training LR | Validation LR |
|---|---|
| 73% | 71% |
| 82% | 84% |
| 82% | 83% |
| 92% | 90% |
| 94% | 110% |
| 98% | 105% |
| 99% | 87% |
| 104% | 102% |
| 110% | 112% |
| 110% | 111% |
| 111% | 119% |
| 116% | 113% |
| 123% | 106% |
| 130% | 122% |

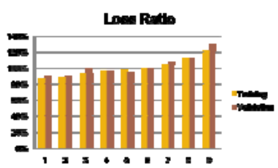
# How well does it do?

- Correlation 0.87
- Lift
  - Using exposure weighted standard deviation of loss ratios
  - Training 16.3%
  - Validation 15.2%

| | Deviance | Squared error | Chi squared error |
|---|---|---|---|
| GLM Premiums | 41.60947 | 2241.3318 | 7.9576 |
| Estimated Premiums | 18.30203 | 737.3713 | 1.814276 |

- Result
  - Consistent and large signal present in GLM residuals
  - Better fitting premiums

# Signal in GLM Residuals




| Loss Ratio | |
|---|---|
| Training | Validation |
| 87.64% | 90.52% |
| 89.24% | 90.41% |
| 93.09% | 100.12% |
| 97.06% | 95.87% |
| 97.62% | 95.24% |
| 99.03% | 99.45% |
| 104.64% | 107.57% |
| 113.41% | 112.19% |
| 122.37% | 129.69% |
| Correlation | 0.962 |

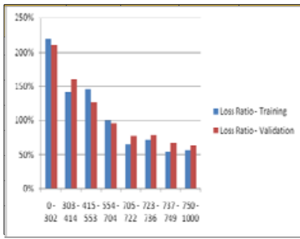
| Frequency | |
|---|---|
| Training | Validation |
| 10.75% | 11.02% |
| 14.18% | 14.01% |
| 14.82% | 15.19% |
| 11.73% | 11.59% |
| 13.95% | 14.03% |
| 17.99% | 17.43% |
| 16.17% | 16.70% |
| 19.09% | 18.04% |
| 15.36% | 15.44% |
| Correlation | 0.986 |

| Severity | |
|---|---|
| Training | Validation |
| $1,775 | $1,799 |
| $1,864 | $1,895 |
| $2,045 | $2,123 |
| $1,880 | $1,886 |
| $2,080 | $2,038 |
| $2,509 | $2,613 |
| $2,239 | $2,220 |
| $2,611 | $2,655 |
| $2,449 | $2,582 |
| Correlation | 0.989 |

# Continuous Estimates

- Modify induction to produce continuous estimates
- Estimates made of exposures based on a 0 to 1000 range
  - 0 is best loss ratio
  - 1000 is worst loss ratio
  - An insurance score – parallel to credit score

## Training Results




| Training Loss Ratio | Validation Loss Ratio |
|---|---|
| 219% | 210% |
| 142% | 160% |
| 146% | 126% |
| 99% | 96% |
| 66% | 77% |
| 71% | 78% |
| 56% | 68% |
| 57% | 65% |

## How well does it do?

- Correlation 0.98
- Lift
  - Using exposure weighted standard deviation of loss ratios
  - Training 30.4%
  - Validation 28.9%

| | Deviance | Squared Error | Chi Squared Error |
|---|---|---|---|
| GLMs | 44.75 | 5722 | 21.74 |
| Output Ranges | 18.23 | 528 | 1.99 |

- Result
  - Consistent and large signal present in GLM residuals
  - Better fitting premiums

## Current Research

- Domain Driven Algorithms
  - Fraud detection for electronic payment transactions
  - Market manipulation detection
- Fusion Algorithms
  - Combining two or more existing algorithms

## Fusion Algorithms

$$F(\mathbf{x}) = \hat{a}_0 + \sum_{k=1}^{K} \hat{a}_k r_k(\mathbf{x}) + \sum_{j=1}^{n} \hat{b}_j l_j(x_j)$$

*r* = rule identity function

- Combination of:
  - Rules (hundreds to thousands)
  - Linear model
  - Additive

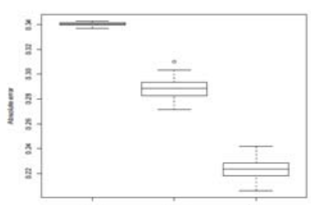
"Predictive Learning via Rule Ensembles"
Jerome Friedman and Bogdan Popescu

## Fusion Algorithms

$$(\{\hat{a}_k\}_0^K, \{\hat{b}_j\}_1^n) = \arg \min_{\{a_k\}_0^K, \{b_j\}_1^n} \sum_{i=1}^{N} L\left(y_i, a_0 + \sum_{k=1}^{K} a_k r_k(\mathbf{x}_i) + \sum_{j=1}^{n} b_j l_j(x_{ij})\right) + \lambda \cdot \left(\sum_{k=1}^{K} |a_k| + \sum_{j=1}^{n} |b_j|\right).$$

- While minimizing the loss function *L*
  - *L* can be any loss function
- Note
  - Lasso term
  - Controls complexity

## Performance




$$\left\{ y_i = 10 \cdot \prod_{j=1}^{5} e^{-2x_{ij}^2} + \sum_{j=6}^{35} x_{ij} + \varepsilon_i \right\}_{i=1}^{N}$$

$x_{ij} \sim U(0,1)$ and $\varepsilon \sim N(0, \sigma^2)$ Gaussian with 2.0 signal to noise

## What about Real Data?

- Does real data exhibit linear and nonlinear components?
- Credit card applications from major bank
  - Bank model continuously developed
  - Compound variables identified
  - Derived variables developed
- Recast algorithm within logistic regression framework

## Credit Card Applications Bank Results

| Score Band | Accepts | Goods | Bads | %Accepts | % Goods | % Bads | Bad Rate | Odds | Cumul Accepts | Cumul. Goods | Cumul Bads | Diagonal | Gini Calc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | 0% | 0% | 0% | 0% | |
| 1 | 11,318 | 9,765 | 1,553 | 5% | 4% | **41%** | 13.72% | 6.29 | 4.9% | 4.3% | 41.1% | 41% | 0.018 |
| 2 | 11,226 | 10,610 | 616 | 5% | 5% | 16% | 5.49% | 17.22 | 9.7% | 9.0% | 57.5% | 57% | 0.022 |
| 3 | 11,316 | 10,878 | 438 | 5% | 5% | 12% | 3.87% | 24.84 | 14.6% | 13.7% | 69.1% | 69% | 0.026 |
| 4 | 11,664 | 11,366 | 298 | 5% | 5% | 8% | 2.55% | 38.14 | 19.7% | 18.7% | 77.0% | 77% | 0.026 |
| 5 | 11,119 | 10,928 | 191 | 5% | 5% | 5% | 1.72% | 57.22 | 24.5% | 23.5% | 82.0% | 82% | 0.021 |
| 6 | 10,824 | 10,650 | 174 | 5% | 5% | 5% | 1.61% | 61.21 | 29.2% | 28.2% | 86.6% | 87% | 0.024 |
| 7 | 12,535 | 12,420 | 115 | 5% | 5% | 3% | 0.92% | 108.00 | 34.6% | 33.7% | 89.7% | 90% | 0.019 |
| 8 | 11,704 | 11,604 | 100 | 5% | 5% | 3% | 0.85% | 116.04 | 39.7% | 38.8% | 92.3% | 92% | 0.019 |
| 9 | 10,476 | 10,421 | 55 | 5% | 5% | 1% | 0.53% | 189.47 | 44.2% | 43.4% | 93.8% | 94% | 0.012 |
| 10 | 12,602 | 12,549 | 53 | 5% | 6% | 1% | 0.42% | 236.77 | 49.6% | 48.9% | 95.2% | 95% | 0.013 |
| 11 | 11,442 | 11,396 | 46 | 5% | 5% | 1% | 0.40% | 247.73 | 54.6% | 53.9% | 96.4% | 96% | 0.013 |
| 12 | 10,990 | 10,948 | 42 | 5% | 5% | 1% | 0.38% | 260.67 | 59.3% | 58.7% | 97.5% | 98% | 0.013 |
| 13 | 11,825 | 11,793 | 32 | 5% | 5% | 1% | 0.27% | 368.54 | 14.9% | 63.9% | 98.4% | 98% | 0.010 |
| 14 | 12,361 | 12,340 | 21 | 5% | 5% | 1% | 0.17% | 587.63 | 20.2% | 69.3% | 98.9% | 99% | 0.007 |
| 15 | 11,643 | 11,634 | 9 | 5% | 5% | 0% | 0.08% | 1292.69 | 25.2% | 74.4% | 99.2% | 99% | 0.003 |
| 16 | 10,213 | 10,202 | 11 | 4% | 4% | 0% | 0.11% | 927.48 | 29.7% | 78.9% | 99.4% | 99% | 0.004 |
| 17 | 12,341 | 12,330 | 11 | 5% | 5% | 0% | 0.09% | 1120.93 | 35.0% | 84.3% | 99.7% | 100% | 0.005 |
| 18 | 12,355 | 12,350 | 5 | 5% | 5% | 0% | 0.04% | 2470.03 | 40.3% | 89.8% | 99.9% | 100% | 0.002 |
| 19 | 11,111 | 11,107 | 4 | 5% | 5% | 0% | 0.04% | 2776.79 | 45.1% | 94.6% | 100.0% | 100% | 0.002 |
| 20 | 12,192 | 12,191 | 1 | 5% | 5% | 0% | 0.01% | 12191.03 | 50.4% | 100.0% | 100.0% | 100% | 0.001 |
| Total | 231,258 | 227,483 | 3,775 | 100% | 100% | 100% | 1.63% | **60.26** | | | | | 0.260 |

**GINI** **74.03%**

## Fusion Algorithm Result

| Score Band | Accepts | Goods | Bads | %Accepts | % Goods | % Bads | Bad Rate | Odds | Cumul Accepts | Cumul. Goods | Cumul Bads | Diagonal | Gini Calc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | 0% | 0% | 0% | 0% | |
| 1 | 11606 | 9815 | 1791 | 5% | 4% | **47%** | 15.43% | 5 | 5.02% | 4.31% | 47.44% | 47.44% | 0.020470 |
| 2 | 11606 | 10748 | 858 | 5% | 5% | 23% | 7.39% | 13 | 10.04% | 9.04% | 70.17% | 70.17% | 0.030352 |
| 3 | 11504 | 11065 | 439 | 5% | 5% | 12% | 3.82% | 25 | 15.01% | 13.90% | 81.80% | 81.80% | 0.026681 |
| 4 | 11598 | 11285 | 313 | 5% | 5% | 8% | 2.70% | 36 | 20.03% | 18.86% | 90.09% | 90.09% | 0.027169 |
| 5 | 11642 | 11496 | 146 | 5% | 5% | 4% | 1.25% | 79 | 25.06% | 23.92% | 93.96% | 93.96% | 0.016546 |
| 6 | 11999 | 11933 | 66 | 5% | 5% | 2% | 0.55% | 181 | 30.25% | 29.16% | 95.71% | 95.71% | 0.009280 |
| 7 | 11164 | 11131 | 33 | 5% | 5% | 1% | 0.30% | 337 | 35.08% | 34.06% | 96.58% | 96.58% | 0.005527 |
| 8 | 12275 | 12247 | 28 | 5% | 5% | 1% | 0.23% | 437 | 40.39% | 39.44% | 97.32% | 97.32% | 0.005451 |
| 9 | 12109 | 12078 | 31 | 5% | 5% | 1% | 0.26% | 390 | 45.62% | 44.75% | 98.15% | 98.15% | 0.006914 |
| 10 | 10178 | 10158 | 20 | 4% | 4% | 1% | 0.20% | 508 | 50.02% | 49.22% | 98.68% | 98.68% | 0.004978 |
| 11 | 13981 | 13964 | 17 | 6% | 6% | 0% | 0.12% | 821 | 56.07% | 55.35% | 99.13% | 99.13% | 0.004709 |
| 12 | 10083 | 10073 | 10 | 4% | 4% | 0% | 0.10% | 1007 | 60.43% | 59.78% | 99.39% | 99.39% | 0.003050 |
| 13 | 6750 | 6747 | 3 | 3% | 3% | 0% | 0.04% | 2249 | 63.35% | 62.75% | 99.47% | 99.47% | 0.000974 |
| 14 | 8748 | 8746 | 2 | 4% | 4% | 0% | 0.02% | 4373 | 67.13% | 66.59% | 99.52% | 99.52% | 0.000685 |
| 15 | 10961 | 10958 | 3 | 5% | 5% | 0% | 0.03% | 3653 | 71.87% | 71.41% | 99.60% | 99.60% | 0.001097 |
| 16 | 14179 | 14172 | 7 | 6% | 6% | 0% | 0.05% | 2025 | 78.00% | 77.64% | 99.79% | 99.79% | 0.002764 |
| 17 | 12719 | 12712 | 7 | 5% | 6% | 0% | 0.06% | 1816 | 83.50% | 83.23% | 99.97% | 99.97% | 0.002983 |
| 18 | 12719 | 12718 | 1 | 5% | 6% | 0% | 0.01% | 12718 | 89.00% | 88.82% | 100.00% | 100.00% | 0.000456 |
| 19 | 12719 | 12719 | 0 | 5% | 6% | 0% | 0.00% | #DIV/0! | 94.50% | 94.41% | 100.00% | 100.00% | 0.000000 |
| 20 | 12719 | 12719 | 0 | 5% | 6% | 0% | 0.00% | #DIV/0! | 100.00% | 100.00% | 100.00% | 100.00% | 0.000000 |
| | 231258 | 227483 | 3775 | 100% | 100% | 100% | | | | | | | 0.170085 |

GINI 82.99%

# Insurance Fusion Algorithm

- Use personal auto data set with GLM derived premiums
- Adapt fusion algorithm
  - Multiplicative rather than additive
  - Recast as maximizing likelihood
- Can it improve GLM premiums?
  - Use test and validation data (random 70%/30%)
  - Measure Tweedie deviance against data set (p=1.72)

# Insurance Results - Summary

| | Training | Validation |
|---|---|---|
| **Null** | 19,165,615 | 8,213,623 |
| **Existing GLM** | 18,659,759 | 7,965,385 |
| **Improvement** | 2.64% | 3.02% |
| **Fusion Model** | 18,551,066 | 7,949,261 |
| **Improvement** | 3.21% | 3.22% |

# Insurance Results - Relativities

| Variable | Rel | Variable | Rel | Variable | Rel | Variable | Rel | Variable | Rel |
|---|---|---|---|---|---|---|---|---|---|
| a1 | 3.73 | b1 | 0.95 | b21 | 1.06 | e1 | 1.23 | n1 | 0.97 |
| a2 | 1.19 | b2 | 1.00 | b22 | 1.21 | e2 | 1.00 | n2 | 1.00 |
| a3 | 1.19 | b3 | 1.23 | b23 | 1.12 | | | n3 | 1.01 |
| a4 | 1.06 | b4 | 1.17 | b24 | 1.12 | f1 | 1.18 | n4 | 1.08 |
| a5 | 1.07 | b5 | 1.09 | b25 | 1.15 | f2 | 1.00 | n5 | 1.32 |
| a6 | 1.09 | b6 | 0.99 | b26 | 1.02 | | | n6 | 1.41 |
| a7 | 1.00 | b7 | 1.05 | b27 | 1.14 | g1 | 1.16 | n7 | 1.47 |
| a8 | 1.00 | b8 | 1.09 | b28 | 1.13 | g2 | 1.06 | n8 | 1.54 |
| a9 | 1.12 | b9 | 1.09 | b29 | 1.22 | g3 | 0.98 | n9 | 1.70 |
| a10 | 1.00 | b10 | 1.12 | b30 | 1.24 | g4 | 1.00 | | |
| a11 | 1.12 | b11 | 1.21 | b31 | 1.20 | g5 | 1.01 | o1 | 0.83 |
| | | b12 | 1.27 | b32 | 1.11 | g6 | 1.06 | o2 | 1.00 |
| c1 | 0.86 | b13 | 1.06 | b33 | 1.10 | g7 | 1.13 | o3 | 0.76 |
| c2 | 1.00 | b14 | 1.12 | b34 | 1.16 | | | o4 | 0.97 |
| | | b15 | 1.00 | b35 | 1.24 | h1 | 1.00 | o5 | 0.79 |
| d1 | 0.93 | b16 | 1.04 | b36 | 1.10 | h2 | 0.86 | | |
| d2 | 0.94 | b17 | 1.21 | b37 | 1.39 | h3 | 1.02 | | |
| d3 | 0.90 | b18 | 1.33 | | | | | | |
| d4 | 0.96 | b19 | 1.18 | | | | | | |
| d5 | 0.86 | b20 | 1.63 | | | | | | |
| d6 | 0.92 | | | | | | | | |
| d7 | 0.89 | | | | | | | | |
| d8 | 1.00 | | | | | | | | |
| d9 | 0.98 | | | | | | | | |
| d10 | 1.00 | | | | | | | | |

## Summary

- Point estimates (a piecewise constant function) derived from training data fit the validation data much better than the GLM alone
  - Improvement in fit is very significant
- Regardless of who fits the GLM
- Continuous estimates (scores) also fit validation data better than the GLM alone
  - Have greater lift
  - More lift and better fit than credit scores
- Fusion Algorithms
  - Contribute significantly more variables, and rules
  - Produce better fitting premiums

## Conclusion

- GLMs
  - Overfit their beta values (validation)
  - Underfit the signal in the data
- Machine learning methods can be a valuable supplement to GLMs
  - Can find extra lift
  - Can improve fit of pure premiums
- More accuracy
  - Better risk selection
  - Reliable price optimization