Imposing Domain Knowledge on Algorithmic Learning
An Effective Approach to Construct Deployable Predictive Models

Dr. Gerald Fahner
FICO
August 28, 2013
Objective

» What lenders seek from models:
   1. Accurate predictions of customer behavior
   2. Insights into fitted relationships
   3. Ability to impose domain knowledge before deploying models

» Algorithmic learning procedures address 1. and 2., but pay scant attention to 3.

» As a consequence, algorithmic models may not be deployable for some credit scoring applications

➢ We propose an effective approach to impose domain knowledge on algorithmic learning that paves the way for deployment
Agenda

» Algorithmic Learning Illustration

» Case Study - Modeling US Home Equity Data
  » Diagnosing Complex Models
  » Algorithmic Learning-Informed Construction of Palatable Scorecards

» Summary
Sketch of Tree Ensemble Models

Data

Prediction Function \((x_1, x_2) \rightarrow E[y]\)

Examples:
- Random Forest [1]
- Stochastic Gradient Boosting [2]
Demonstration Problem

Simulated noisy data from an underlying “true” prediction function to create a synthetic development data set

Next fit Stochastic Gradient Boosting to approximate the data
Stochastic Gradient Boosting

1 Tree

Prediction Function

Residual error
Stochastic Gradient Boosting
5 Trees

Prediction Function

Residual error
Stochastic Gradient Boosting
25 Trees

Prediction Function

Residual error
Stochastic Gradient Boosting
100 Trees

Prediction Function

Residual error
Stochastic Gradient Boosting
200 Trees

Prediction Function

Residual error
pure noise
Agenda

» Algorithmic Learning Illustration

» Case Study - Modeling US Home Equity Data
   » Diagnosing Complex Models
   » Algorithmic Learning-Informed Construction of Palatable Scorecards

» Summary
Problem and Data

Predict likelihood of default on US home equity loans. Binary Good/Bad target.

Sample size: N(total) = 5,960, N(bad) = 1,189. 12 predictor candidates

<table>
<thead>
<tr>
<th>Predictor Candidates</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REASON</td>
<td>‘Home improvement’ or ‘debt consolidation’</td>
</tr>
<tr>
<td>JOB</td>
<td>Six occupational categories</td>
</tr>
<tr>
<td>LOAN</td>
<td>Amount of loan request</td>
</tr>
<tr>
<td>MORTDUE</td>
<td>Amount due on existing mortgage</td>
</tr>
<tr>
<td>VALUE</td>
<td>Value of current property</td>
</tr>
<tr>
<td>DEBTINC</td>
<td>Debt-to-income ratio</td>
</tr>
<tr>
<td>YOJ</td>
<td>Years at present job</td>
</tr>
<tr>
<td>DEROG</td>
<td>Number of major derogatory reports</td>
</tr>
<tr>
<td>CLNO</td>
<td>Number of trade lines</td>
</tr>
<tr>
<td>DELINQ</td>
<td>Number of delinquent trade lines</td>
</tr>
<tr>
<td>CLAGE</td>
<td>Age of oldest trade line in months</td>
</tr>
<tr>
<td>NINQ</td>
<td>Number of recent credit inquiries</td>
</tr>
</tbody>
</table>

Data source as of 07/25/2013: [http://old.cba.ua.edu/~mhardin/DATAMiningdatasets2/hmeq.xls](http://old.cba.ua.edu/~mhardin/DATAMiningdatasets2/hmeq.xls)
Stochastic Gradient Boosting (SGB) Experiments

» Use SGB to approximate log(Odds) of being “Good” [3]
» Will use Area Under Curve (AUC) as evaluation measure throughout

1. Model that is additive in the predictors (trees have 2 leaves)
2. Model capable of capturing interactions between predictors (trees have 15 leaves)

- Interactions appear to be substantial
### Variable Importance
Relative to most important variable

<table>
<thead>
<tr>
<th>Variable</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBTINC</td>
<td>1.00</td>
</tr>
<tr>
<td>DELINQ</td>
<td>0.49</td>
</tr>
<tr>
<td>VALUE</td>
<td>0.45</td>
</tr>
<tr>
<td>CLAGE</td>
<td>0.40</td>
</tr>
<tr>
<td>DEROG</td>
<td>0.34</td>
</tr>
<tr>
<td>LOAN</td>
<td>0.25</td>
</tr>
<tr>
<td>CLNO</td>
<td>0.24</td>
</tr>
<tr>
<td>MORTDUE</td>
<td>0.23</td>
</tr>
<tr>
<td>NINQ</td>
<td>0.23</td>
</tr>
<tr>
<td>JOB</td>
<td>0.20</td>
</tr>
<tr>
<td>YOJ</td>
<td>0.20</td>
</tr>
<tr>
<td>REASON</td>
<td>0.07</td>
</tr>
</tbody>
</table>

### Interaction Test Statistics
Whether a variable interacts with any other variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Interaction Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBTINC</td>
<td>0.029</td>
</tr>
<tr>
<td>VALUE</td>
<td>0.022</td>
</tr>
<tr>
<td>CLNO</td>
<td>0.014</td>
</tr>
<tr>
<td>CLAGE</td>
<td>0.013</td>
</tr>
<tr>
<td>DELINQ</td>
<td>0.010</td>
</tr>
<tr>
<td>YOJ</td>
<td>0.008</td>
</tr>
<tr>
<td>MORTDUE</td>
<td>0.007</td>
</tr>
<tr>
<td>LOAN</td>
<td>0.007</td>
</tr>
<tr>
<td>DEROG</td>
<td>0.006</td>
</tr>
<tr>
<td>JOB</td>
<td>0.006</td>
</tr>
</tbody>
</table>
Score out complex model at regular grid points across the range of one predictor at a time (here $x_1$), while keeping all other predictors fixed.

Here we fix the other predictors to their values for observation 1.

Sweep through range of $x_1$.

Other predictors fixed to their values for observation 1.

Score (logOdds) as function of $x_1$, after centering.
Diagnosing Predictive Relationships Learned by Complex Models

Sweep through range of $x_1$

Score (logOdds) as function of $x_1$, after centering

Other predictors fixed to their values for observations 1, 2, 3

Curves 1, 2, 3

Range of $x_1$
Conditioning on Observations 1 to N

Sweep through range of $x_1$

$(x_1, x_{12} \ldots x_{1P})$

Other predictors fixed to their values for observations 1 to N

Trained SGB

Score (logOdds) as function of $x_1$, after centering

Curves 1 to N

Range of $x_1$
Partial Dependence Plot

Defined in [4] as sample average over the N curves

When $x_1$ enters the model additively…

… then this plot summarizes influence of $x_1$ on score well

When $x_1$ participates in significant interactions…

… then explore relationship further by creating two-dimensional partial dependence plots
One-dimensional Partial Dependence Plots
Examples from Case Study

Partial dependence (higher the better)

Weight of Evidence

Counts

Debt-to-income ratio

Concerned about increasing score?

Number of recent inquiries

Concerned about increasing score?
Interaction between Debt-to-income ratio and Property value

Property value matters less for larger Debt-to-income ratios
## Pros and Cons of Algorithmic Learning

### Pros

<table>
<thead>
<tr>
<th>Objective, no strong assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate fit to historic data</td>
</tr>
<tr>
<td>Push-button procedures (close to)</td>
</tr>
<tr>
<td>Informative diagnostics, considerable insights</td>
</tr>
</tbody>
</table>

### Cons

<table>
<thead>
<tr>
<th>Predictive relationships may not be palatable</th>
</tr>
</thead>
<tbody>
<tr>
<td>No obvious way to impose domain expertise</td>
</tr>
<tr>
<td>May not be deployable</td>
</tr>
</tbody>
</table>
Agenda

» Algorithmic Learning Illustration
» Case Study - Modeling US Home Equity Data
  » Diagnosing Complex Models
  » Algorithmic Learning-Informed Construction of Palatable Scorecards
» Summary
Alternative Modeling Strategies to Fit a Scorecard

Comparing two fitting objectives
1. Approximate log(Odds) of being “Good”
   • Penalized Bernoulli Likelihood
2. Approximate ensemble score from Stochastic Gradient Boosting
   • Penalized Least Squares

Structure of score formula
Score = Sum of staircase functions in 12 binned predictors
Categorical and missing values receive their own bins
Continuous predictors are binned into approximately equal-size bins

Constraints on score formula
Apply linear inequality (‘\(\geq\)’) constraints between score weight of neighboring bins covering these intervals:

Debt ratio in \([5\%, \inf)\)
Number of inquiries in \([0, \inf)\)
to force monotonic decreasing patterns

Constraining staircase functions is an effective approach to avoid counterintuitive score weight patterns

5-fold cross-validation of AUC

Approximate log(Odds)
Approximate SGB score

Avg. number of bins / characteristic
Conjecture About Statistical Benefit of Approximating Ensemble Score

» Ensemble score can be seen as well smoothed version of original dependent variable, carrying less noise than original target and also approximately unbiased

➢ We conjecture that variance of the estimates is reduced by predicting ensemble score instead of predicting the binary target. As a result, score performance is improved

Note:

» Use of penalized MLE to approximate original target also reduces variance of estimates. But empirically we find this approach to be less effective
Constrained Scorecards Address Palatability Issues

Debt to income ratio

Number of recent inquiries

SGB

Scorecard

SGB

Scorecard
Capturing Interactions With Segmented Scorecards
2-stage Approach for Growing Palatable Scorecard Trees

**Stage I**
- Use algorithmic learning to generate ensemble score
- Generate model diagnostics

**Stage II**
- Recursively grow segmented scorecard tree to approximate ensemble score
- Guide split decisions by interaction diagnostics
- May impose domain knowledge via constraints on score formula
CART-like Greedy Recursive Scorecard Segmentation
Informed by Algorithmic Learning and Domain Knowledge

Interaction test statistics provides list of segmentation candidate (domain knowledge can override)

→ {DEBTINC, VALUE, CLNO, ...}

Develop candidate scorecards for tentative splits and pick winner*, if any

Fit (constrained) scorecards as palatable approximations of ensemble score

Sub-pop 1

Score weights

Scorecard 1

Scorecard 2

Sub-pop 2

Parent population

*Need objective function to decide on splitting:

» Least Squares is consistent with scorecard fit objective. But can use other favorite metrics (e.g. AUC with respect to original target) to make split decisions

» Best practice to evaluate splitting objective on a validation set (different from test set)
All Modeling Strategies Compared
Cross-validated Alternative Approaches on Same Folds

5-fold cross-validation of AUC

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>2 leaves</th>
<th>15 leaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stochastic Gradient Boosting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Additive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/Interactions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scorecard 2-stage / Approximate SGB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scorecard Approximate log(odds)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Segmented S/c</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2013 Fair Isaac Corporation. Confidential.
Results With Segmented Scorecards

Case study experiences

» Split decisions are noisy, but more stable when approximating SGB score instead of log(Odds)

» Various tree alternatives come up during cross-validation - finding the best tree is an illusion

» 2-stage approach effective at producing segmented scorecards with performance close to SGB, while a gap remains

Example result from 2-stage approach

Academic research on scorecard segmentation

» Segmentation sometimes but not always helps [5], [6]

Our experiences with big data sets

» Applied 2-stage approach to credit scoring and fraud problems with several million observations and 100+ predictors. Often find segmentations outperforming single scorecards by a few % in AUC, KS

» Restricting segmentation candidates based on SGB interaction diagnostics reduces hypothesis space and accelerates tree search

» Automated segmentations tend to match intuition (clean/delinquent, young/old, …) and shave weeks off tedious manual segmentation analysis

» Segmented scorecards at times achieve close to SGB performance, at other times there remains a gap
Construction of accurate, deployable credit scoring models faces a unique challenge:

Constrained (segmented) scorecards can meet this challenge.

Proposed 2-stage approach combines algorithmic learning and domain knowledge to inform the search for highly predictive, deployable (segmented) scorecards.

We are testing it for projects across credit scoring, application and insurance claims fraud and see good potential to increase the effectiveness of practical scoring models.

Find a highly predictive model within a flexible yet palatable model family.
THANK YOU
References


