A Machine Learning Framework For Loss Reserving

Drew Golfin | drewgolfin@kpmg.com

Kevin Kuo | kevinkuo@kpmg.com

Casualty Loss Reserving Seminar
and Workshops

September 18-20, 2016

Motivation

- **Traditional approach to loss reserving:** A handful of time-tested techniques judgmentally weighted together.

- **The focus of improving loss reserving:** New methods are continually being added to the repertoire of traditional approaches with emphasis on greater accuracy (and some measure of variability). Many of these methods, for example, Bootstrapping, GLMs, and Markov Chain Monte Carlo techniques, are built on advanced statistical methods.

- **The next generation of loss reserving:** Full use of computer power through machine learning approaches including tree-based methods and their enhancements. This presentation will focus on and test an ensembling approach to the reserving problem.

Agenda

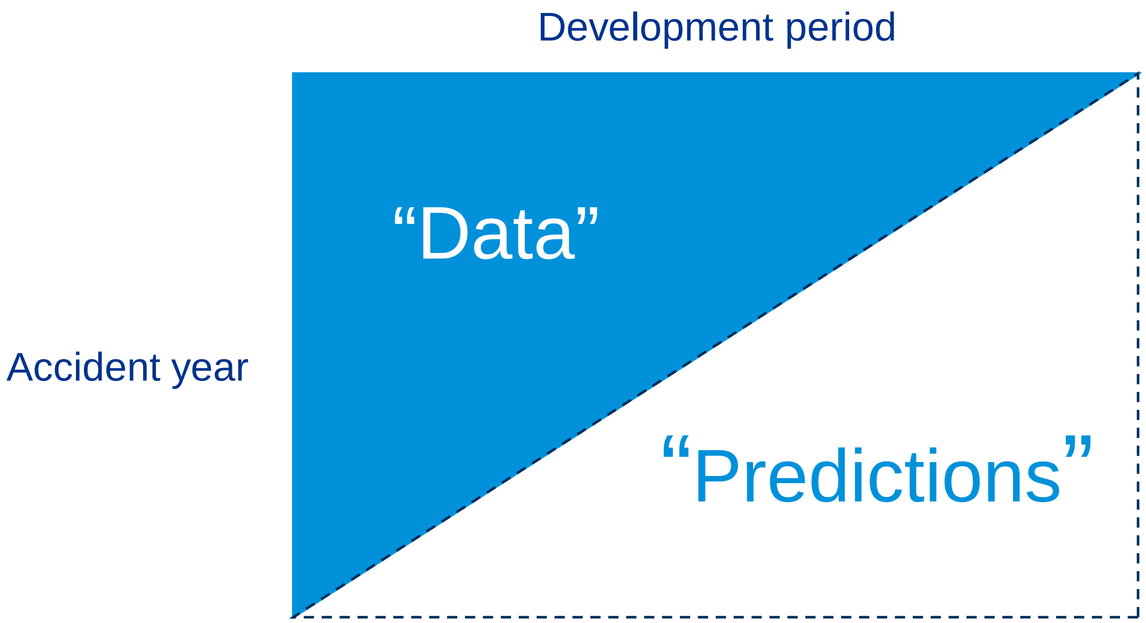
- Loss reserving as a predictive modeling problem
- A model selection framework
- The gradient boosting machine (GBM) algorithm
- An example pipeline
- Performance results on Schedule P data
- Considerations and future work
- Q & A

KPMG

Loss Reserving as a Regression Problem

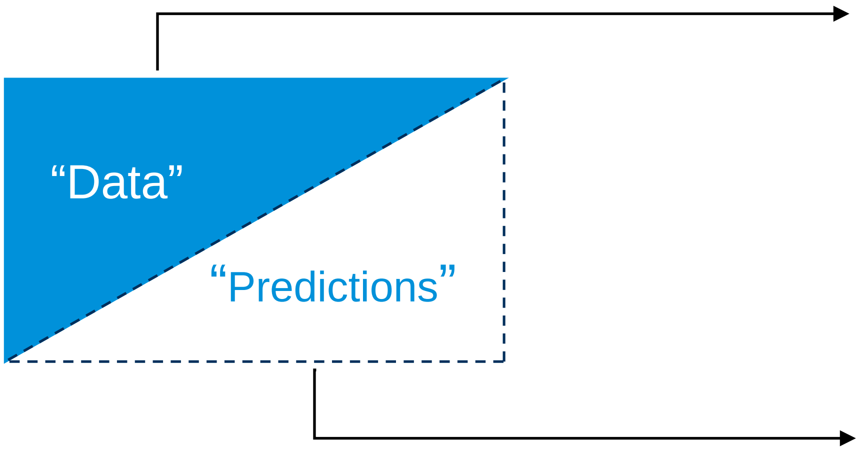
Reserving as a Regression Problem

Unpaid Claims Estimation: "Squaring" the triangle…



Reserving as a Regression Problem

Data in “long” format



AY	Dev	Incremental Loss
2000	1	5,000
2000	2	7,000
2000	3	4,000
2001	1	6,000
…	…	…
2015	3	???
2016	2	???

Reserving as a Regression Problem

Example: Generalized linear model (GLM)

[Predictors] [Response]

AY	Dev	Incremental Loss
2000	1	5,000
2000	2	7,000
2000	3	4,000
2001	1	6,000
…	…	…
2015	3	???
2016	2	???

- Error distribution: Tweedie/Gamma/Poisson
- Predictors: Accident year, development period, etc.

Reserving as a Regression Problem

More generally…

(X) (Y)

AY	Dev	Incremental Loss
2000	1	5,000
2000	2	7,000
2000	3	4,000
2001	1	6,000
…	…	…
2015	3	???
2016	2	???

The Regression Problem:

Find $\hat{f}$ ***such that*** $\hat{f}(X) \approx Y$

Note that when we say "regression" we mean a general prediction problem where the output is numeric. We're *not* limited to things of the form $Y = X\beta + \epsilon$!

Feature Engineering

We're not limited to AY/Dev as predictors!

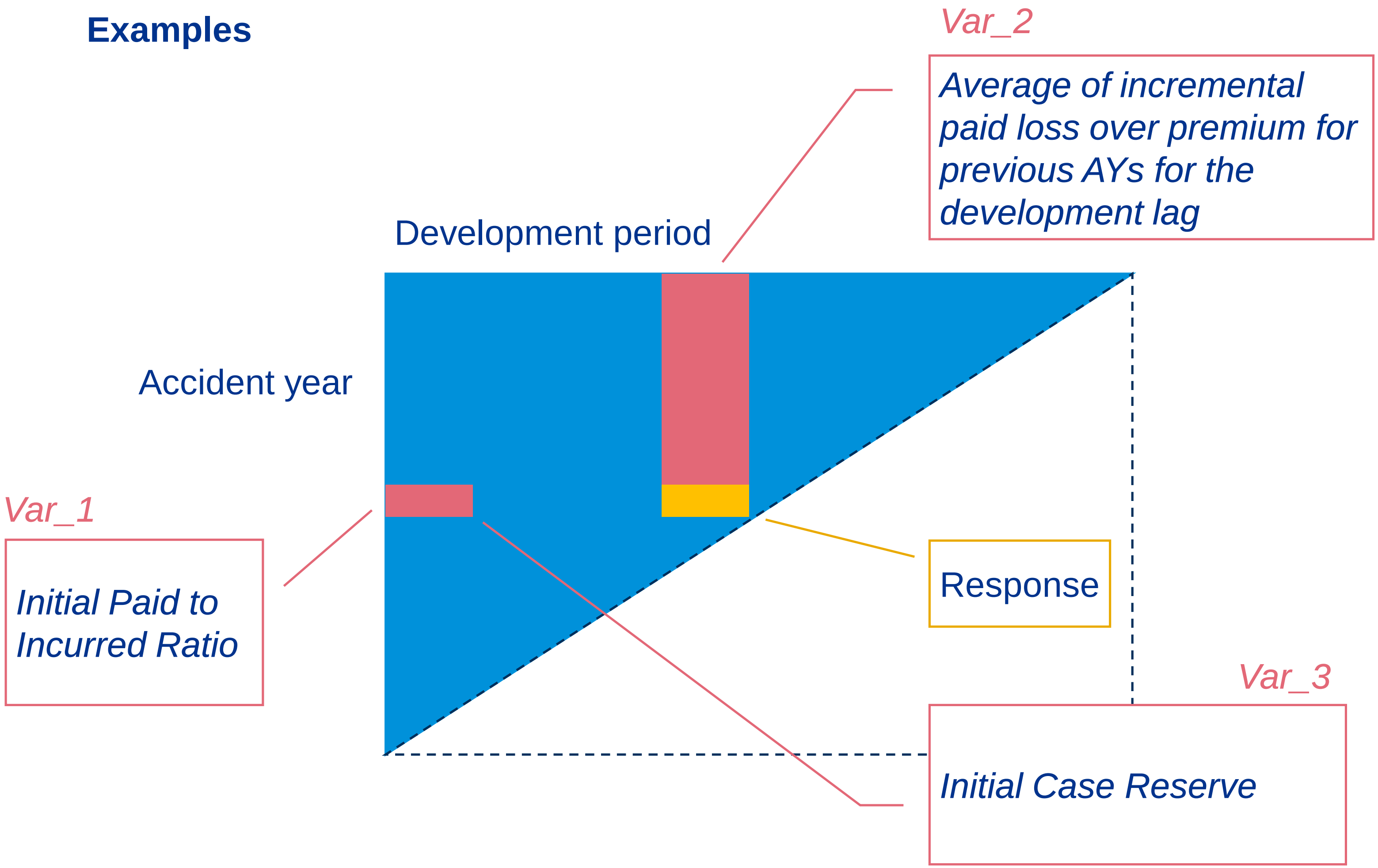
"X" "Y"

AY	Dev	Var_1	Var_2	Var_3	…	Increm loss
2000	1					5,000
2000	2					7,000
2000	3					4,000
2001	1					6,000
…	…					…
2015	3					???
2016	2					???

New Predictors

Feature Engineering

Examples



A Model Selection Framework

Solving the Regression Problem

Terminology: Methods and Models

Method

Training Data

AY	Dev	Var_1	Var_2	Var_3	...	Increm loss
2000	1					5,000
2000	2					7,000
2000	3					4,000
2001	1					6,000
...	...					...

=

Model

Model

Unlabeled “Future” Data

AY	Dev	Var_1	Var_2	Var_3	...	Increm loss
2015	3					???
2016	2					???

=

Predictions

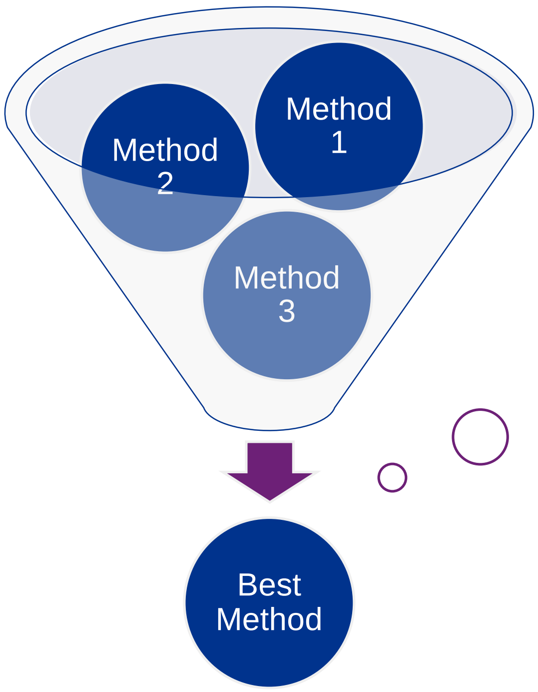
AY	Dev	Increm loss
2015	3	3,500
2016	2	1,500

Example. GLM is a method, the formula that results from model fitting is a model.

Example. Paid Development is a method, the LDFs that result are a model.

Solving the Regression Problem

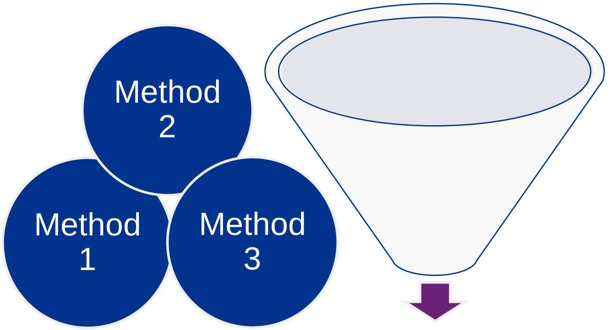
Towards an automated framework



How do we automate the method selection process?

Solving the Regression Problem

Ingredients of an automated method selection framework



1. A set of candidate methods for consideration
   - For example, GLMs with different error distributions
   - For example, paid/reported development, paid/reported BF
2. The error metric on which models will be evaluated on
   - For Example, Mean Squared Error (MSE)
3. A procedure for creating the training and validation datasets

Solving the Regression Problem

Model/Method selection with Cross Validation

Split the dataset into k approximately equal subsets

AY	Dev	Increm loss
2000	1	5,000
2000	2	7,000
2000	3	4,000
2000	4	2,000
2001	1	4,500
2001	2	6,500
2001	3	3,500
2002	1	5,500
2002	2	7,500
2003	1	6,000

Solving the Regression Problem

Model/Method selection with Cross Validation

*For each of the k subsets, use it as the **validation set** and the complement as the **training set***

AY	Dev	Increm loss
2000	1	5,000
2000	2	7,000
2000	3	4,000
2000	4	2,000
2001	1	4,500
2001	2	6,500
2001	3	3,500
2002	1	5,500
2002	2	7,500
2003	1	6,000

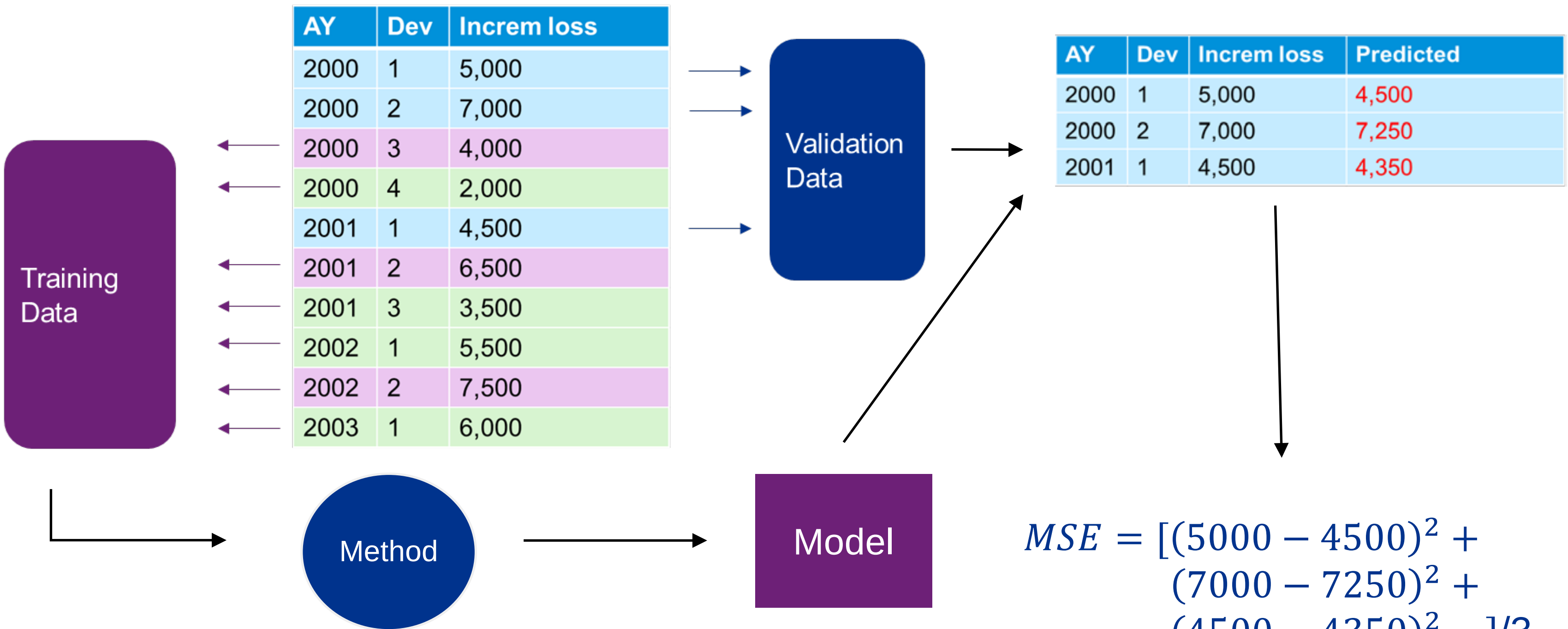
Training Data

Validation Data

Solving the Regression Problem

Model/Method selection with Cross Validation

Use the method to train a model on the training set, then predict losses on the validation set



AY	Dev	Increm loss
2000	1	5,000
2000	2	7,000
2000	3	4,000
2000	4	2,000
2001	1	4,500
2001	2	6,500
2001	3	3,500
2002	1	5,500
2002	2	7,500
2003	1	6,000

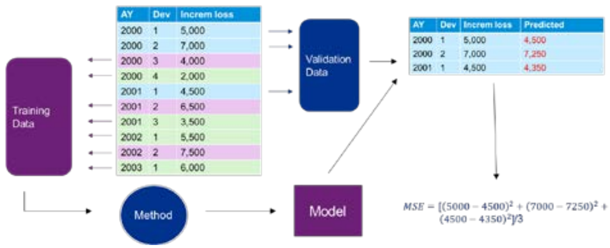
AY	Dev	Increm loss	Predicted
2000	1	5,000	4,500
2000	2	7,000	7,250
2001	1	4,500	4,350

$$MSE = [(5000 - 4500)^2 + (7000 - 7250)^2 + (4500 - 4350)^2 \quad]/3$$

Solving the Regression Problem

Model/Method selection with Cross Validation

Perform the procedure k times and aggregate the results…



k times

MSE for <u>method</u> $= \frac{1}{k}\sum_{i=1}^{k} MSE_i$

…then repeat the above for each method under consideration.

Solving the Regression Problem

Using the best method/model for prediction

Once the "best method" has been selected from cross validation, it can be trained on the full dataset to obtain the "best model" for prediction

AY	Dev	Increm loss
2015	3	???
2016	2	???

Best Method

AY	Dev	Increm loss
2000	1	5,000
2000	2	7,000
2000	3	4,000
2000	4	2,000
2001	1	4,500
2001	2	6,500
2001	3	3,500
2002	1	5,500
2002	2	7,500
2003	1	6,000

Best Model

AY	Dev	Increm loss
2015	3	3,500
2016	2	1,500

KPMG

Gradient Boosted Models (GBM)

Gradient Boosted Regression Trees

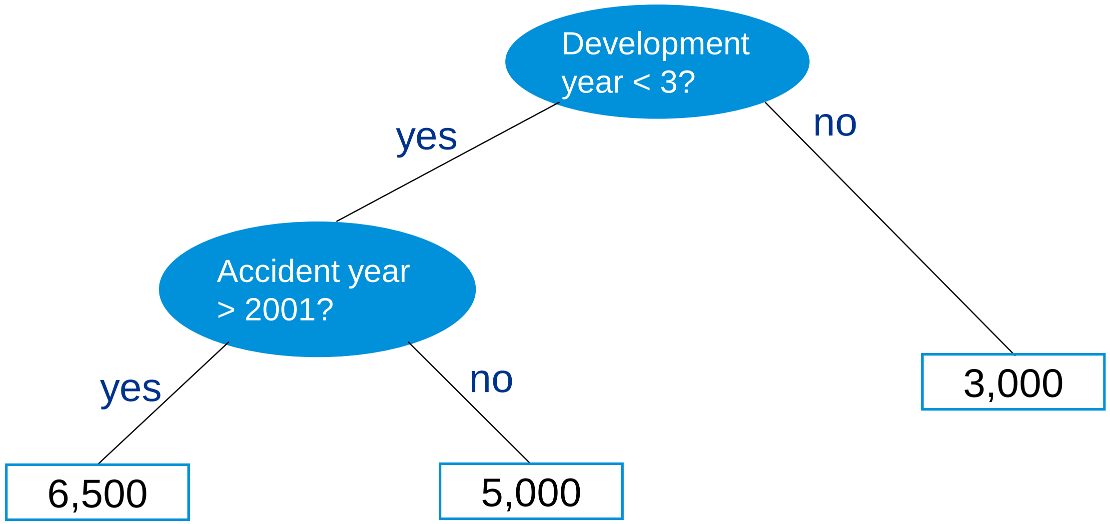
What's in a name?

Gradient Boosting	Regression Tree
• Machine learning technique that combines many weak models into a stronger model (ensembling)	• Predictive model that can be represented using a tree

First described by Friedman (2001).

Some Heuristics*

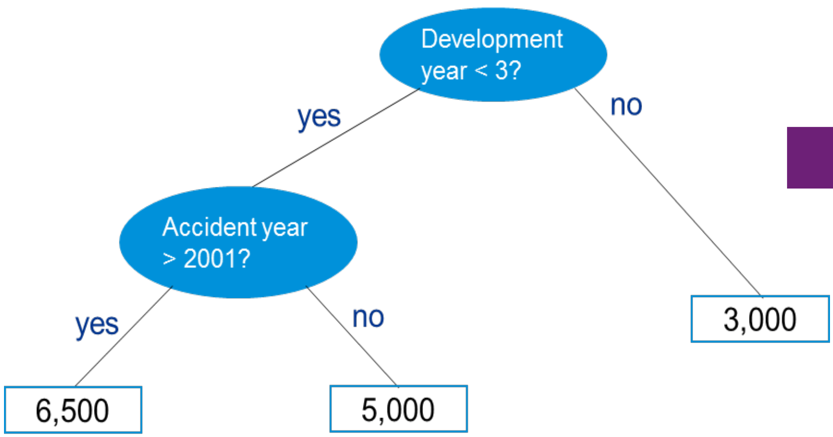
One decision tree



**via a very simplified illustrative example*

Some Heuristics

One decision tree



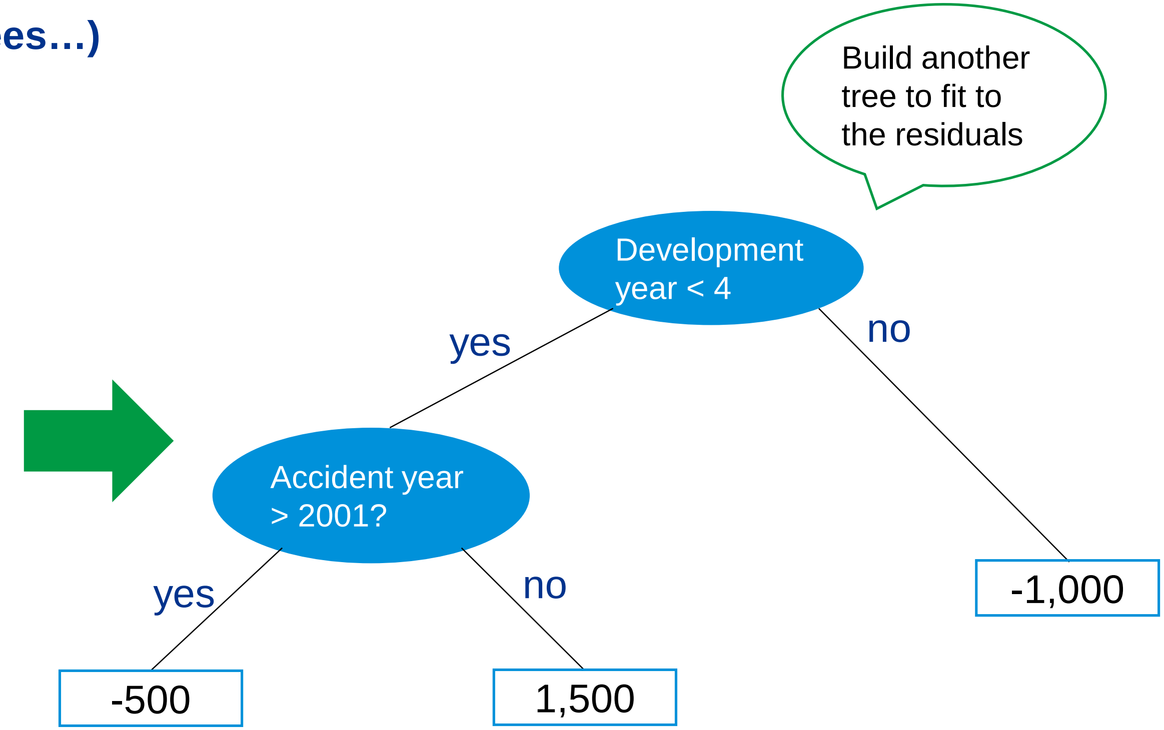
Not so great performance

AY	Dev	Increm loss	Predicted	Residual
2000	1	5,000	5,000	0
2000	2	7,000	5,000	2,000
2000	3	4,000	3,000	1,000
2000	4	2,000	3,000	-1,000
2001	1	4,500	5,000	-500
2001	2	6,500	5,000	1,500
2001	3	3,500	3,000	500
2002	1	5,500	6,500	-1,000
2002	2	7,500	6,500	1,000
2003	1	6,000	6,500	-500

Some Heuristics

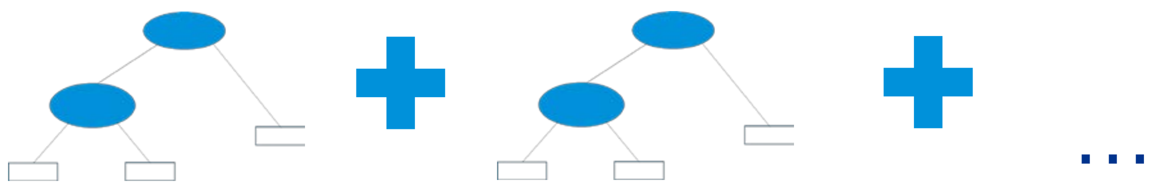
Boosting (more decision trees…)

AY	Dev	Residual
2000	1	0
2000	2	2,000
2000	3	1,000
2000	4	-1,000
2001	1	-500
2001	2	1,500
2001	3	500
2002	1	-1,000
2002	2	1,000
2003	1	-500



Some Heuristics

Boosting (more decision trees…)



AY	Dev	Residual (1st tree)	Prediction (2nd tree)	Residual (2nd tree)	…
2000	1	0	1,500	-1,500	…
2000	2	2,000	1,500	500	…
2000	3	1,000	1,500	-500	…
2000	4	-1,000	-1000	0	…
2001	1	-500	1,500	-2,000	…
2001	2	1,500	1,500	0	…
2001	3	500	1,500	-1,000	…
2002	1	-1,000	-500	-500	…
2002	2	1,000	-500	1,500	
2003	1	-500	-500	0	

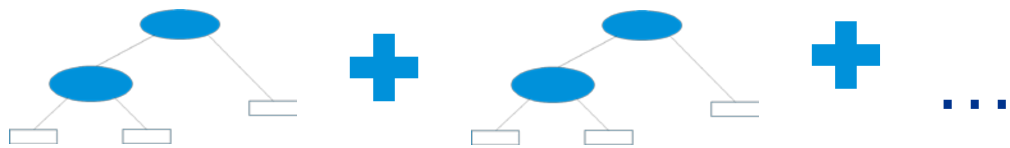
$$2{,}000 - 1{,}500 = 500$$

Each tree tries to correct the error of the previous trees. By constructing a sequence of many trees we'll have ourselves a decent model.

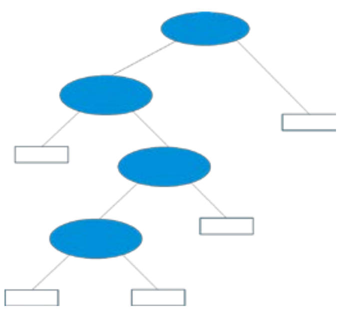
Hyperparameters & Tuning

There are many ways to specify a GBM algorithm; as examples,

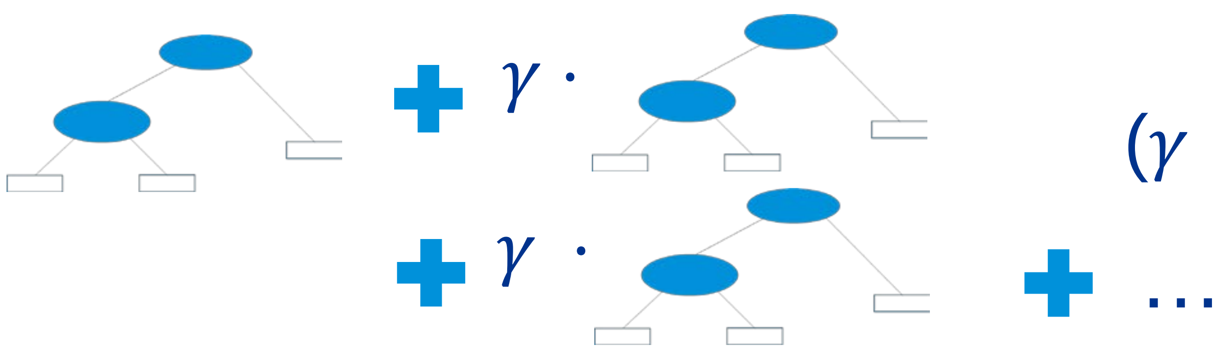
- Number of trees



- Depth of trees



- Learning rate



$(\gamma < 1)$

Hyperparameters & Tuning

There are many ways to specify a GBM algorithm

• Number of trees	• 50, 100, 200
• Depth of trees	• 1, 5, 20
• Learning rate	• 0.01, 0.1
• Sampling rate of training data	• 0.5, 0.8
• Sampling rate of predictors	• 0.5, 0.8
• …	• …

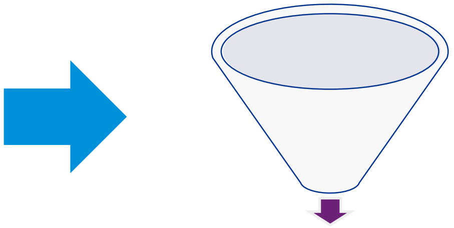
Hyperparameters & Tuning

How do we pick the best one(s)?

- Number of trees
- Depth of trees
- Learning rate
- Sampling rate of training data
- Sampling rate of predictors
- …

- 50, 100, 200
- 1, 5, 20
- 0.01, 0.1
- 0.5, 0.8
- 0.5, 0.8
- …

$3(3)(2)(2)(2) = 72$ combinations!

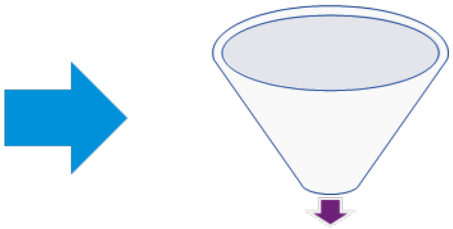


KPMG

Hyperparameters & Tuning

“Autopilot”

$3(3)(2)(2)(2) = 72$ combinations!

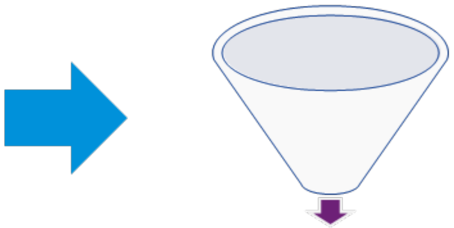


- Models are fit using each of the 72 combinations and are compared using cross-validation, the combination of hyperparameters with the lowest MSE is then fit to the total data set.

Hyperparameters & Tuning

“Autopilot”

$3(3)(2)(2)(2) = 72$ combinations!

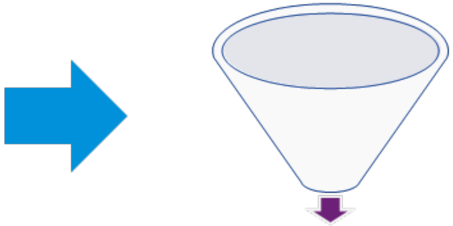


- Models are fit using each of the 72 combinations and are compared using cross-validation, the combination of hyperparameters with the lowest MSE is then fit to the total data set.
- We can feed into our funnel more than one type of algorithm. In other words, we can simultaneously test GBM, GLM, and other techniques such as Random Forests or Neural Networks, much like actuaries considering Chain Ladder and Bornhuetter-Ferguson

Hyperparameters & Tuning

“Autopilot”

$3(3)(2)(2)(2) = 72$ combinations!



- Models are fit using each of the 72 combinations and are compared using cross-validation, the combination of hyperparameters with the lowest MSE is then fit to the total data set.
- We can feed into our funnel more than one type of algorithm. In other words, we can simultaneously test GBM, GLM, and other techniques such as Random Forests or Neural Networks, much like actuaries considering Chain Ladder and Bornhuetter-Ferguson.
- Instead of building one model, we build a pipeline which generates a model on its own for subsequent review dates.

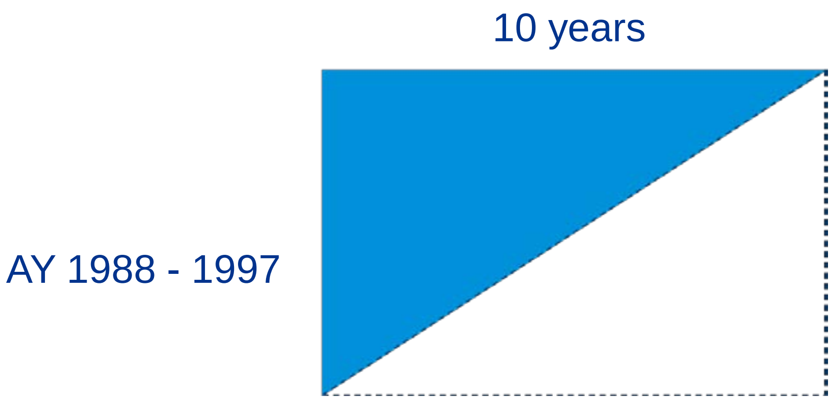
Example Pipeline on Schedule P Data

The Data

NAIC Schedule P Dataset from CAS website

http://www.casact.org/research/index.cfm?fa=loss_reserves_data

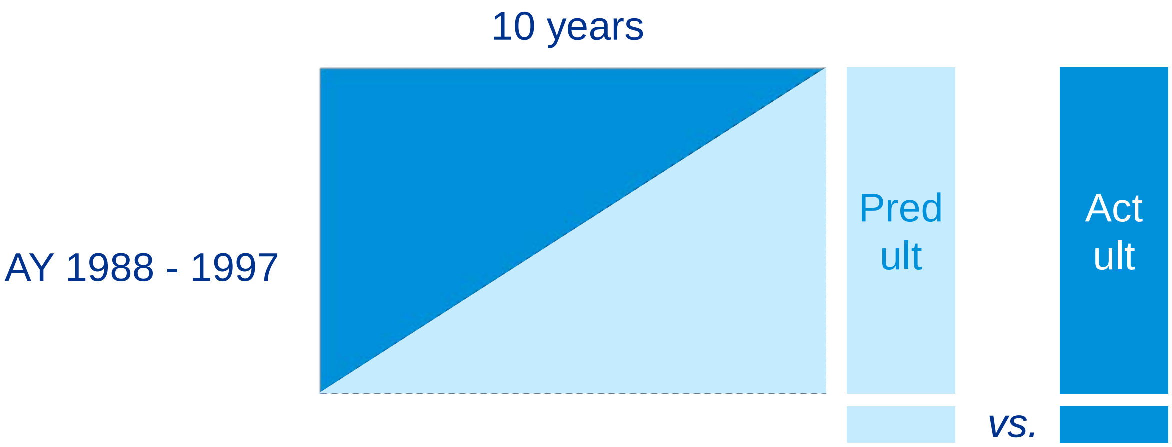
"[D]ata set that contains run-off triangles of six lines of business for all U.S. property casualty insurers. The triangle data correspond to claims of accident year 1988 – 1997 with 10 years development lag. Both upper and lower triangles are included so that one could use the data to develop a model and then test its performance retrospectively"



B Private passenger auto liability/medical
C commercial auto/truck liability/medical
D Workers' compensation
F2 Medical malpractice - Claims made
H1 Other liability - Occurrence
R1 Products liability - Occurrence

The Task

Squaring the triangle



Predict the unpaid losses to calculate ultimate losses and compare to actual ultimates

Pipeline Specs

What did we put in?

Response: Incremental paid loss divided by premium for the AY

Predictors: Accident year, development period, premium, initial paid-to-incurred ratio, initial case reserves, max/min/avg of incremental paid loss ratio for the development period, max/min/avg paid-to-incurred ratio for the development period, initial paid loss divided by premium

ML technique: GBM

- Distribution: gamma
- Number of trees: {50, 100, 150}
- Learning rate: {0.01, 0.05}
- Maximum depth: {1, 10}
- Column sample rate: {0.8, 0.1}
- Sample rate: {0.8, 1}
- Hyperparameters tuned using random 2-fold cross validation

KPMG

Pipeline Specs

Randomness

AY	Dev	Var_1	Var_2	Var_3	...	Increm loss
2000	1					5,000
2000	2					7,000
2000	3					4,000
2001	1					6,000
...	...					...

Actually a random variable!

In other words, applying the same <u>method</u> to the same dataset may give us a different <u>model</u> each time. Analogously, different actuaries may pick different development factors from the same triangle. This is a feature, not a bug.

ML technique: GBM

- Distribution: gamma
- Number of trees: {50, 100, 150}
- Learning rate: {0.01, 0.05}
- Maximum depth: {1, 10}
- Column sample rate: {0.8, 0.1}
- Sample rate: {0.8, 1}
- Hyperparameters tuned using random 2-fold cross validation

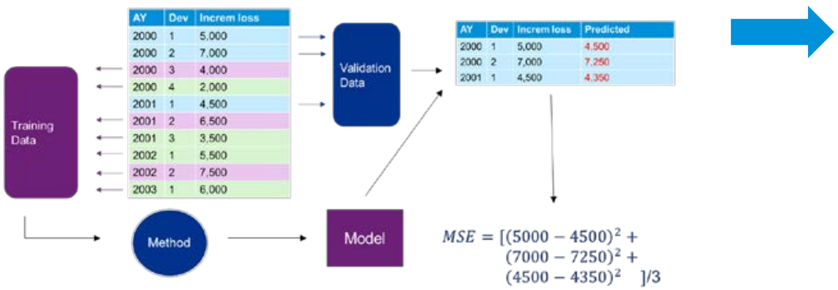
Sources of randomness

Pipeline Specs

Variance reduction: Model averaging (bagging)

Model/Method selection with Cross Validation

Use the method to train a model on the training set, then predict losses on the validation set



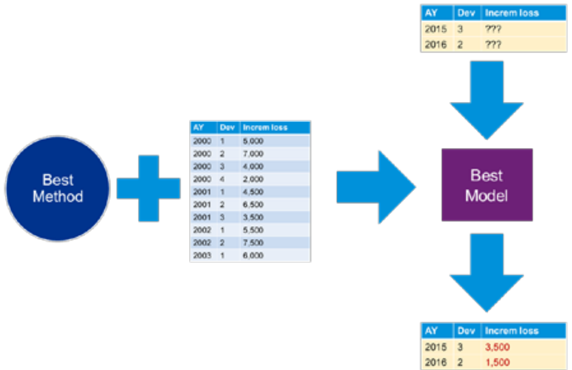
AY	Dev	Increm loss
2000	1	5,000
2000	2	7,000
2000	3	4,000
2000	4	2,000
2001	1	4,500
2001	2	6,500
2001	3	3,500
2002	1	5,500
2002	2	7,500
2003	1	6,000

AY	Dev	Increm loss	Predicted
2000	1	5,000	4,500
2000	2	7,000	7,250
2001	1	4,500	4,350

$$MSE = [(5000-4500)^2 + (7000-7250)^2 + (4500-4350)^2 \quad]/3$$

Using the best method/model for prediction

Once the "best method" has been selected from cross validation, it can be trained on the full dataset to obtain the "best model" for prediction



AY	Dev	Increm loss
2015	3	???
2016	2	???

AY	Dev	Increm loss
2000	1	5,000
2000	2	7,000
2000	3	4,000
2000	4	2,000
2001	1	4,500
2001	2	6,500
2001	3	3,500
2002	1	5,500
2002	2	7,500
2003	1	6,000

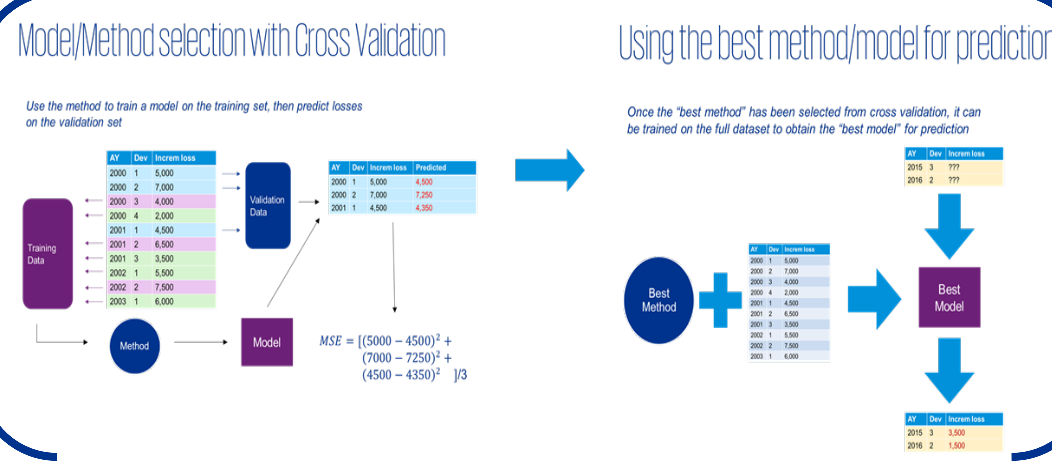
AY	Dev	Increm loss
2015	3	3,500
2016	2	1,500

We're really doing this "meta" procedure a bunch of times then averaging the results…

Pipeline Specs

Model averaging (bagging)

Final Model* $= \frac{1}{N} \sum_{i=1}^{N} \Big($



$\Big)_i$

** Can use median, too*

Performance Results

Performance Results

Aggregate error metrics

LOB	CSR RMSE	GBM RMSE	CSR MAE	GBM MAE
Commercial Auto	6,143	12,889	2,981	4,392
Other Liability	38,924	35,138	11,936	6,875
Personal Auto	122,498	284,322	32,357	53,358
Workers' Comp	35,884	42,996	13,020	16,433

- RMSE is root-mean-square error and MAE is mean absolute error – lower is better.
- CSR refers to the Changing Settlement Rate MCMC model as described in Meyers (2015).
- The metrics for each LOB are aggregated over 50 triangles from different companies. Errors are calculated from the actual and predicted ultimates for each triangle.
- The performance results are comparable.

Performance Results

Select companies – largest ultimates by LOB

LOB	Group Code	Outcome	CSR Estimate	CSR error	GBM Estimate	GBM error
PA	1767	91,360,195	90,601,540	-1%	93,345,986	2%
PA	2003	12,393,224	12,099,970	-2%	12,618,131	2%
PA	4839	3,027,062	3,014,489	0%	3,012,017	0%
PA	7080	1,459,916	1,709,068	17%	1,540,233	6%
CA	1767	2,226,624	2,229,021	0%	2,305,929	4%
CA	388	745,997	737,324	-1%	736,222	-1%
CA	2135	525,310	533,568	2%	527,793	0%
CA	2623	472,426	451,021	-5%	438,426	-7%
OL	1767	2,190,615	2,368,310	8%	2,436,935	11%
OL	620	439,839	414,199	-6%	416,901	-5%
OL	2003	272,941	362,530	33%	280,099	3%
OL	5185	141,013	144,031	2%	143,095	1%
WC	7080	1,836,596	1,801,781	-2%	1,946,800	6%
WC	1767	1,742,600	1,692,375	-3%	1,788,922	3%
WC	86	1,611,800	1,804,628	12%	1,769,656	10%
WC	388	1,233,553	1,094,143	-11%	1,023,739	-17%

Considerations & Extensions

Considerations and Extensions

- Applying these methods to claim level data.
  - ML algorithms were designed with “big” data in mind, not “triangles” with 55 data points!
- ML methods focus on point prediction accuracy – how do we arrive at measures of uncertainty and ranges?
- How can we peek into the “black box”?
- How do we account for tail development?

Q&A

References

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Meyers, G. (2015). Stochastic Loss Reserving Using Bayesian MCMC Models. *CAS Monograph Series*, (1).

kpmg.com/socialmedia

The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavor to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.