Verisk Analytics

Interaction Detection in GLM – a Case Study

THE SCIENCE OF **RISK**SM

Chun Li, PhD
Verisk Analytics

March 2013

Agenda

- Case study
- Approaches
  - Proc Genmod, GAM in R, Proc Arbor
- Details
- Summary

Case Study

- Personal Auto loss prediction
  - Pure premium prediction (GLM – Tweedie)
  - Inputs:
    - Environment components
    - Vehicle components
    - Driver components
    - Household components
  - Objective is to detect interactions among the components to further improve model performance

Components

Environment (frequency and severity for each)

- Traffic density
- Traffic composition
- Traffic generators
- Weather
- Experience and trend

Driver

- Driver characteristics (age, gender, marital, good student etc)
- Violation history
- Claim history

Vehicle

- ISO Symbol relativity
- Price new relativity
- Model year relativity
- Body style and dimension
- Performance and safety
- Theft
- Weather
- Animal
- Glass
- All other perils

Household

- Usage/mileage
- Household composition

Challenges

- There are many different approaches that can be used to detect interactions
- The approach we selected was based on our requirements that:
  - interaction detection be completed in a timely manner
    - despite the large number of observations (>1 million) and large number of interaction pairs (>300)
  - all variables in the final model (including interactions) be interpretable
  - the final model (including interactions) be built in the form of a SAS GLM model

Approach

Step 0
- Build main effect model
- Aim to model the residual using interaction terms

Step I
- Automated pair-wise selection
- Based on standalone contribution

Step II
- Manual selection from Step I results
- Based on marginal contribution in GLM

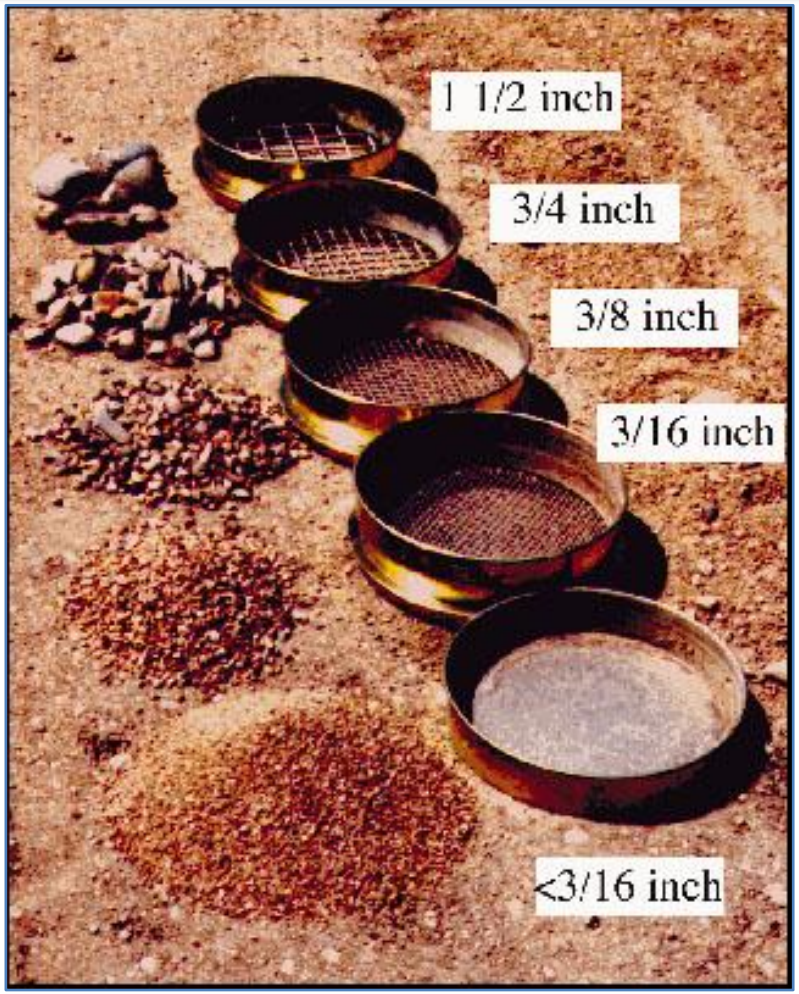
Step III
- Validation/Refinement/Finalization

**** We'll be focusing on Step I***

Step I - Details

The purpose of Step I is to separate significant interaction pairs from insignificant ones, so that we can focus on those that have higher potential.

The principle is to add each pair to the model to predict the residual, measure their contribution, and rank the pairs based on contribution.



Step I - Details

Three methods are used

- Proc Genmod in SAS
- GAM in R
- Proc Arbor (Regression Tree) in SAS

Proc Genmod in SAS

- Use main effect model as offset
- Add a component pair to the model
- Use ‘Increase in Gini’ as the performance metric
- Created SAS macro to loop through all component pairs and output these pairs ranked according to the performance metric

Proc Genmod in SAS

- Interaction terms
  - Both linear
  - Both binned
  - One linear and one binned

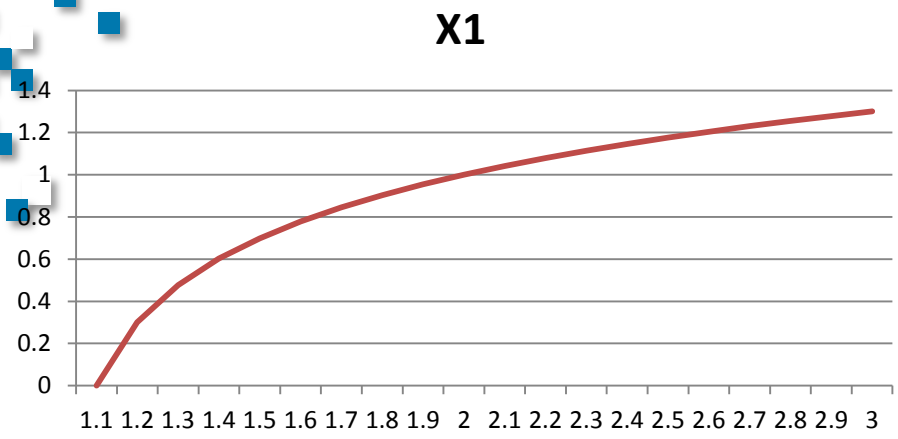
The linear assumption is based on the fact that the components (or sometimes, the log transformation of the components) are developed in the way that they have linear relationship with the target.

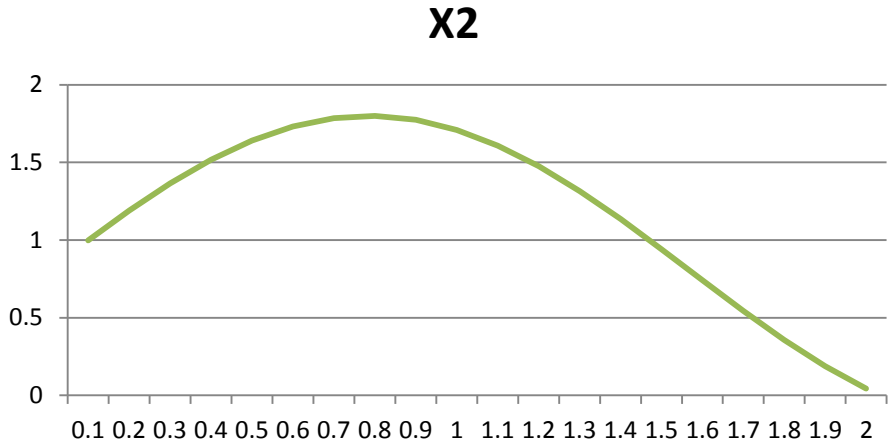
GAM in R

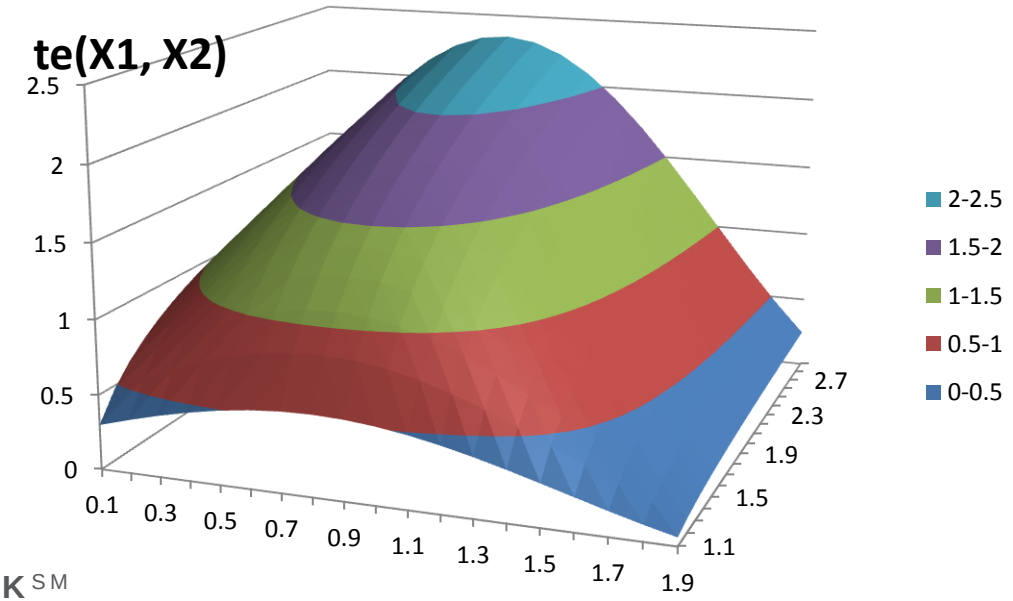
GAM = Generalized Additive Model

- In R package: mgcv
- Able to do Tweedie distribution with Log link
- Fits splines
- Multi-dimentional smoothing for interactions
  - Smoothing classes: s(a, b)
  - Tensor product smoothing: te(a, b)

Illustration of interaction surface







GAM in R

- Use main effect model as offset
- Add a component pair to the model
- Use ‘Decrease in AIC’ as the performance metric
- Create R process to loop through all possible component pairs and output these pairs ranked according to the performance metric

Proc Arbor in SAS

Proc Arbor in SAS

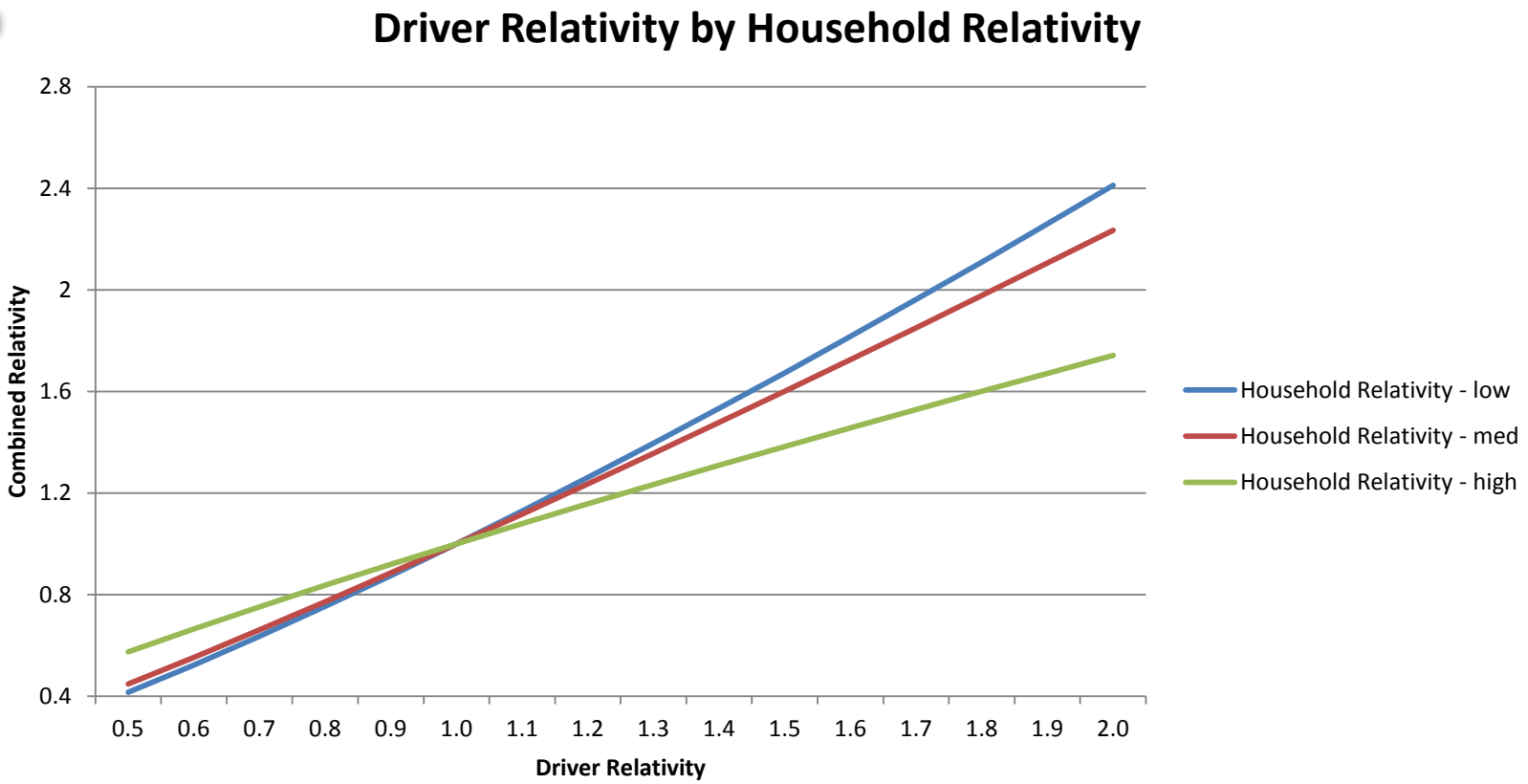
- The same algorithm behind EMiner's Decision Tree Node
- Can be part of a programmable process
  - Loop through component pairs
  - Build model
  - Evaluate model performance

Proc Arbor in SAS

Proc Arbor in SAS

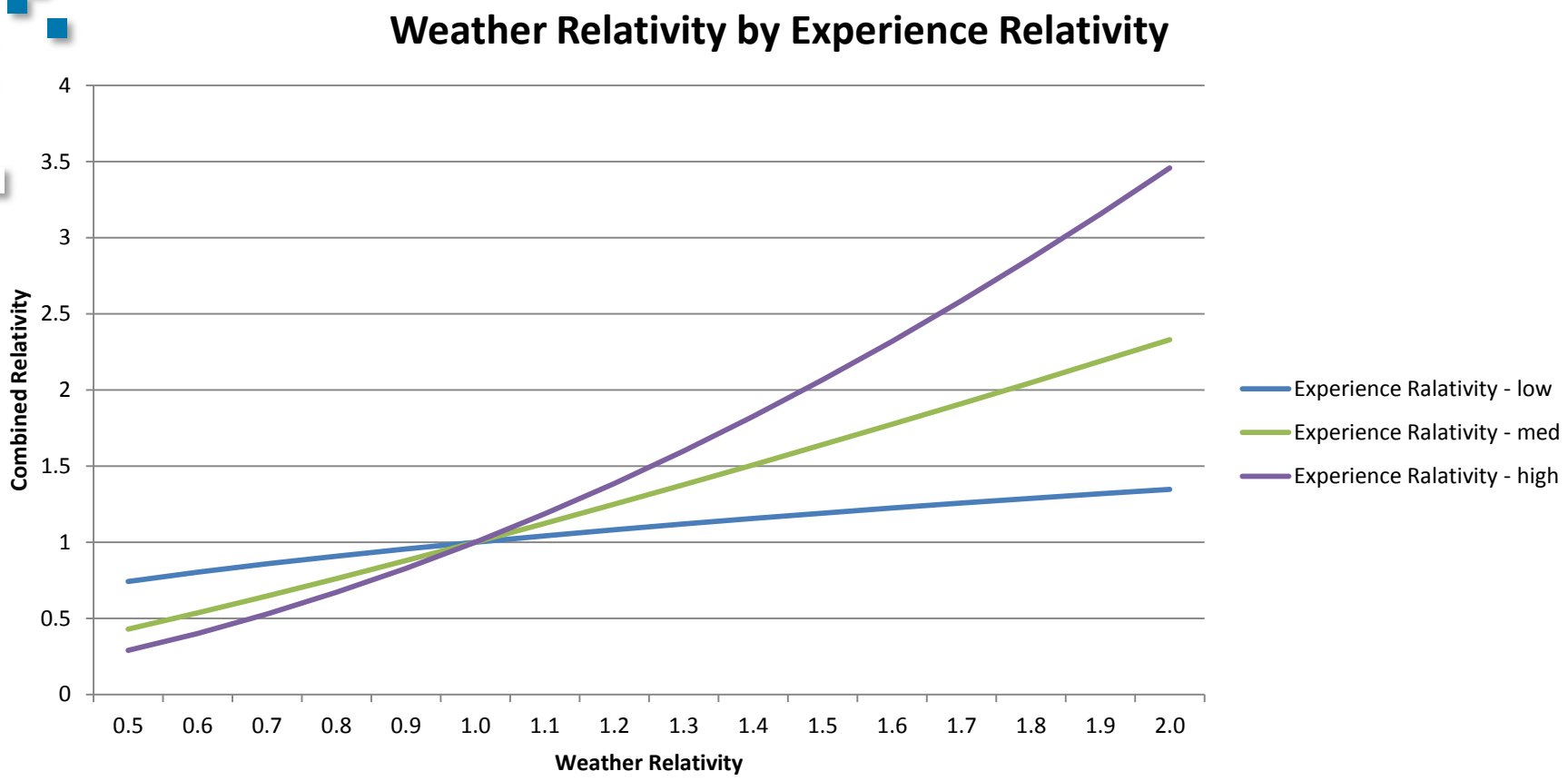
- Use residual of main effect mode as target
- Build regression tree using a pair of components
- Performance metric
  - sqrt(MSE*Leaf_Count)
- Created SAS macro to loop through all possible component pairs and output these pairs ranked according to the performance metric

Example – Collision Coverage



Drivers in the low household relativity segment should have the driver relativity adjusted higher, and high lower.

Example – Collision Coverage



In the location where the loss experience is low, the weather relativity needs to be adjusted lower, and high higher

Summary

- Most of the significant pairs are captured by proc Genmod method
  - Closest to the final model format
- Both GAM in R and proc Arbor detect additional significant interaction pairs
  - Need to convert to the format that Proc Genmod can handle

Take away

- The methodologies described can be applied generally to variable selection processes
  - May need to do variable de-correlation process beforehand (eg. variable clustering)
- Significantly reduces the time/effort needed for variable selection

Q & A

Questions?

Contact: cli@iso.com