AKUR8

Penalized regression - Between Credibility and GBMs

Low-Exposure Levels

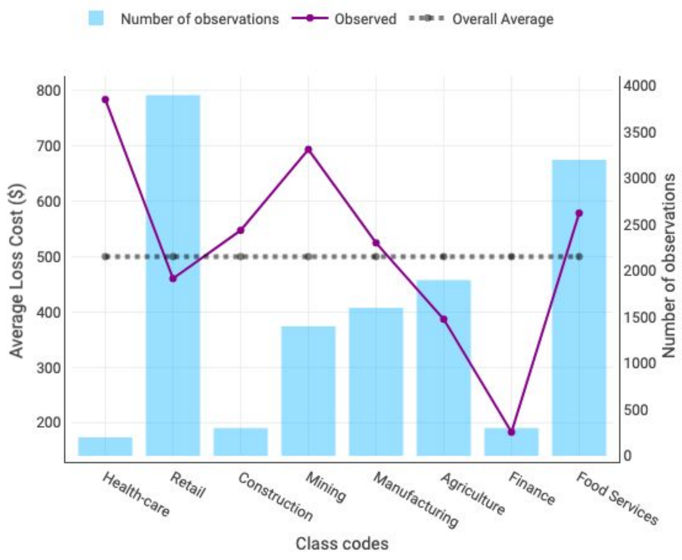
Credibility: Worker’s Compensation example

Loss Cost by class code example

Losses and exposures for companies are collected, and we want to compute an estimation of the average loss cost per class code.

The data **can be represented visually**:

- The **blue bars** represent the number of observations for a given class;
- The **purple lines** represent the **Observed Experience** as the average loss cost for each class;
- The **black line** represent the **overall average** (or grand average) of $500 in this example.



Credibility: Univariate estimate

A natural estimate is the average loss cost by class code.

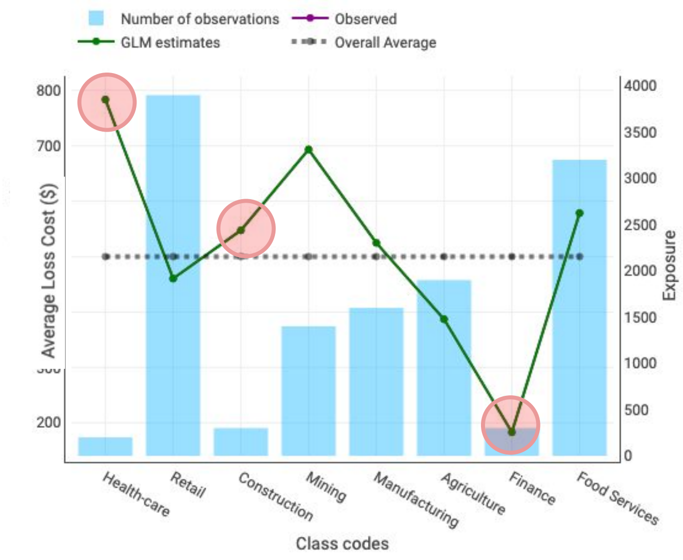
Such estimate may be inappropriate for class Health-Care which has low exposure.

The same argument applies for Finance and Construction

This approach is followed in the GLM framework, that fully trusts the data:

$$\beta_{\mathrm{GLM}} = \max_{\beta} \mathrm{LogLikelihood}(\mathrm{Observed}, \beta)$$

estimate of loss cost by class code



Removing non-significant levels

Removing low-significance levels

A classic approach is to use the **statistical significance** of the different levels.

Levels that have low exposure (or small effects) are grouped together, or put at the average value.

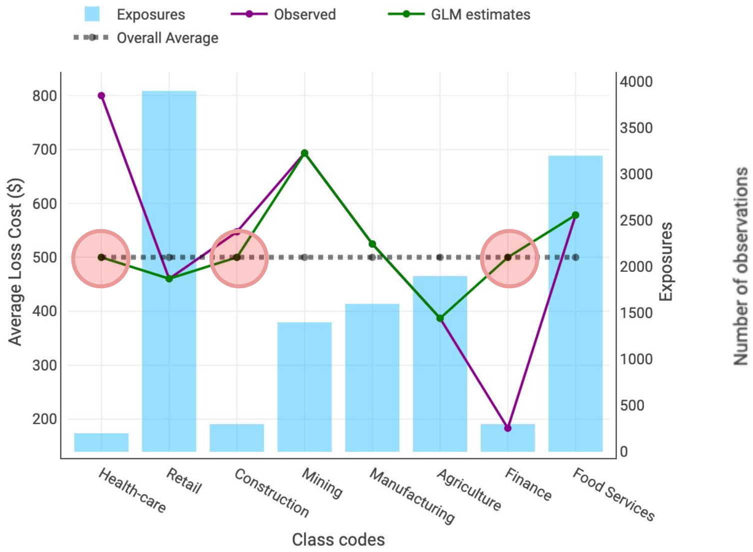
The goal of this approach is to avoid trusting very noisy models with a few observations.

The result obtained will depend on the **significance threshold** above which levels will be kept into the final model or grouped:

- If a level is **more significant** than the threshold, it is **kept**
- If a level is **less significant** than the threshold, it is **removed**

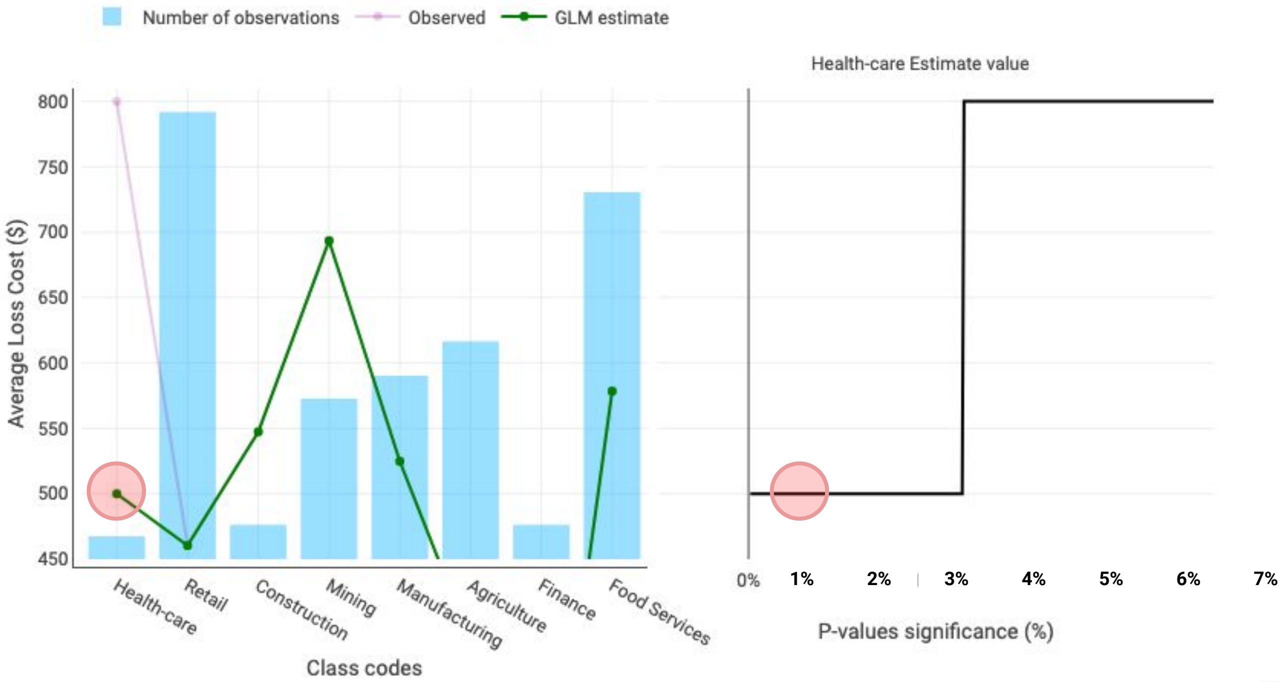
Modelers often use a “5% significance level” but any other value can be selected.

estimate of loss cost by class code



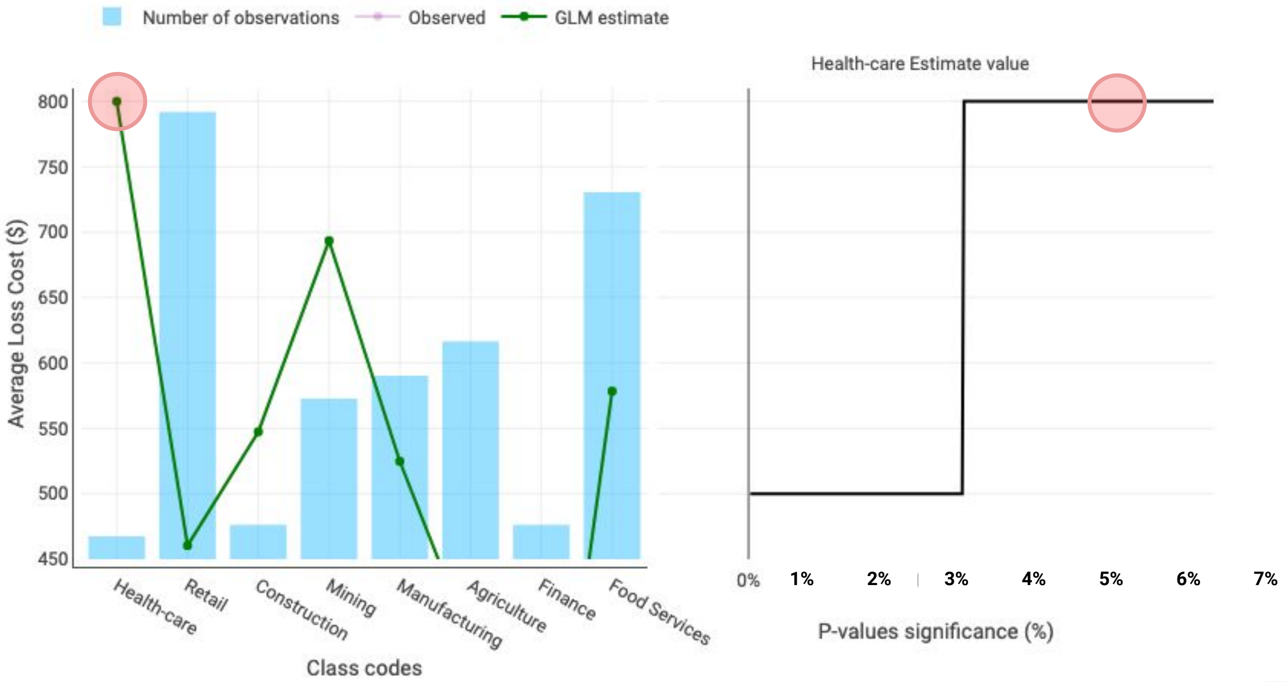
Fitted model depends on the threshold

Strong (low) significance thresholds are hard to validate and lead to a **robust** model.



Fitted model depends on the threshold

Weak (high) significance threshold are easy to validate and lead to a **volatile** model.



Strengths & limits of levels selection

This approach has well know strengths and limits:

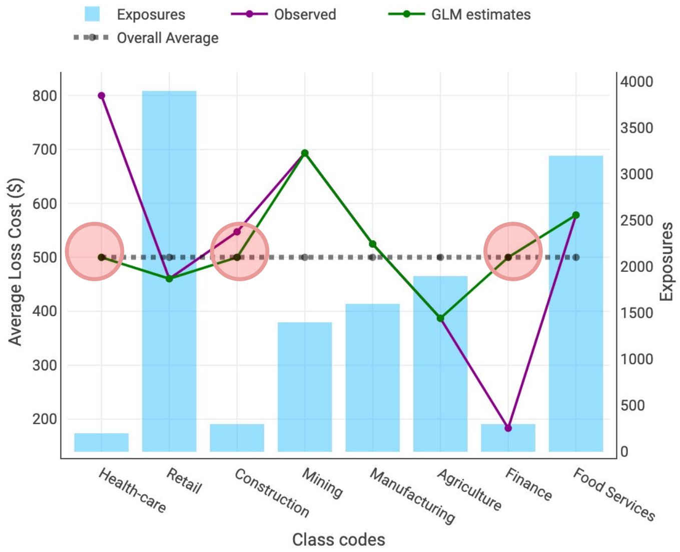
✅ It is a binary method, leading to clear decisions

✅ It is very frequently used and widely accepted

✅ It relies on very classic statistics

❌ It is a binary method: it does not use efficiently the limited observations we have on “health-care”

❌ Tests justification rely on hypothesis often not met in practice

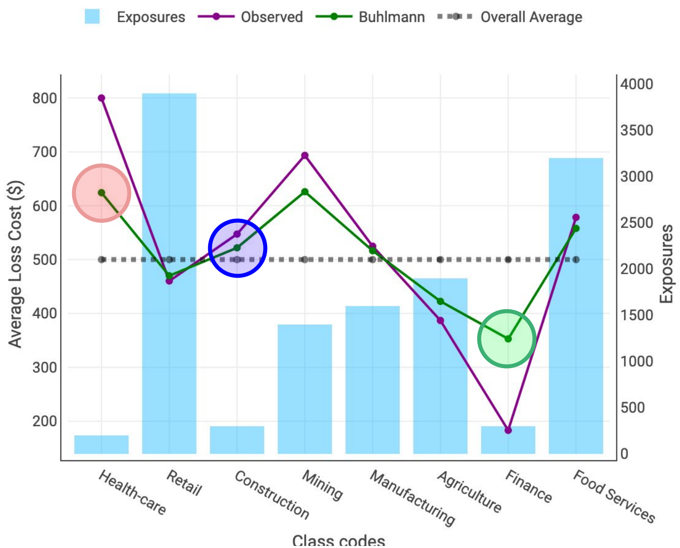


Credibility

The Credibility solution

The idea of a credibility framework is to create a model between these two extreme “yes” and “no” solutions. Low-exposure levels are:

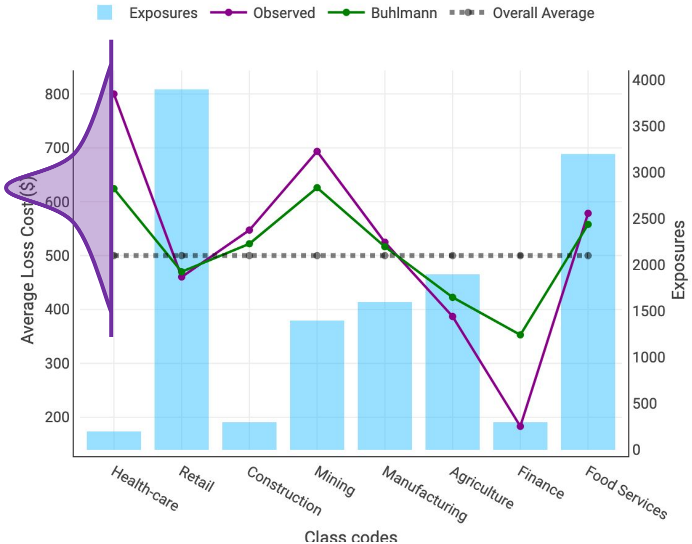
- **Not fully trusted** (like they would in a standard GLM framework).
- **Not fully discarded** (like they would if we applied a grouping of non-significant levels).



What is the idea motivating Credibility ?

The Buhlmann credibility creates predictions by mixing two sources of informations:

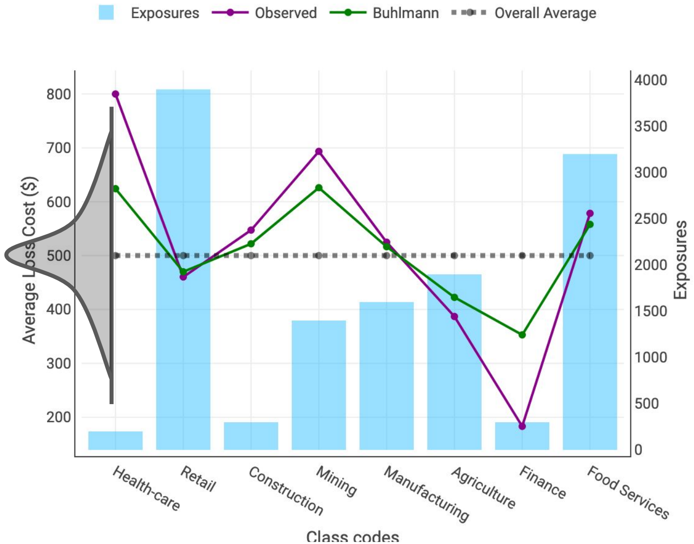
- **The “pure GLM” predictions, centered on the observed values**
- The “a-priori” distribution of the observations, centered on the grand-average



What is the idea motivating Credibility ?

The Buhlman credibility creates predictions by mixing two sources of informations:

- The “pure GLM” predictions, centered on the observed values
- **The “a-priori” distribution of the observations, centered on the grand-average**



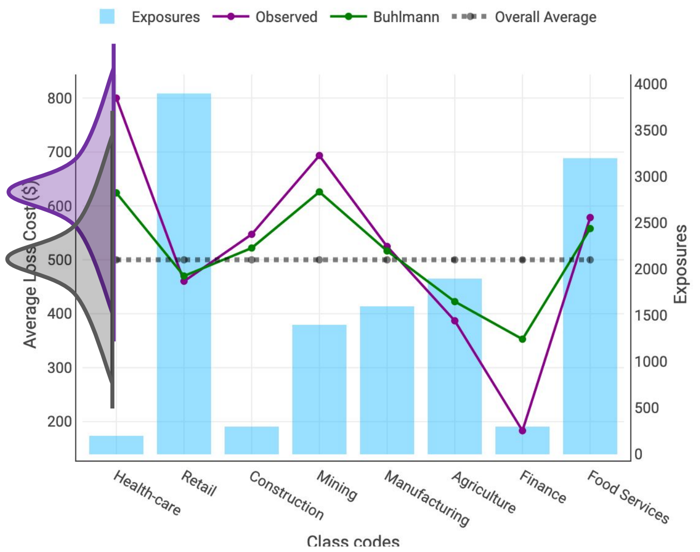
What is the idea motivating Credibility ?

The Buhlman credibility creates predictions by mixing two sources of informations:

- The “pure GLM” predictions, centered on the observed values
- The “a-priori” distribution of the observations, centered on the grand-average

More data means the observed values vary less around the predictions, meaning they can be trusted: a **strong weight** is given to **the observed values**.

Less data means the observed values vary a lot around the predictions, meaning they can’t be trusted: a **strong weight** is given to the **a-priori (grand average)**.



Quick Reminder… What is Credibility

"Credibility, simply put, is the weighting together of different estimates to come up with a combined estimate."

Foundations of Casualty Actuarial Science

When the volume of data is not enough to accurately estimate the losses, Credibility methodologies provide ways to **complement the observed experience with additional information**.

The Credibility formula is:

Estimate = Z * **Observed Experience** + (1 – Z) * Complement of Credibility

where the Credibility factor **Z** is a number between 0 and 1.

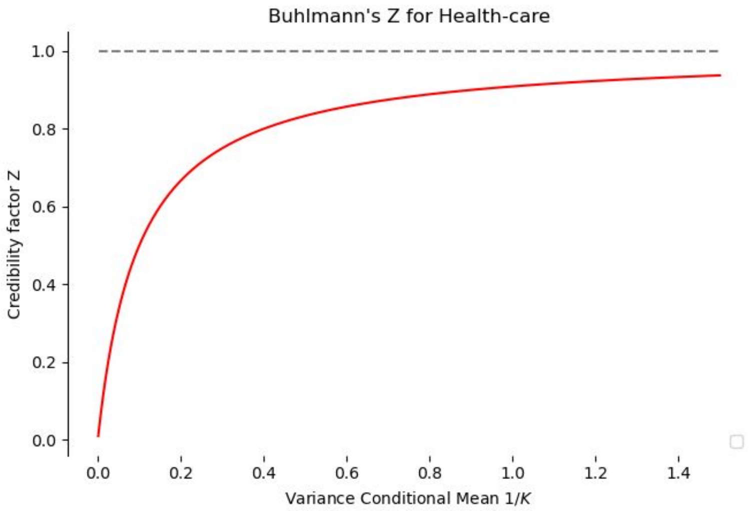
Buhlmann Credibility: Computing Z

The modeler decides to use Buhlmann Credibility.

The formula for credibility is

$$Z = \frac{n}{n+K}$$

Where K can be estimated from the data via standard formulas.



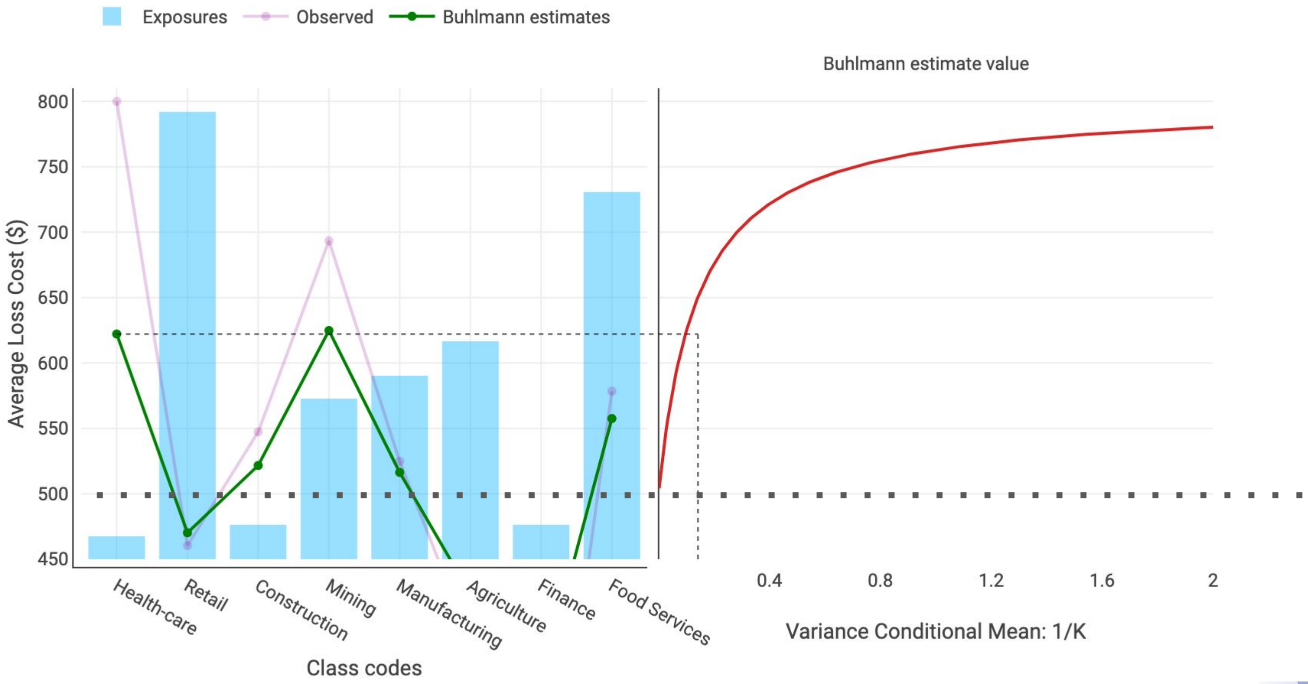
[1] K in R^+ is the ratio between the variances of the two distributions presented earlier: mean of conditional variance (in purple, Expected Process Variance, EPV) / variance of conditional means (in grey, Variance of the Hypothetical Mean, VHM)

Example: Health Care estimate

Large K (**low credibility**)

Weak information on the predictions can be derived from the observations (the distributions of the observations around the prediction has a large variance)

Predictions are **close to the overall average**.

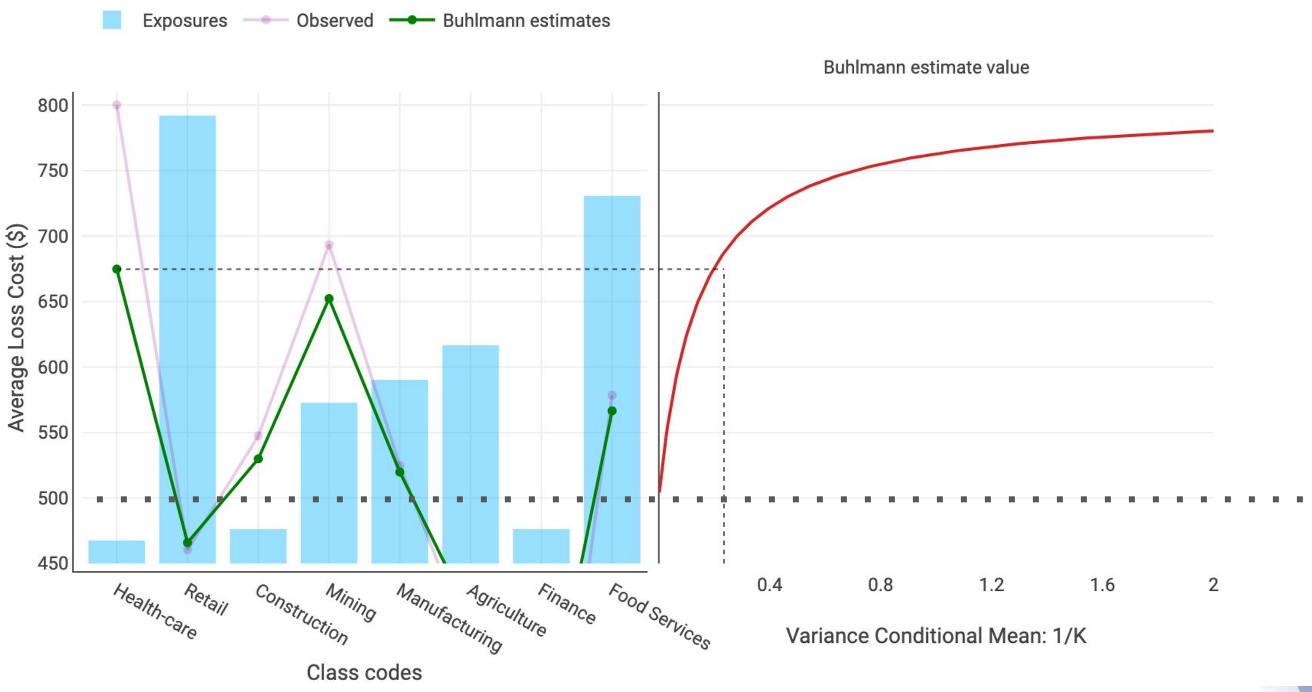


Example: Health Care estimate

Medium K (**intermediate credibility**)

Intermediate information on the predictions can be derived from the observation (the distributions of the observations around the prediction has a medium variance)

Predictions are **between the overall average and the observations**.

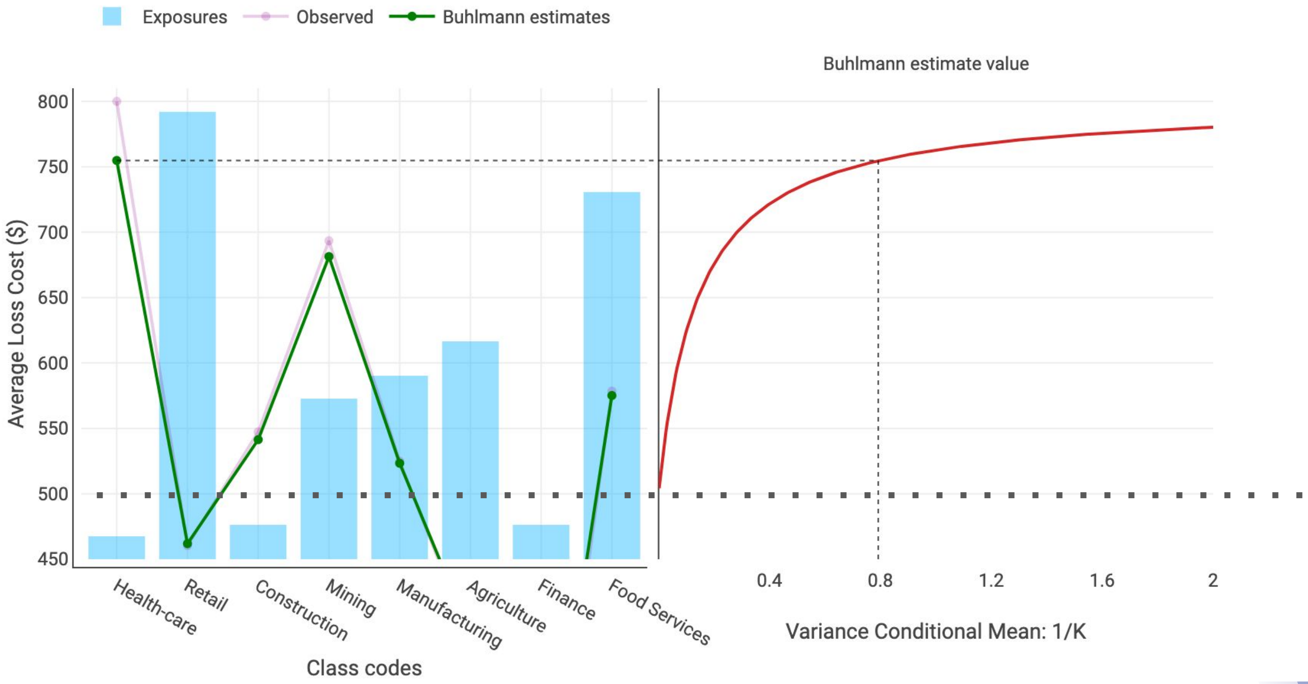


Example: Health Care estimate

Small K (**strong credibility**)

Strong information on the predictions can be derived from the observation (the distributions of the observations around the prediction has a small variance)

Predictions are **close to the observations.**

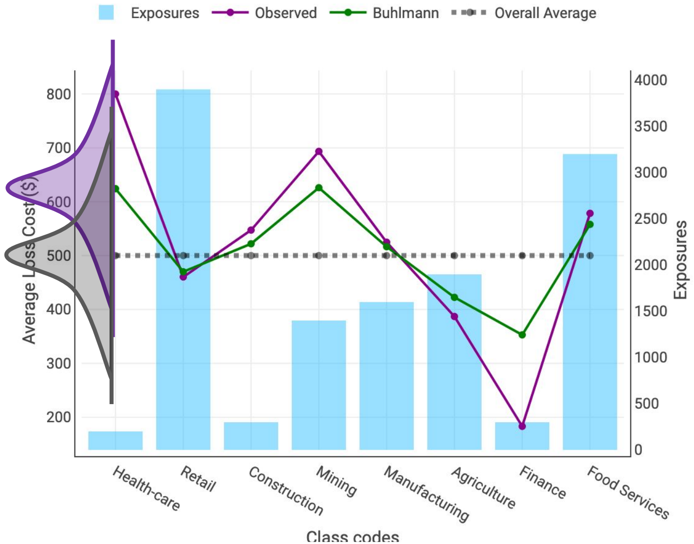


Credibility works on a single dimension !

Credibility hypothesis are on the observed values, not the coefficients !

Integration of credibility is done as a post-processing, after the GLM has been built.

It can be applied to a single variable: it is not a multivariate analysis !



The statisticians who designed our GLMs were unaware we intended to subject GLM estimates to the violence of a subsequent round of ad hoc credibility adjustments. If they had known, they might have suggested a better starting point than GLM estimates..”

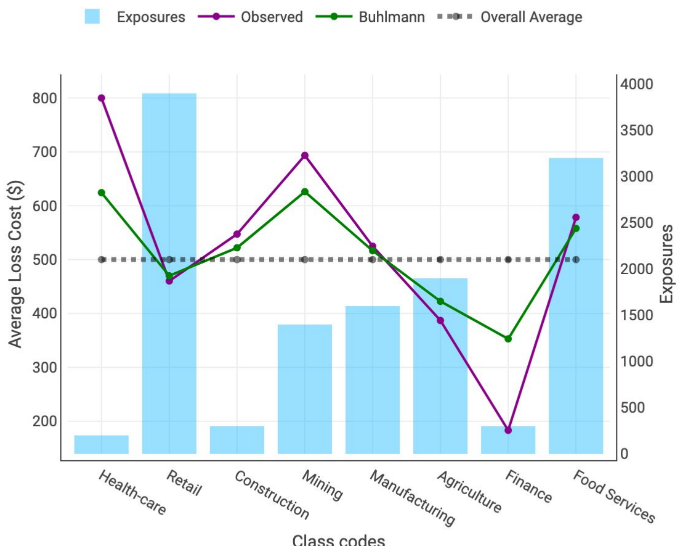
F. Klinker, Generalized Linear Mixed Models for Ratemaking 2010

Strengths & limits of Buhlmann Credibility

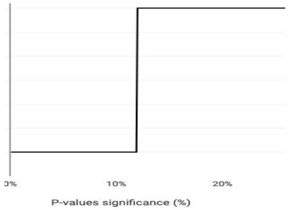
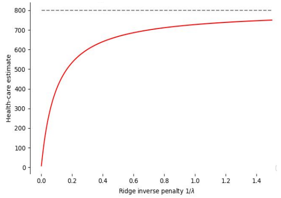
This approach has also well-documented strengths & limits:

- ✅ It allows to leverage all the available data,
- ✅ It is very frequently used and widely accepted
- ✅ It relies on very classic statistics
- ✅Results can be computed without a computer (which didn't exist in the 1960's when the method was proposed)

- ❌ It is applied as a post-processing, only to one variable in the model



Comparing different techniques

<table>
<tr><th></th><th>Levels Selection</th><th>Credibility</th></tr>
<tr><td>Control low-exposure segments to prevent overfitting</td><td colspan="2">All the techniques presented today aim at controlling overfitting.</td></tr>
<tr><td>Set coefficients of low-exposure segments at zero</td><td>Selection of effects</td><td>No selection of effects</td></tr>
<tr><td>Shrink low-exposure segments</td><td>No</td><td>This allows to tolerate segments with limited (yet usable) data</td></tr>
<tr><td>Works for multivariate models</td><td>Yes</td><td>No</td></tr>
<tr><td>Creates transparent models (GLM or additive models)</td><td colspan="2">Designed for the GLM framework</td></tr>
<tr><td>Natively manage non-linear effects</td><td colspan="2">These techniques work on “pure GLM” (linear & categorical effects)</td></tr>
<tr><td>Coefficient depending on the robustness parameter</td><td>
</td><td>
</td></tr>
</table>

Enriching the GLM framework

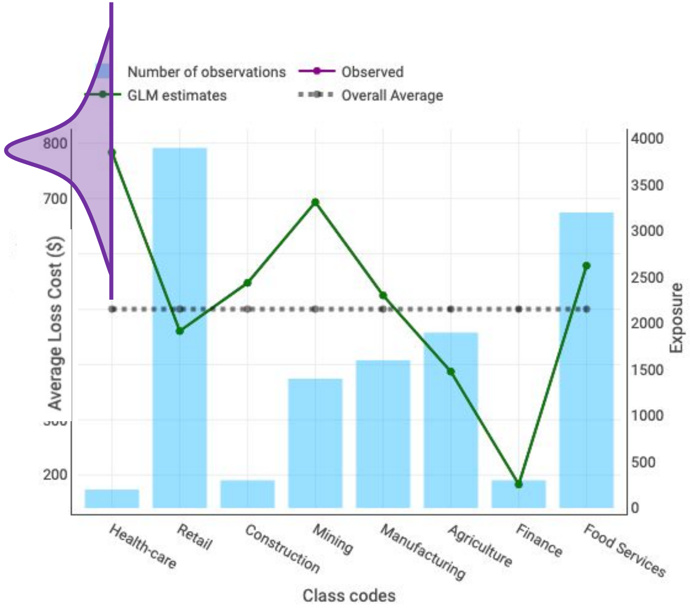
Why the GLM lacks credibility

GLM coefficients are the **maximum of likelihood** (probability of observing the data, given the model):

$$\beta^* = Argmax\ Likelihood(Obs., \beta)$$

The probability of observations is displayed in purple on the right.

.



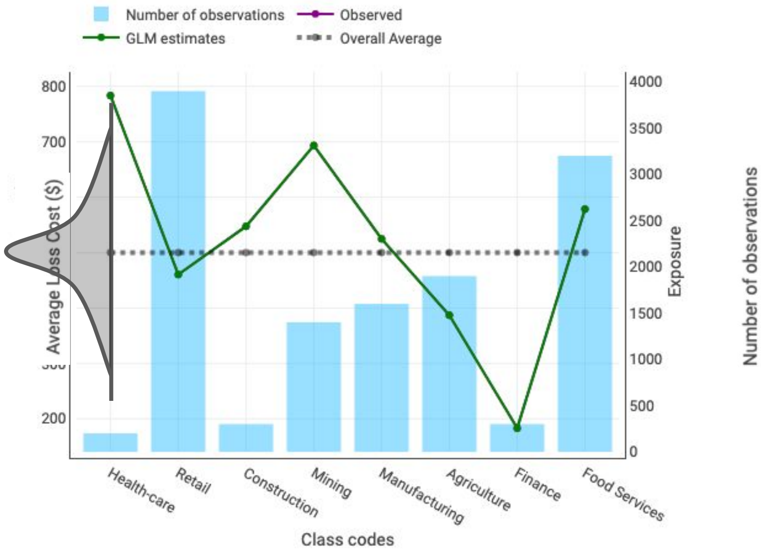
The Penalized GLM Formula

Like for Credibility, Penalized Regressions **integrate another prior hypothesis**.

But this time, **the prior hypothesis is directly on the coefficient** values: we integrate a probability for different values of the coefficients.

For instance, in the Ridge-regression framework, we assume coefficients follow a normal distribution:

$$\beta \sim N\left(0, {}^{1}/_{\lambda}\right)$$



The Penalized GLM Formula

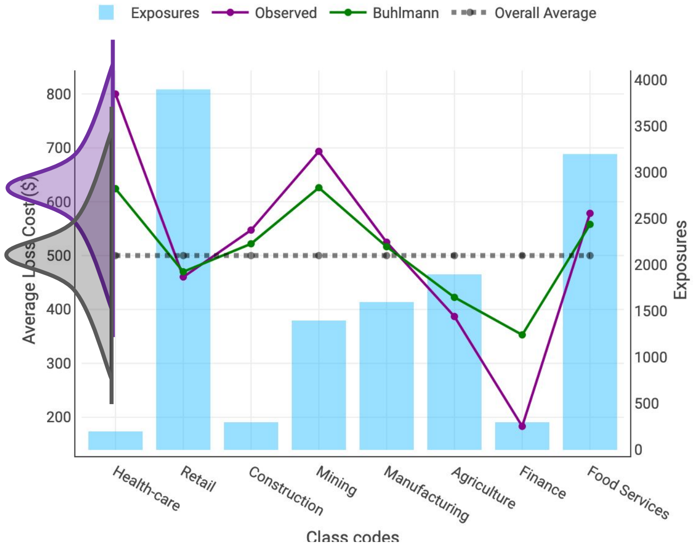
The idea of Penalized Regression is to include a second hypothesis in the GLM framework: the coefficients have a a-priori distribution.

This prior is visible in the maximum of likelihood definition:

$$\beta^* = Argmax\ Likelihood(Obs., \beta) \times \alpha\ e^{\frac{-\beta^2}{1/\lambda}}$$

Which means:

$$\beta^* = Argmax\ LogLikelihood(Obs., \beta) - \lambda\ \beta^2$$



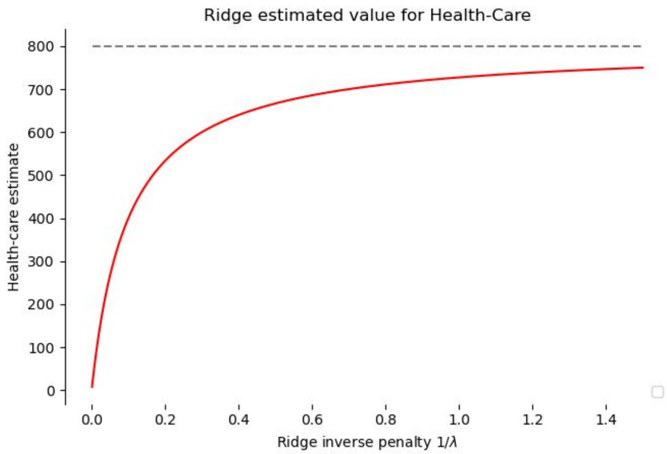
This hypothesis is **similar to the Buhlmann credibility** ; they are equivalent for a one-dimensional model.

The Ridge

The coefficients computed depend on the λ parameter.

- For small lambda, the coefficients will be close to a simple GLM.
- For large lambda, the coefficients will be close to zero (and the predictions will be close to the base-level).

$$\beta^* = Argmax\ LogLikelihood(Obs., \beta) - \lambda\ \beta^2$$



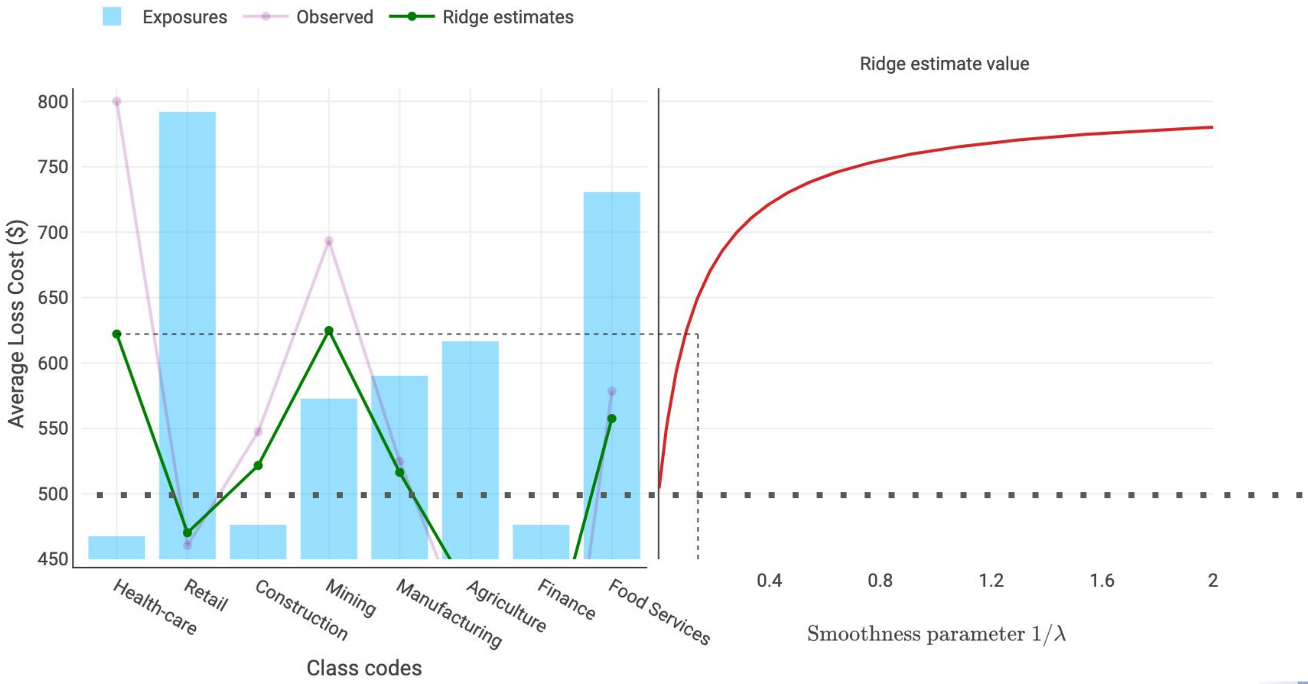
1/Lambda

Example: Health Care estimate

Large λ (**large penalty**)

Strong prior on the coefficient (the prior distribution has a small variance)

Coefficients and predictions are **close to the overall average**.

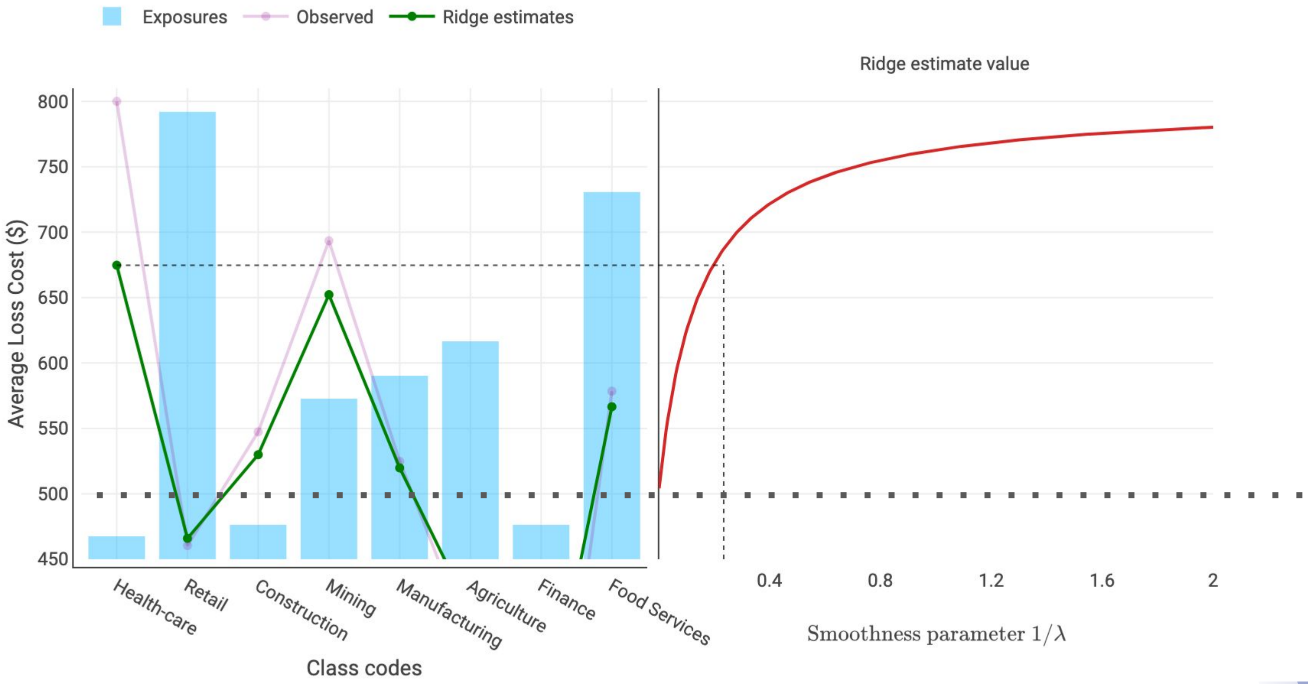


Example: Health Care estimate

Medium λ (**medium penalty**)

Intermediate prior on the coefficient (the prior distribution has a small variance)

Coefficients and predictions are **further to the overall average**.

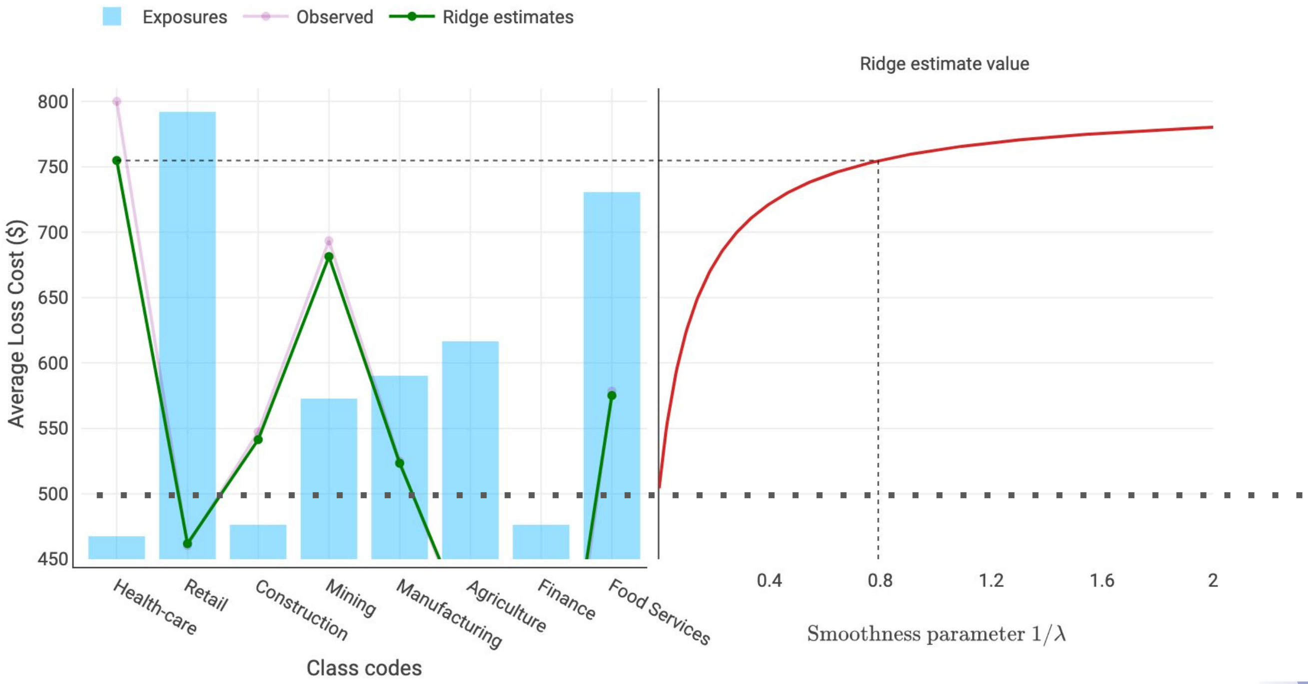


Example: Health Care estimate

Small λ (**small penalty**)

Weak prior on the coefficient (the prior distribution has a large variance)

Coefficients and predictions are **close to the observed value**.



Blending GLM with Credibility

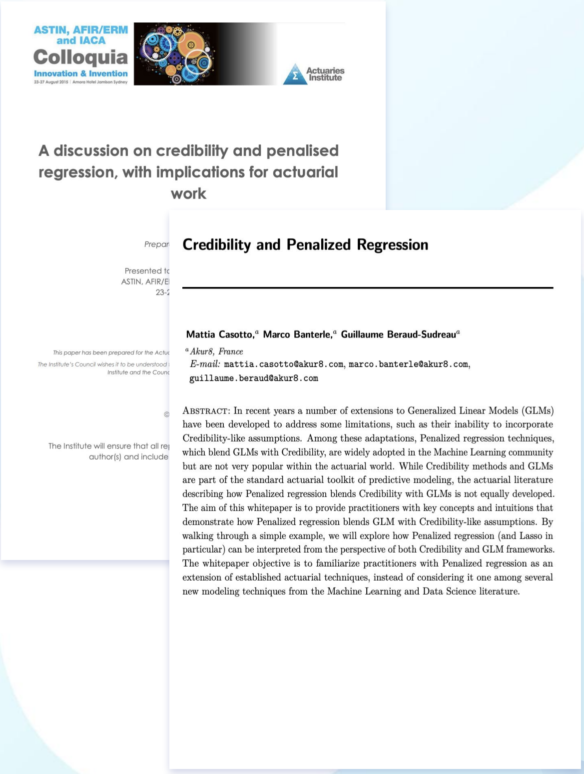
Penalized GLMs share the same properties as **Credibility** in the following ways:

1. Both **shrink** GLM estimates toward the complement of Credibility (grand average);
2. Both apply **more shrinkage** to segments with **low volume** of data / credibility
3. Both based on a **Bayesian model**, as in Buhlmann Credibility

   The theoretical connection between Credibility and Penalized GLM can be found in:

   - Fry, Taylor. "A discussion on credibility and penalised regression, with implications for actuarial work" (2015)
   - M.Casotto et al. "Credibility and Penalized Regression" (2022) ; this topic was also presented last year during the CAS seminar.

4. However, while the Credibility approach can be applied to one variable after the GLM fit, the ridge regression can be applied to all variables simultaneously.



ASTIN, AFIR/ERM and IACA Colloquia Innovation & Invention

Actuaries Institute

A discussion on credibility and penalised regression, with implications for actuarial work

Credibility and Penalized Regression

Mattia Casotto,[a] Marco Banterle,[a] Guillaume Beraud-Sudreau[a]

[a] Akur8, France

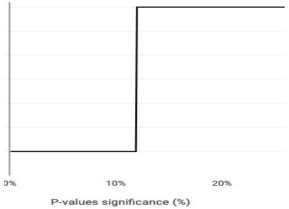
E-mail: mattia.casotto@akur8.com, marco.banterle@akur8.com, guillaume.beraud@akur8.com

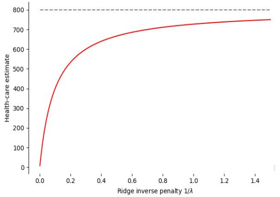
Abstract: In recent years a number of extensions to Generalized Linear Models (GLMs) have been developed to address some limitations, such as their inability to incorporate Credibility-like assumptions. Among these adaptations, Penalized regression techniques, which blend GLMs with Credibility, are widely adopted in the Machine Learning community but are not very popular within the actuarial world. While Credibility methods and GLMs are part of the standard actuarial toolkit of predictive modeling, the actuarial literature describing how Penalized regression blends Credibility with GLMs is not equally developed. The aim of this whitepaper is to provide practitioners with key concepts and intuitions that demonstrate how Penalized regression blends GLM with Credibility-like assumptions. By walking through a simple example, we will explore how Penalized regression (and Lasso in particular) can be interpreted from the perspective of both Credibility and GLM frameworks. The whitepaper objective is to familiarize practitioners with Penalized regression as an extension of established actuarial techniques, instead of considering it one among several new modeling techniques from the Machine Learning and Data Science literature.

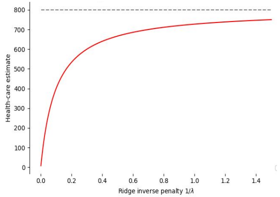
Comparing different techniques

	Levels Selection	Credibility	Ridge Regression
Control low-exposure segments to prevent overfitting	All the techniques presented today aim at controlling overfitting.		
Set coefficients of low-exposure segments at zero	Selection of effects	No selection of effects	
Shrink low-exposure segments	No	This allows to tolerate segments with limited (yet usable) data	
Works for multivariate models	Yes	No	Yes
Creates transparent models (GLM or additive models)	Designed for the GLM framework		
Natively manage non-linear effects	These techniques work on “pure GLM” (linear & categorical effects)		

Coefficient depending on the robustness parameter





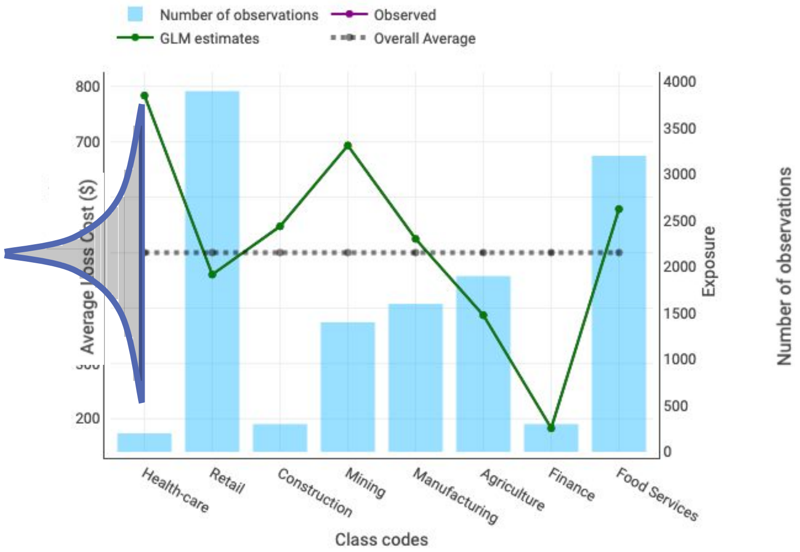


The Penalized GLM Formula: the Lasso

Like the Ridge, Lasso-regression framework, assumes coefficients follow a given distribution.

But this time the distribution used is the Laplace distribution:

$$\beta \sim Laplace\left(0, {}^{1}/_{\lambda}\right)$$



The Penalized GLM Formula

Ridge-regression also includes a second hypothesis in the GLM framework: the coefficients a-priori follow the Laplace distribution.

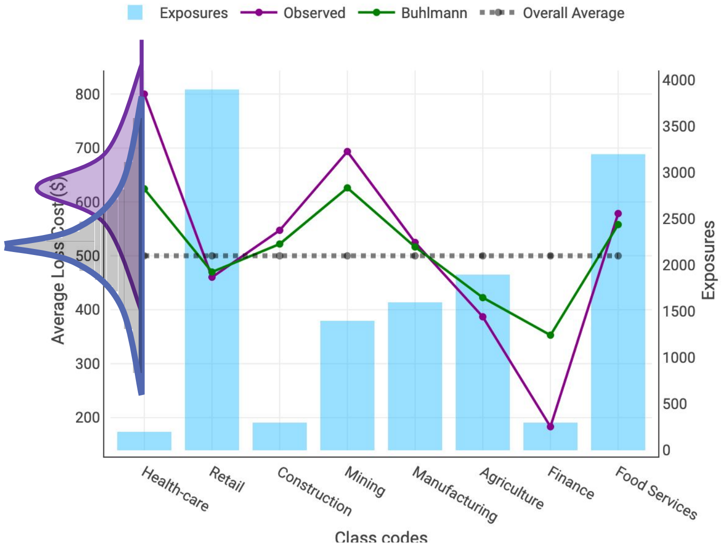
This prior is included in the maximum of likelihood definition:

$$\beta^* = Argmax\ Likelihood(Obs., \beta) \times \alpha\ e^{\frac{-|\beta|}{1/\lambda}}$$

Which means:

$$\beta^* = Argmax\ LogLikelihood(Obs., \beta) - \lambda|\beta|$$

This is very similar to the Ridge regression (and the credibility), but the distribution used is different. Here it is very "pointy" (coefficients have a high probability of being exactly zero).



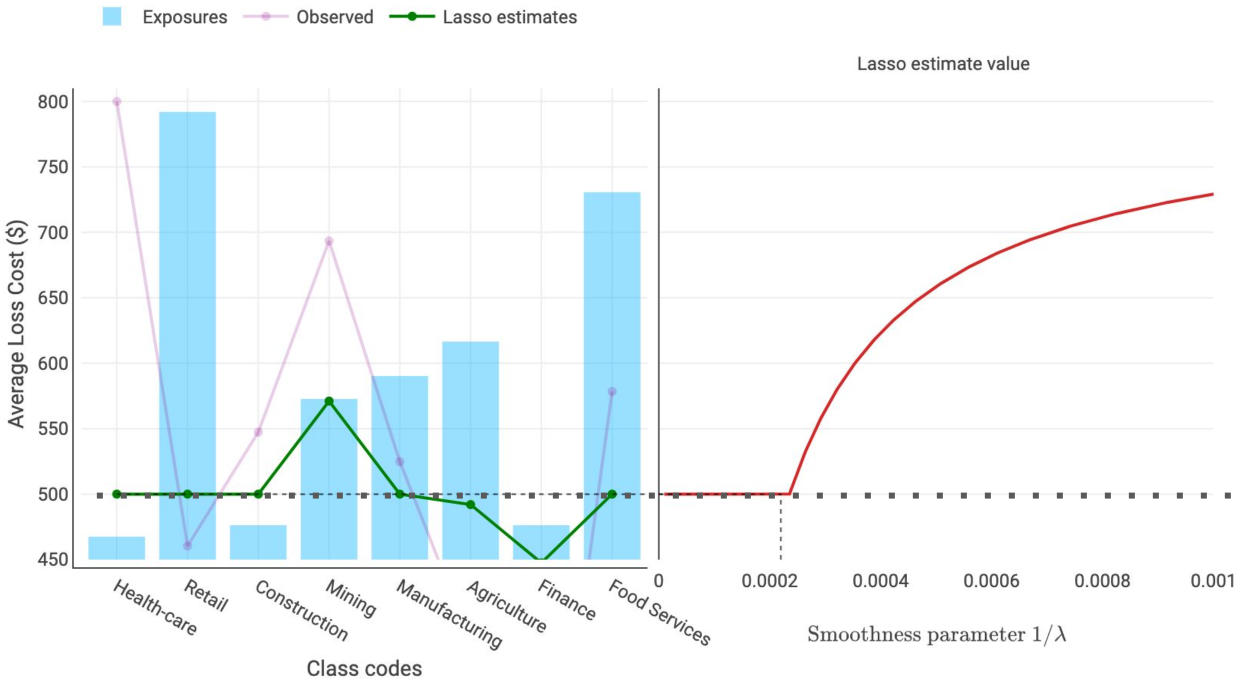
Impact of smoothness to Lasso estimates

Workers Compensation example

Large λ (**large penalty**)

Strong prior on the coefficient (the prior distribution has a small variance)

Coefficients and predictions are **close to the overall average**.



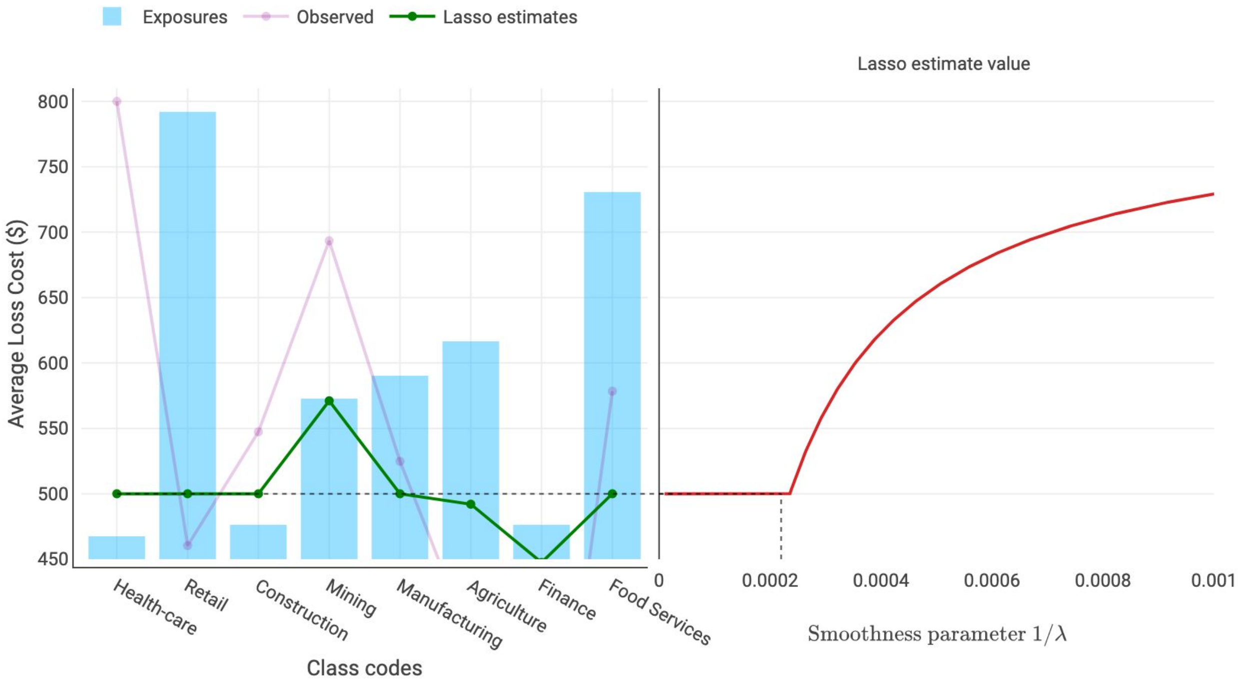
Impact of smoothness to Lasso estimates

Workers Compensation example

Large λ (**large penalty**)

Strong prior on the coefficient (the prior distribution has a small variance)

Coefficients and predictions are **close to the overall average**.



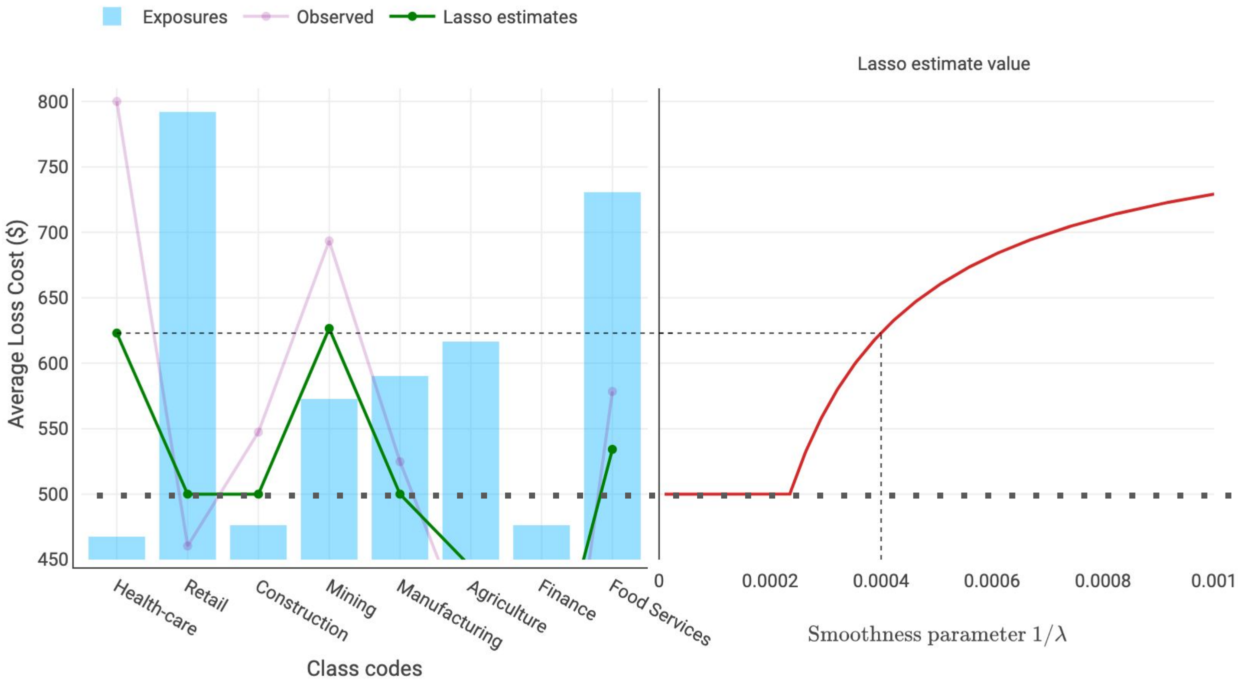
Impact of smoothness to Lasso estimates

Workers Compensation example

Medium λ (**medium penalty**)

Intermediate prior on the coefficient (the prior distribution has a small variance)

Coefficients and predictions are **further to the overall average**.



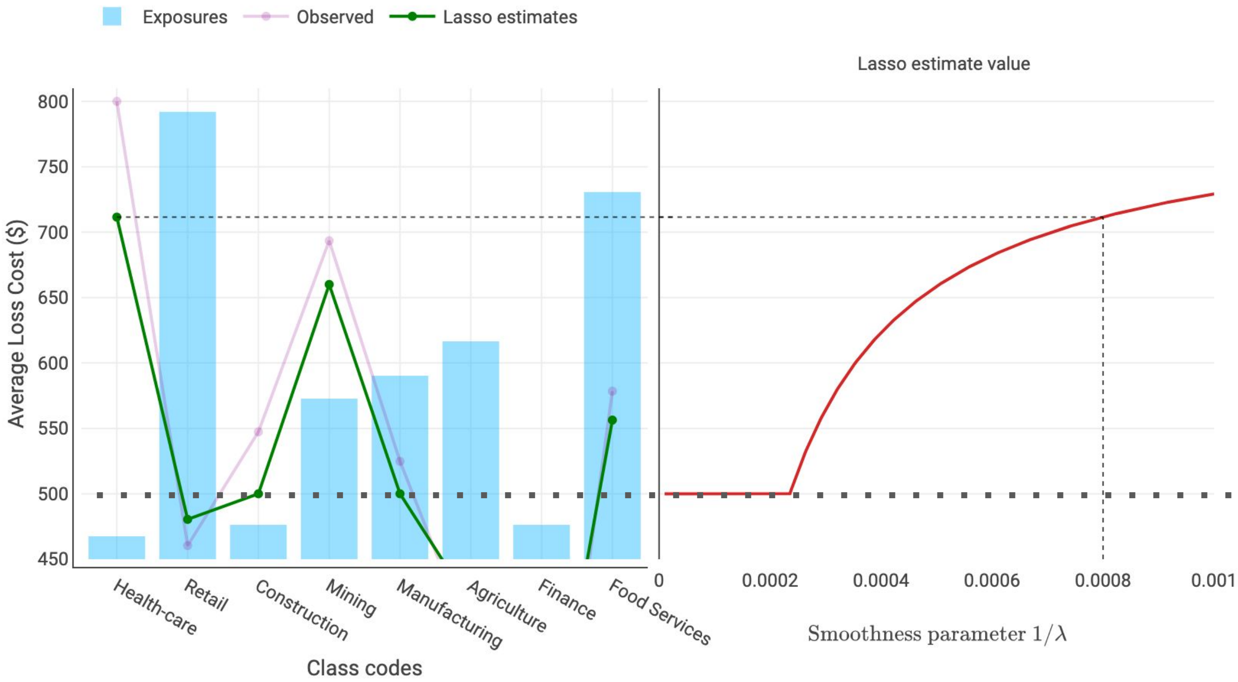
Impact of smoothness to Lasso estimates

Workers Compensation example

Small λ (**small penalty**)

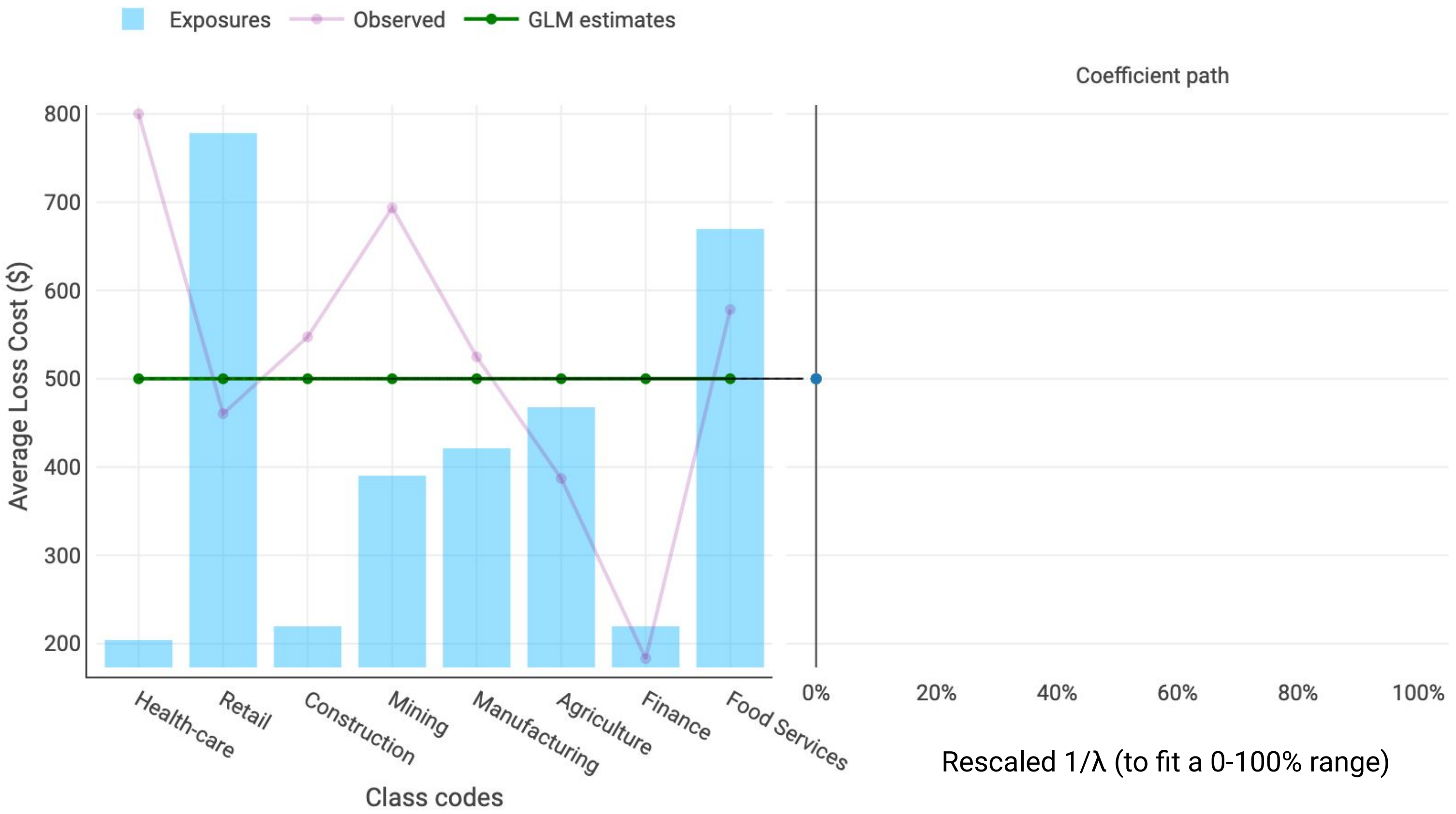
Weak prior on the coefficient (the prior distribution has a large variance)

Coefficients and predictions are **close to the observed value**.



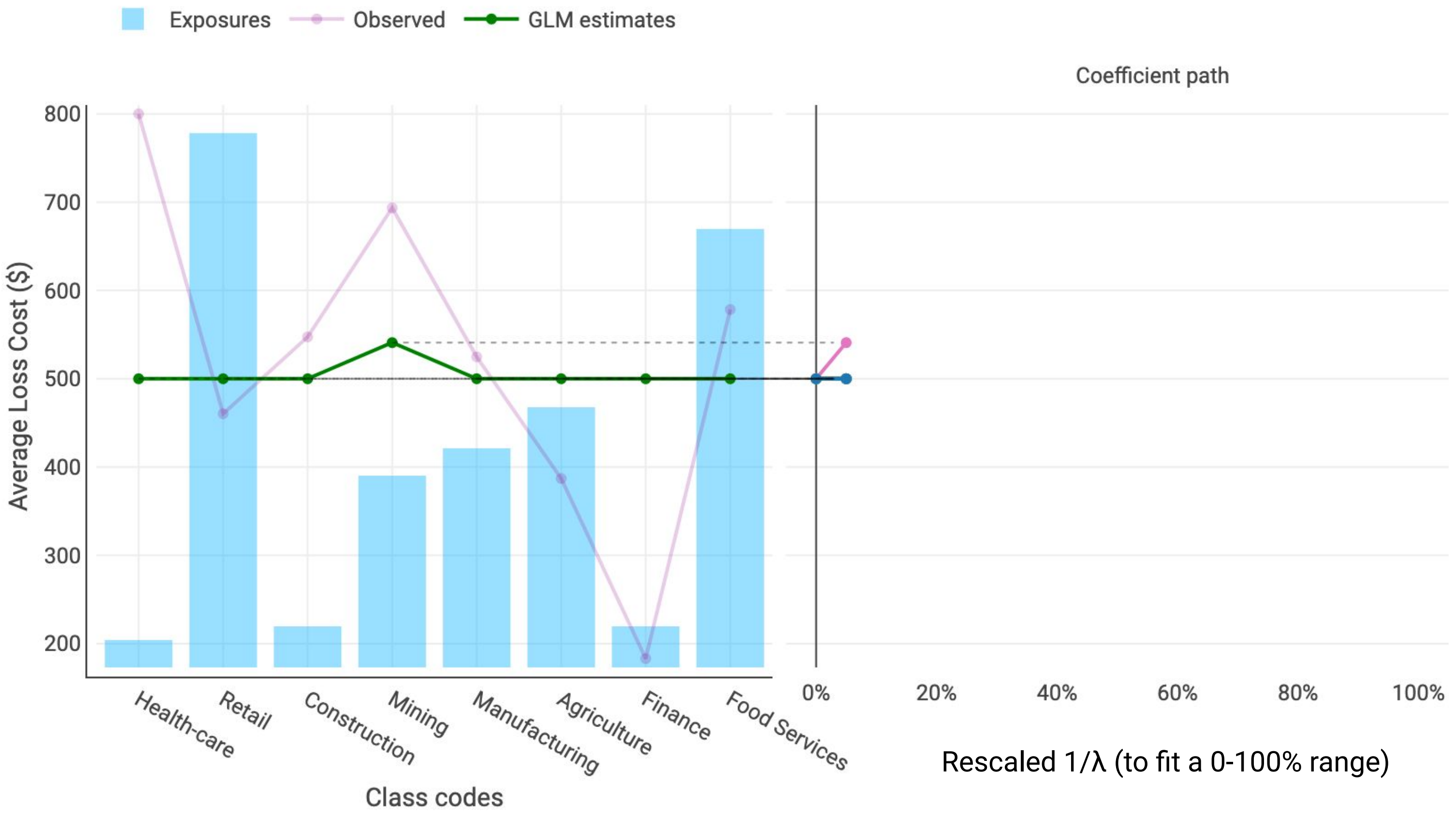
Coefficient path graph of the Lasso

Workers Compensation example



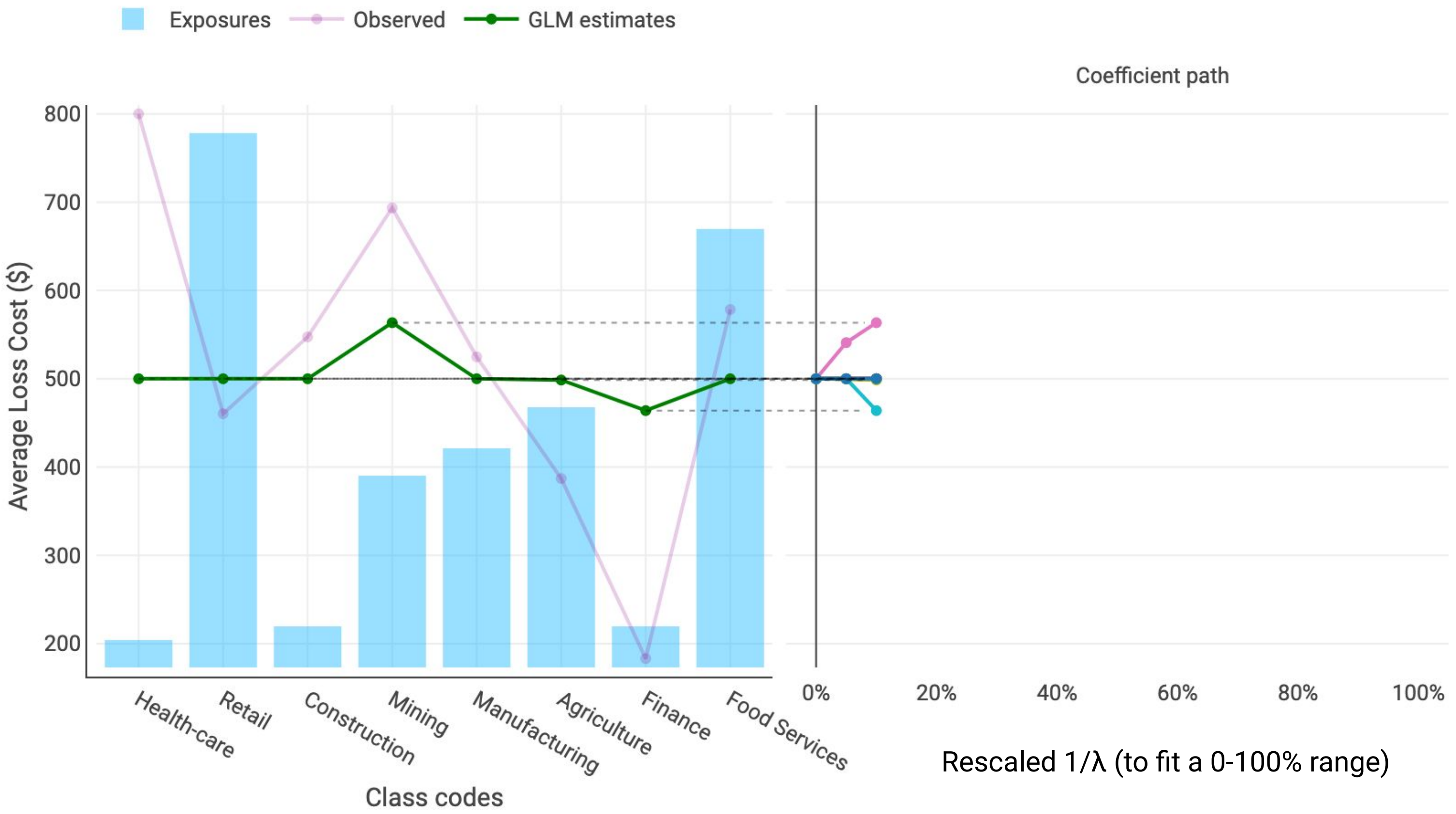
Coefficient path graph of the Lasso

Workers Compensation example



Coefficient path graph of the Lasso

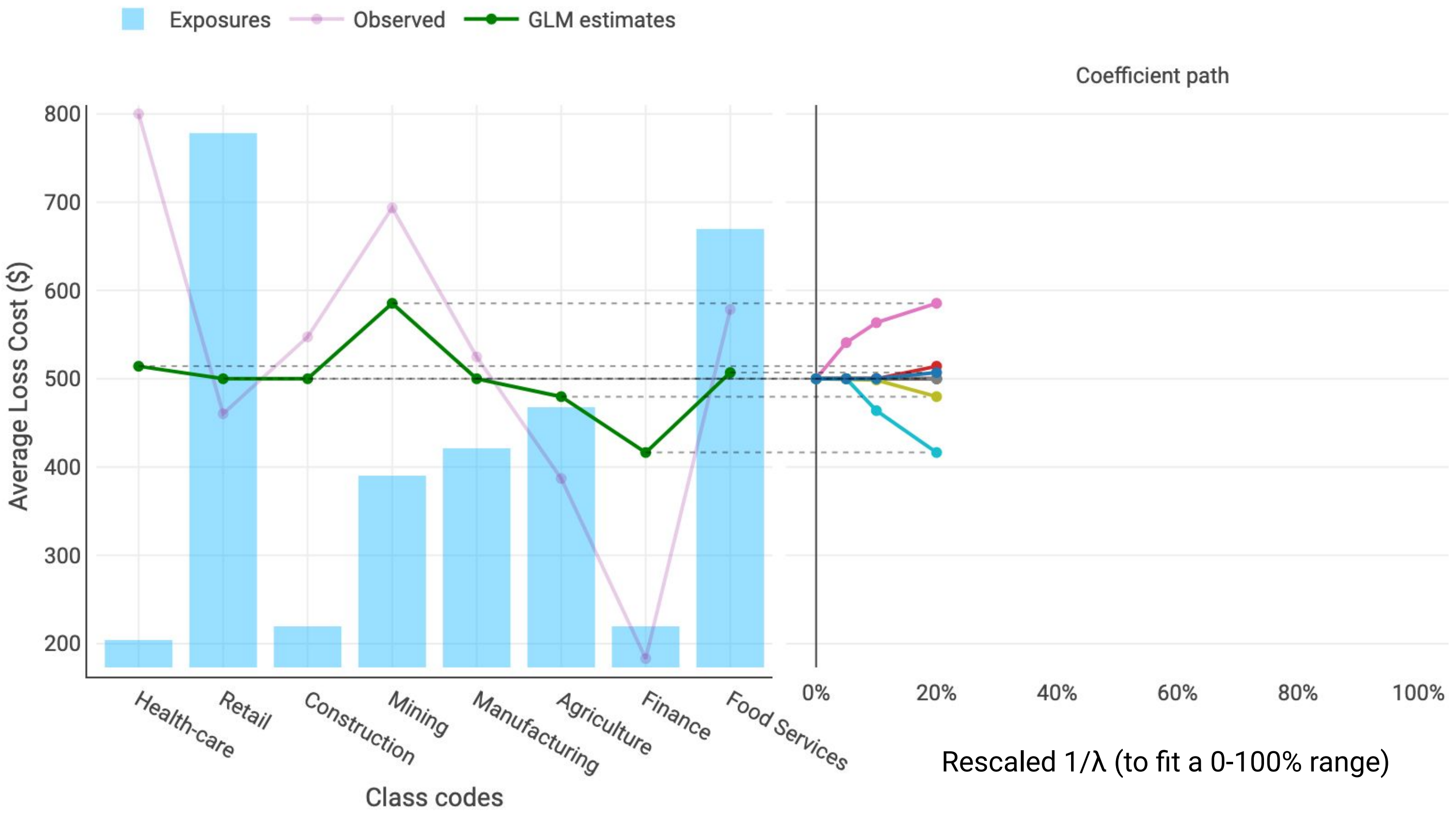
Workers Compensation example



Rescaled 1/λ (to fit a 0-100% range)

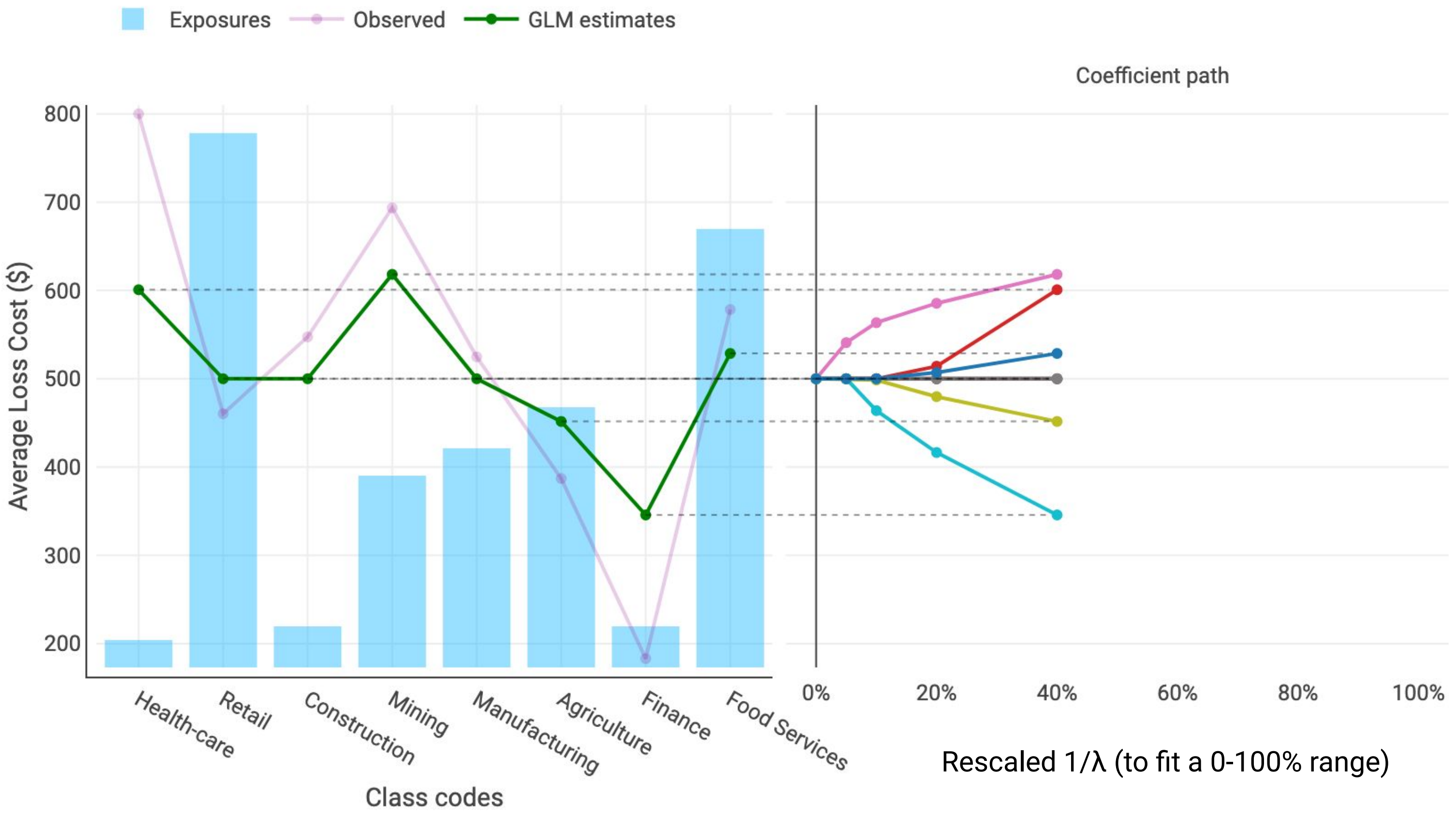
Coefficient path graph of the Lasso

Workers Compensation example



Coefficient path graph of the Lasso

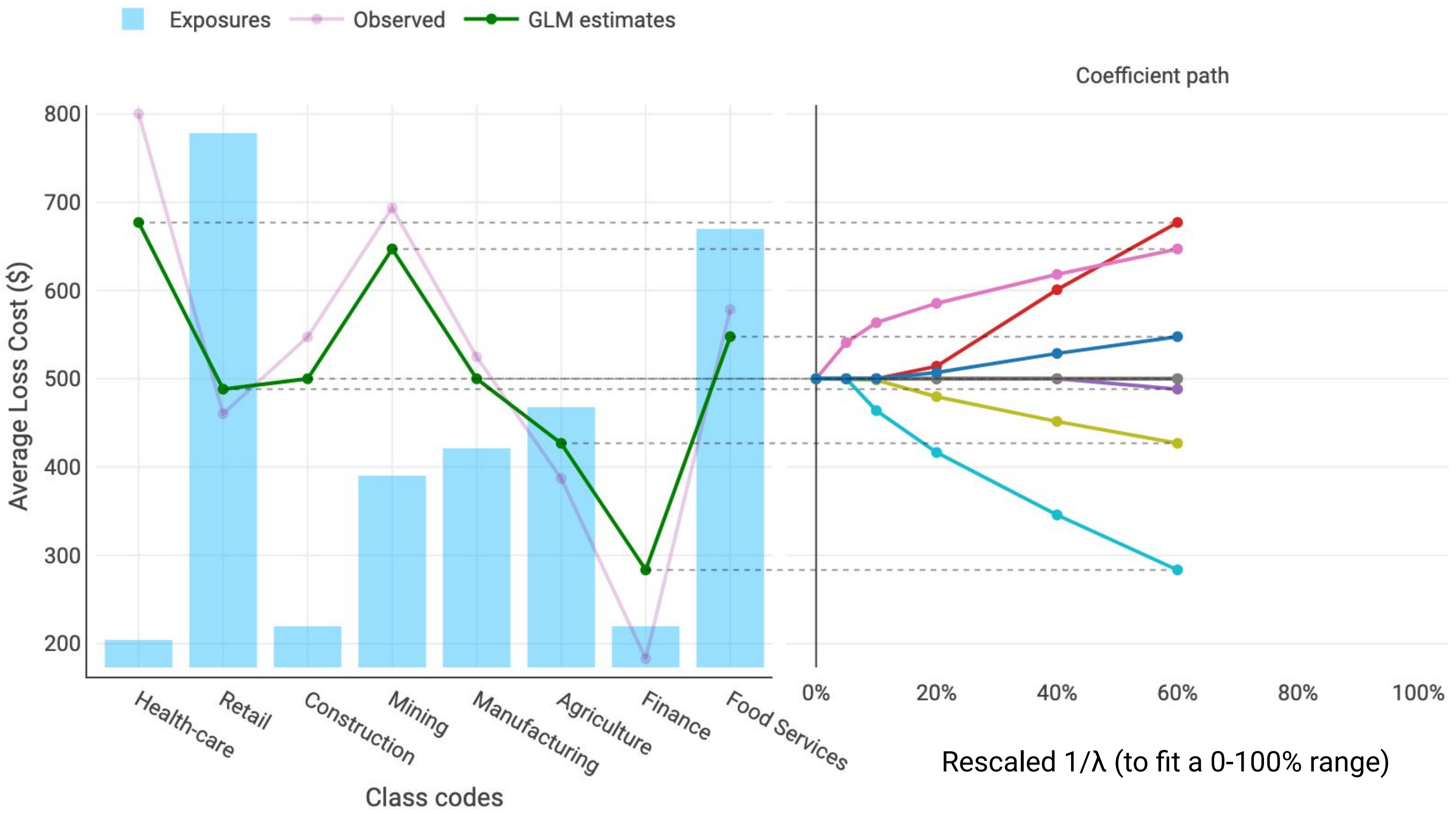
Workers Compensation example



Rescaled 1/λ (to fit a 0-100% range)

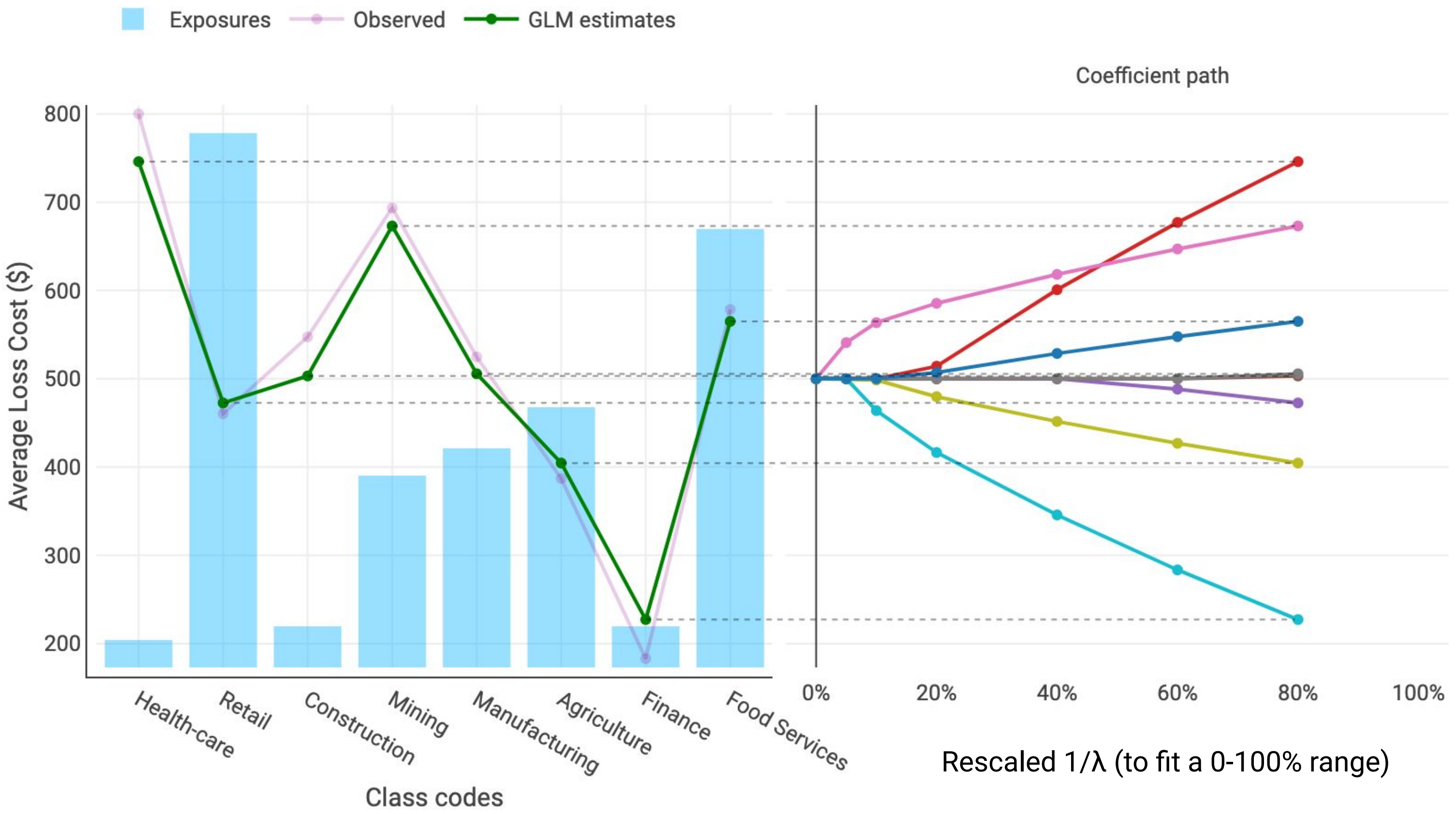
Coefficient path graph of the Lasso

Workers Compensation example



Coefficient path graph of the Lasso

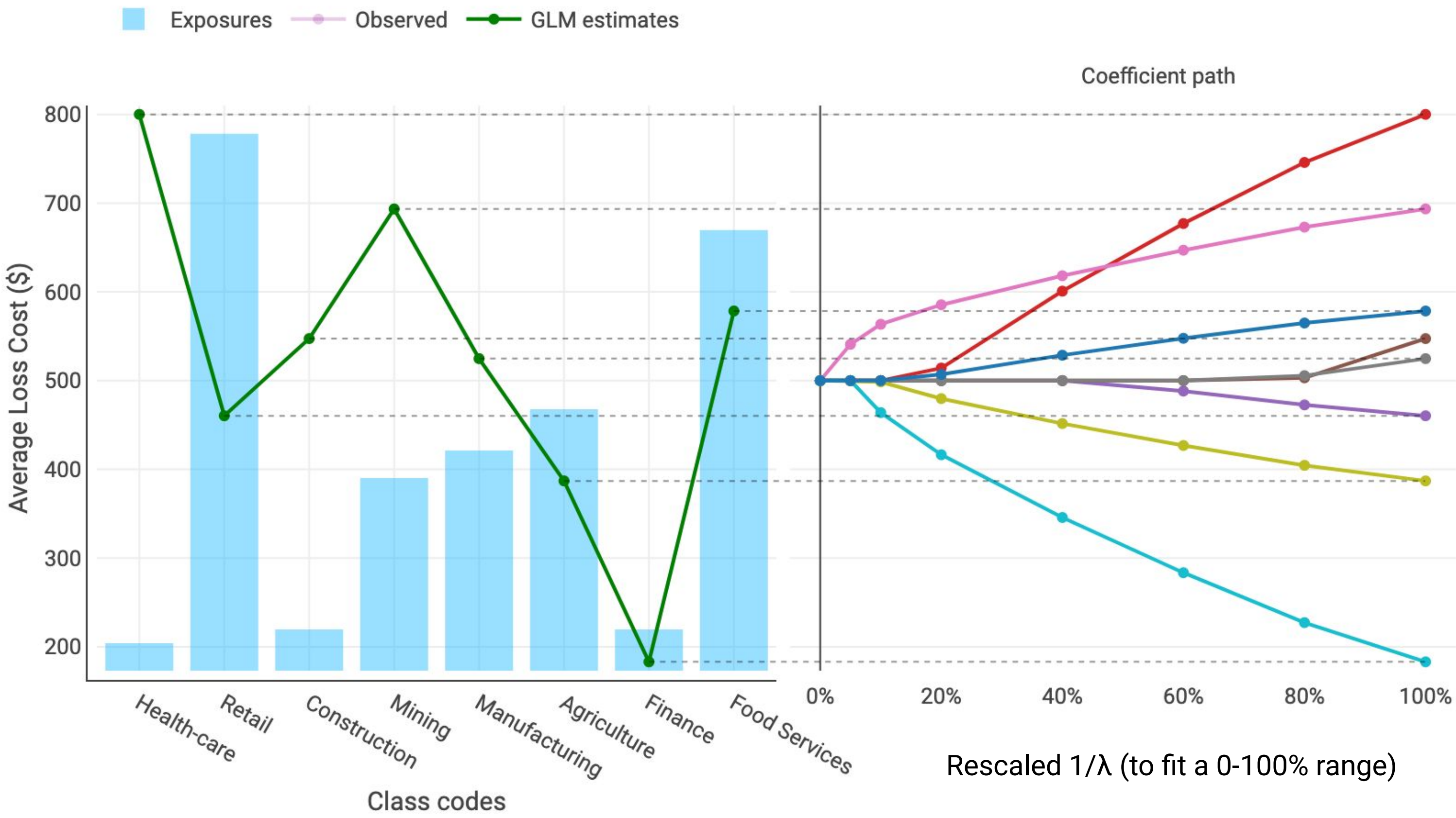
Workers Compensation example



Rescaled 1/λ (to fit a 0-100% range)

Coefficient path graph of the Lasso

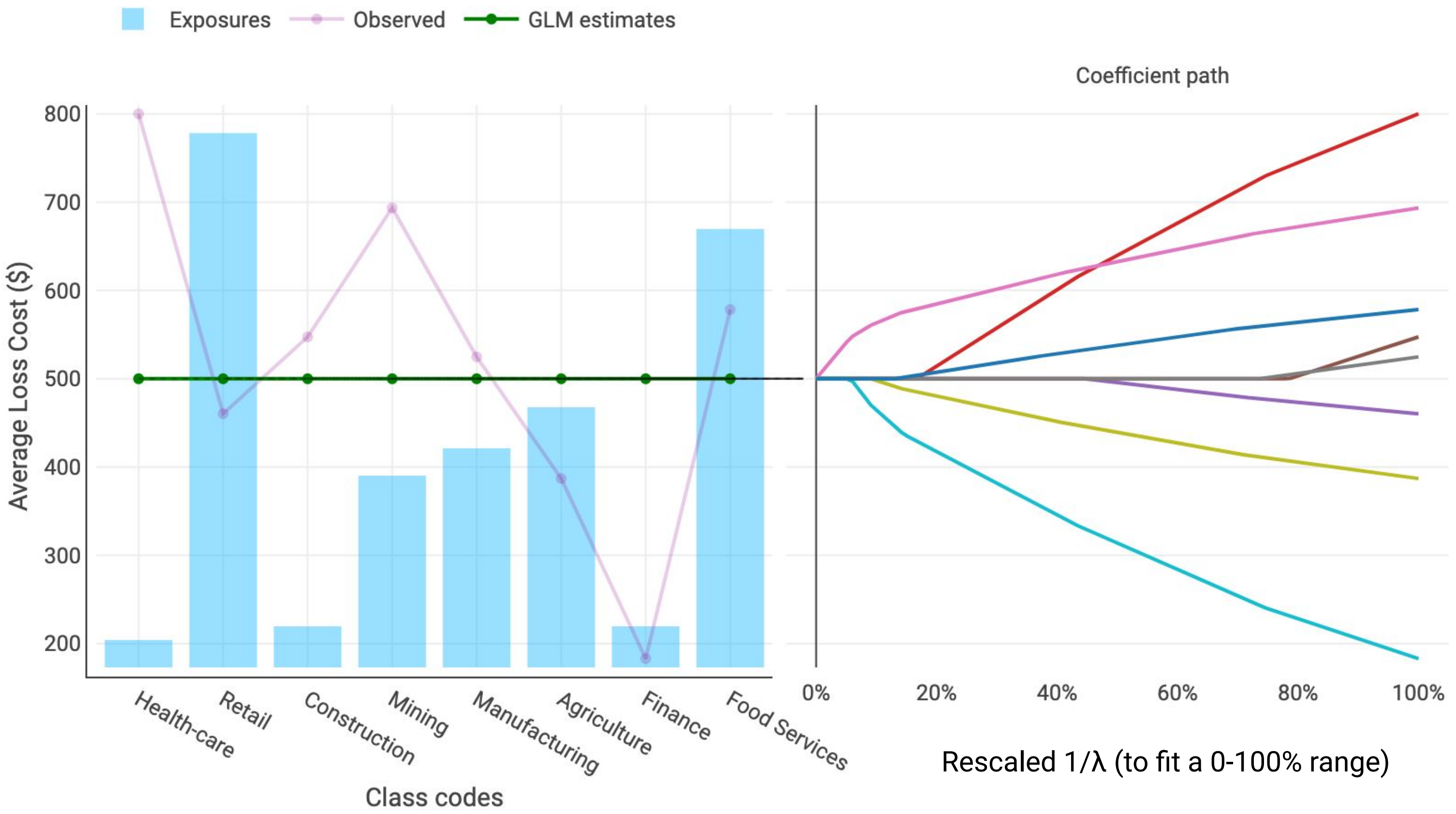
Workers Compensation example



Rescaled 1/λ (to fit a 0-100% range)

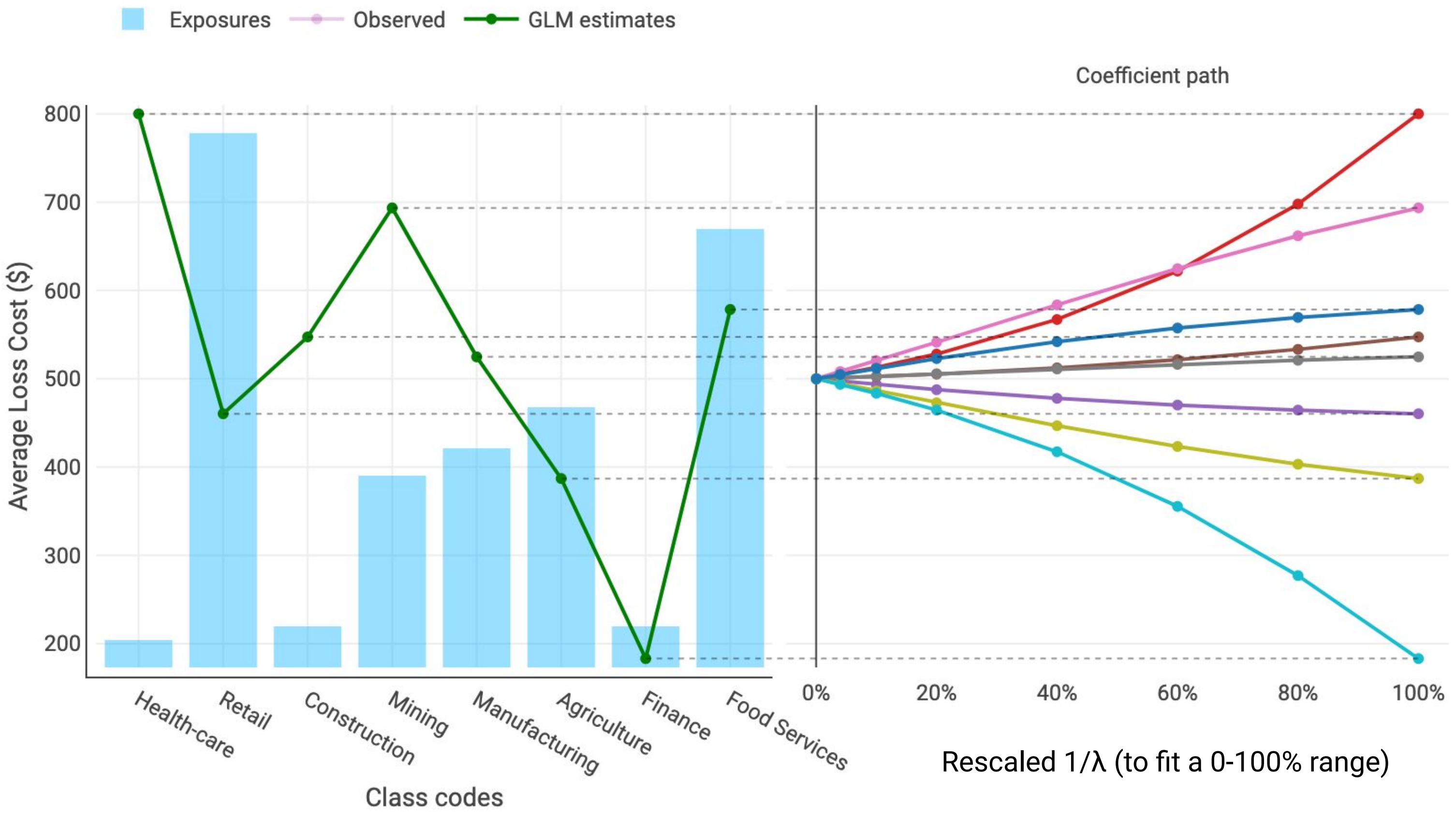
Coefficient path graph of the Lasso

Workers Compensation example



Coefficient path graph of the Ridge

The same graph can be computed for a Ridge regression

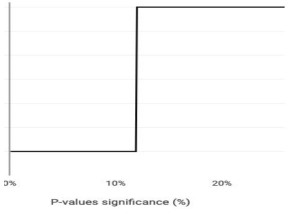


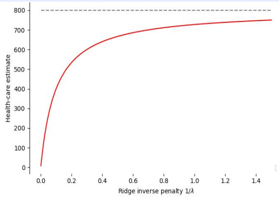
Rescaled 1/λ (to fit a 0-100% range)

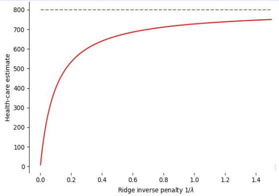
Comparing different techniques

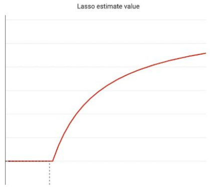
	Levels Selection	Credibility	Ridge Regression	Lasso Regression
Control low-exposure segments to prevent overfitting	All the techniques presented today aim at controlling overfitting.			
Set coefficients of low-exposure segments at zero	Selection of effects	No selection of effects		Selection of effects
Shrink low-exposure segments	No	This allows to tolerate segments with limited (yet usable) data		
Works for multivariate models	Yes	No	Yes	Yes
Creates transparent models (GLM or additive models)	Designed for the GLM framework			
Natively manage non-linear effects	These techniques work on “pure GLM” (linear & categorical effects)			

Coefficient depending on the robustness parameter









Gradient Boosting and GLMs

AKUR8

There is a strong relationship between Credibility and Penalized Regression methods.

There is an equal **connection**, between **Gradient Boosting Machines** (GBMs) and **Penalized Regression.**

Such additional connection highlights the flexibility of the Penalized framework, which can be used to enhance components of the current methodologies of insurance pricing.

Introduction to GBM

What is a Boosted Tree ?

GBMs are also referred as **Boosted Trees**.

- **Boosted** as in Boosting - a learning technique that “learns from the mistakes” by iterating models on residuals.

- **Trees** as in Decision Tree - simple model that predicts a target based on decision rules learnt from the data.

What is a tree

Trees estimate losses via recursive if/else decision rules.

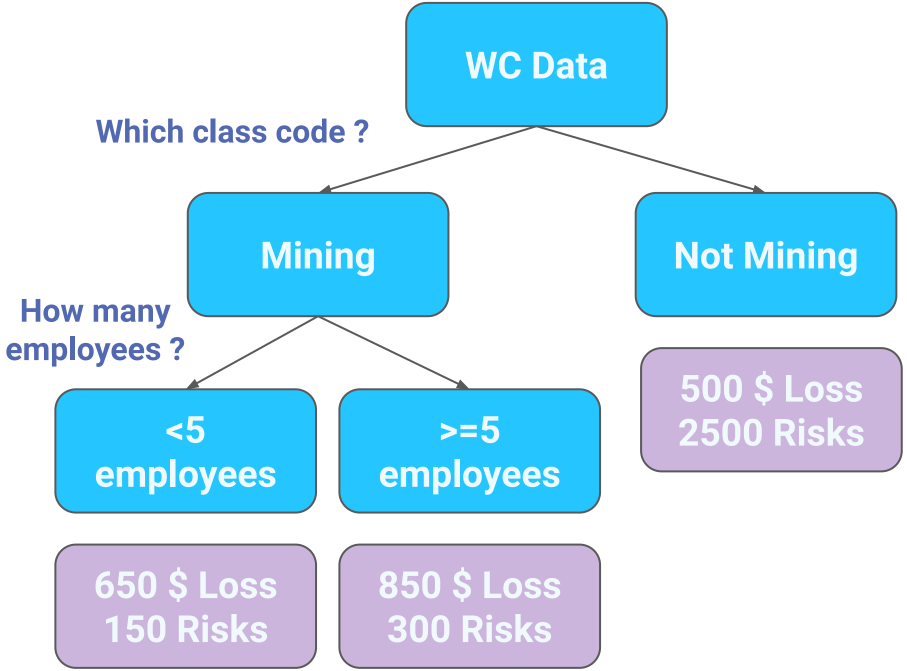
Rules are inferred from the data in a **greedy fashion.**

Each possible two way split of the data is evaluated by comparing the averages of the two complementary partitions.

The split leading to the biggest likelihood increase will be selected.

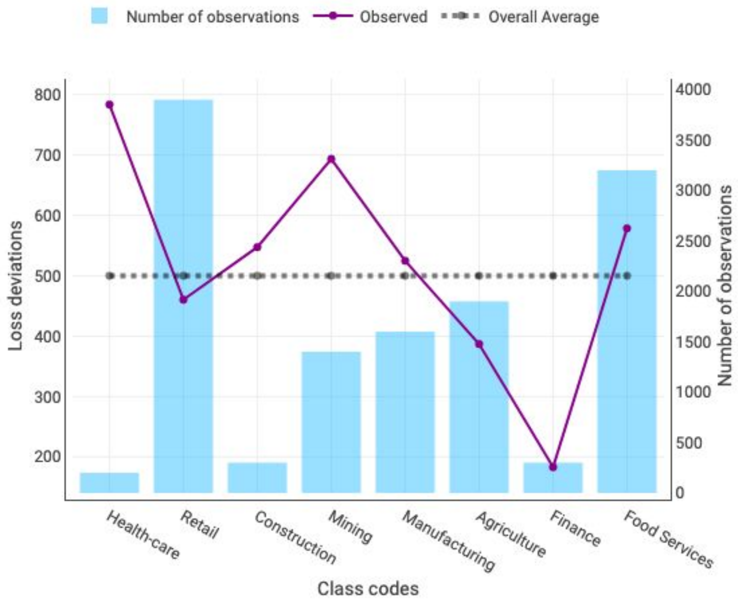
The search is then iterated on each subpopulation until one stopping criteria is met, such as

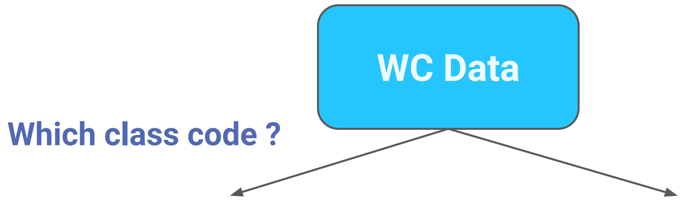
- Maximum tree depth
- Minimum amount observation per leaf
- Min deviance gain…



Application: Worker Compensation

Building a tree on Worker Compensation data



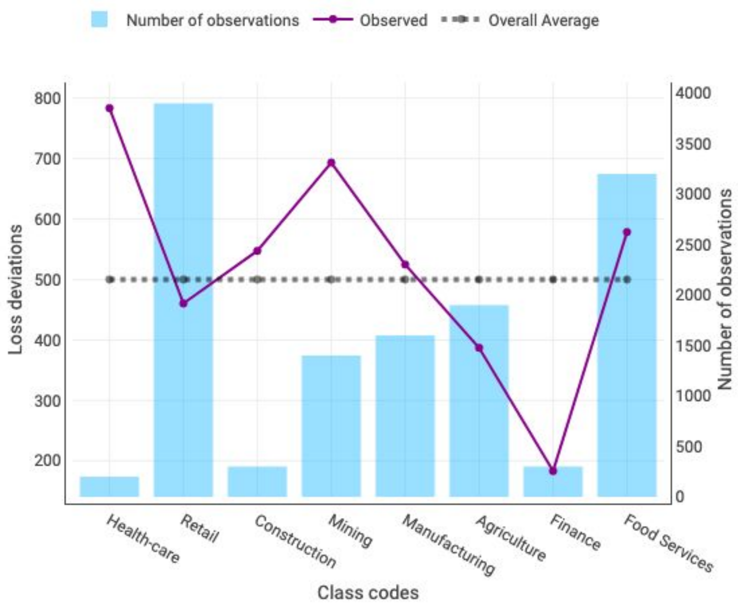


Application: Worker Compensation

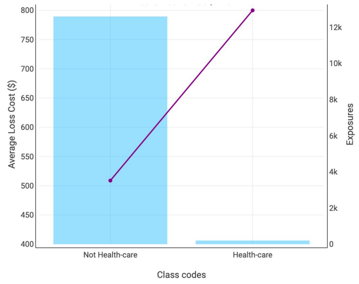
Building a tree on Worker Compensation data

WC Data

Which class code ?

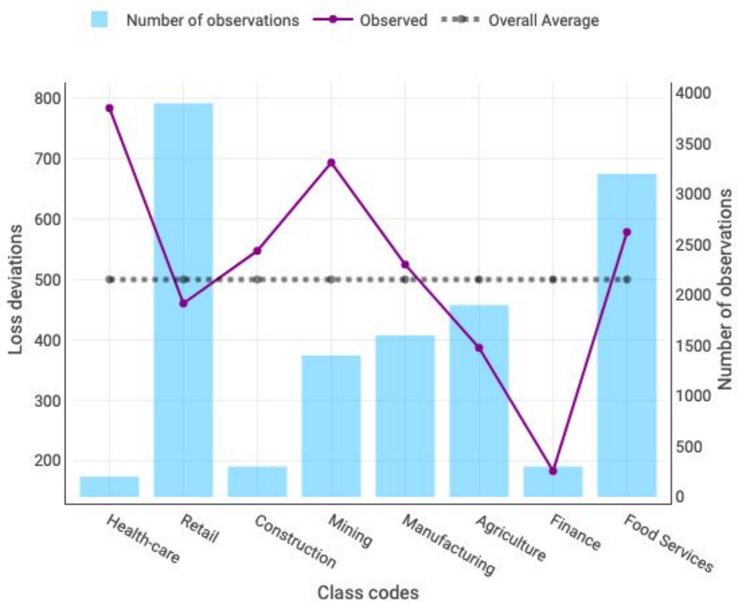


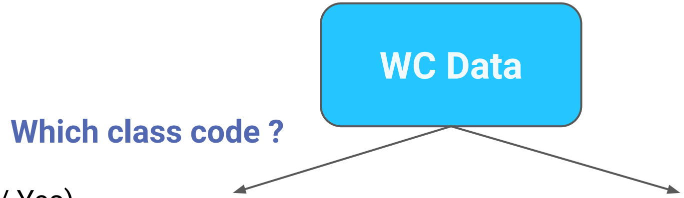
Health Care (No / Yes)



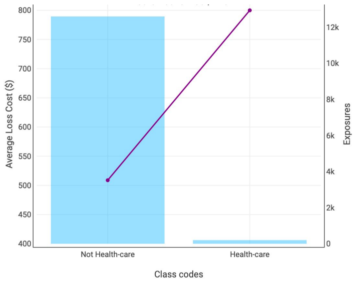
Application: Worker Compensation

Building a tree on Worker Compensation data

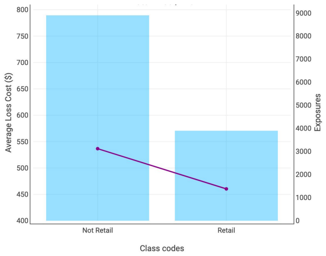




Health Care (No / Yes)

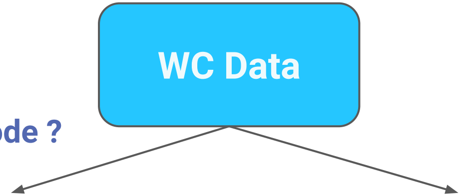


Retail (No / Yes)

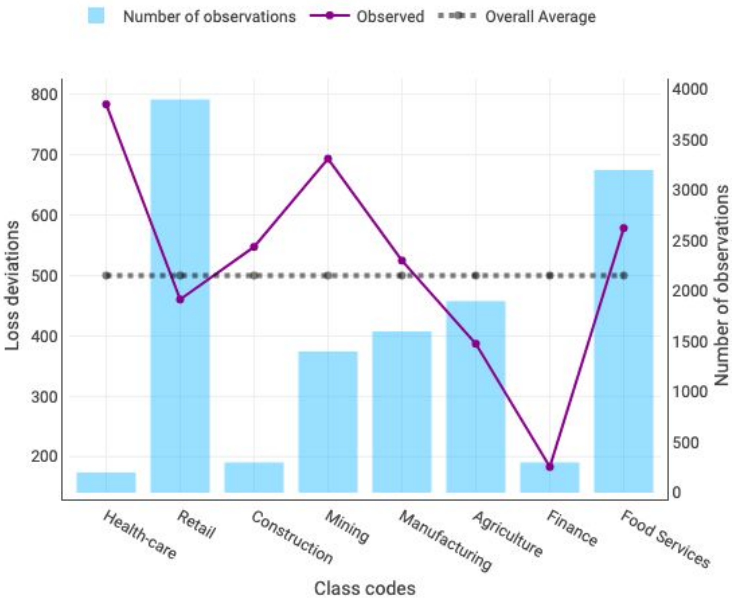


Application: Worker Compensation

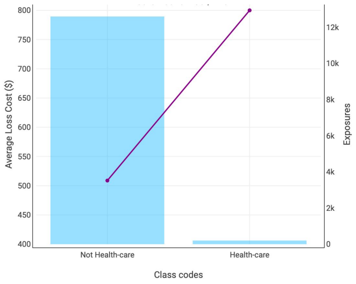
Building a tree on Worker Compensation data



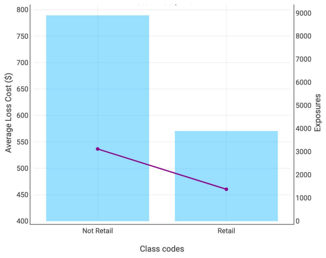
Which class code ?



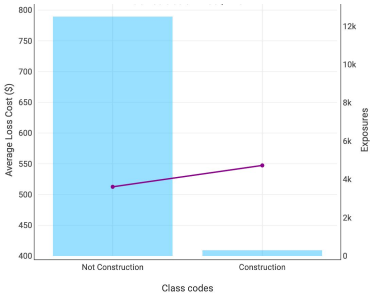
Health Care (No / Yes)



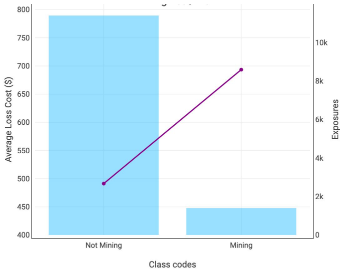
Retail (No / Yes)



Construction (No / Yes)



Mining (No / Yes)

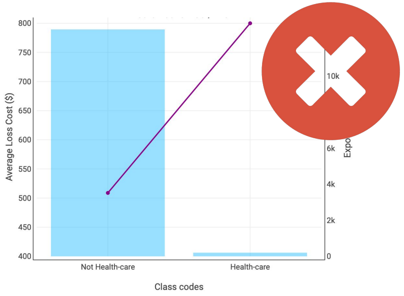


Application: Worker Compensation

WC Data

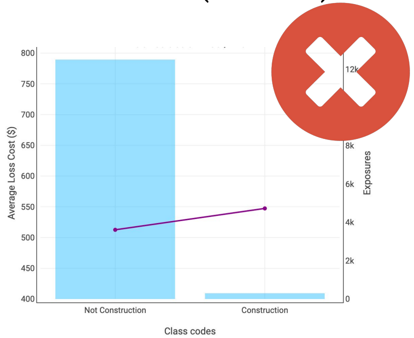
Which class code ?

Health Care (No / Yes)

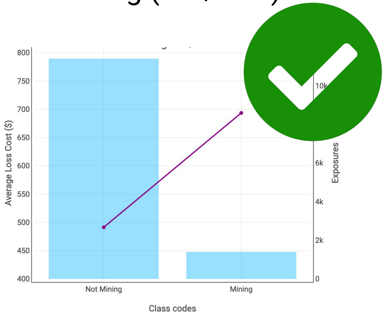


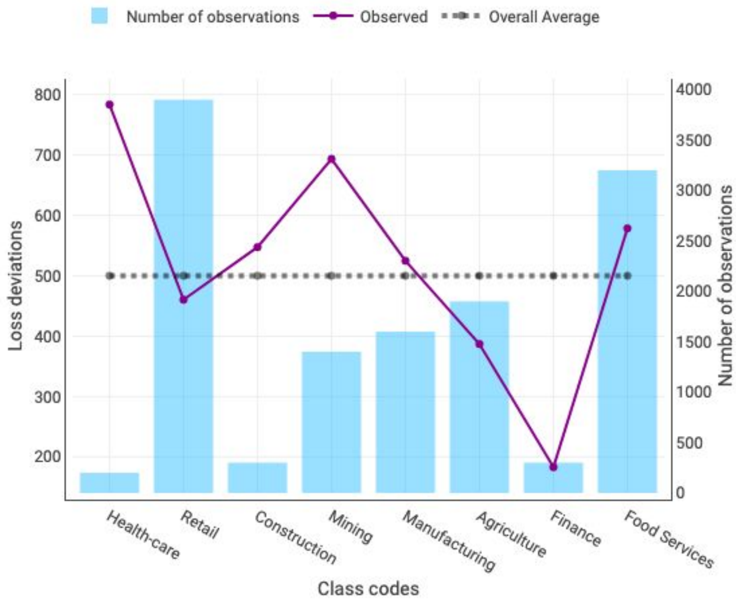
Retail (No / Yes)

Construction (No / Yes)



Mining (No / Yes)



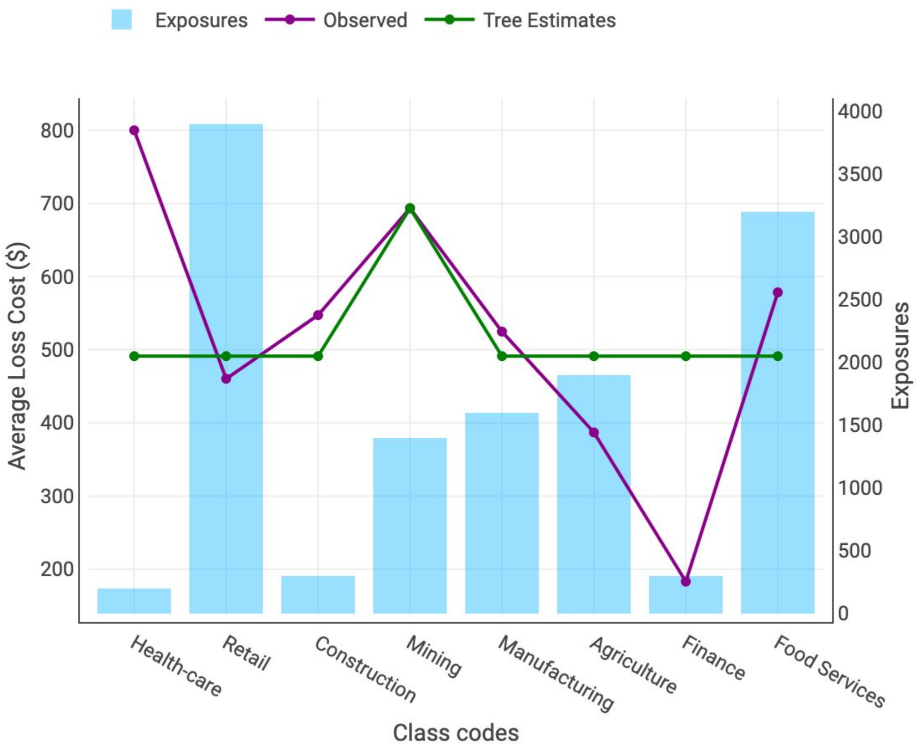


Application: Worker Compensation

The tree split the dataset between Mining and Not Mining, leading to two different predictions.

In a GBM, the **first** tree is the first step of the learning procedure: the **boosting.**

The boosting procedure consist of three steps:



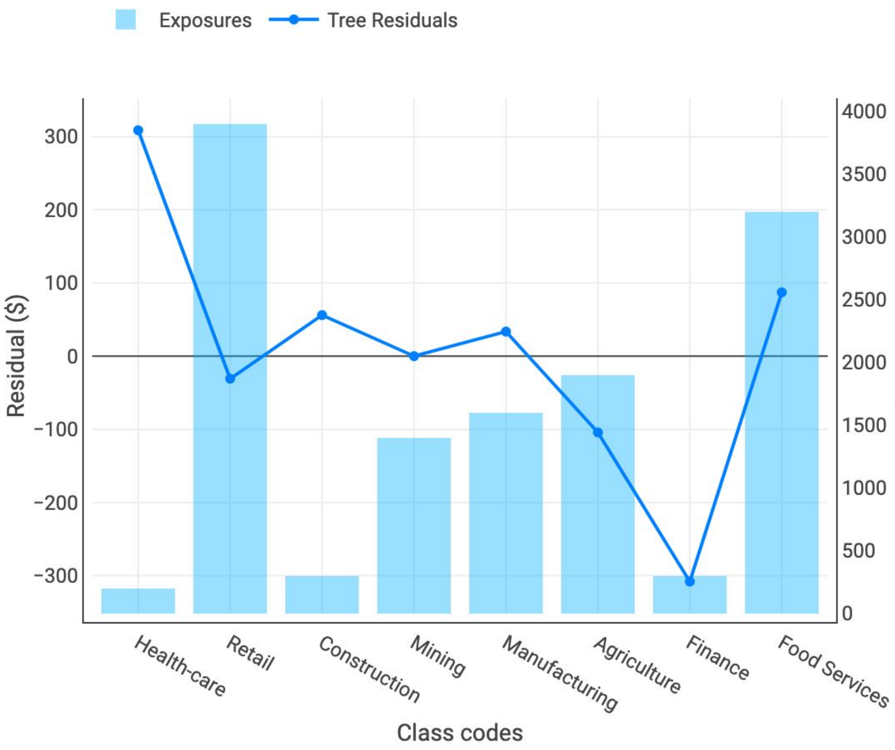
Application: Worker Compensation

The tree split the dataset between Mining and Not Mining, leading to two different predictions.

In a GBM, the **first** tree is the first step of the learning procedure: the **boosting.**

The boosting procedure consist of three steps:

1. Compute the **residuals**



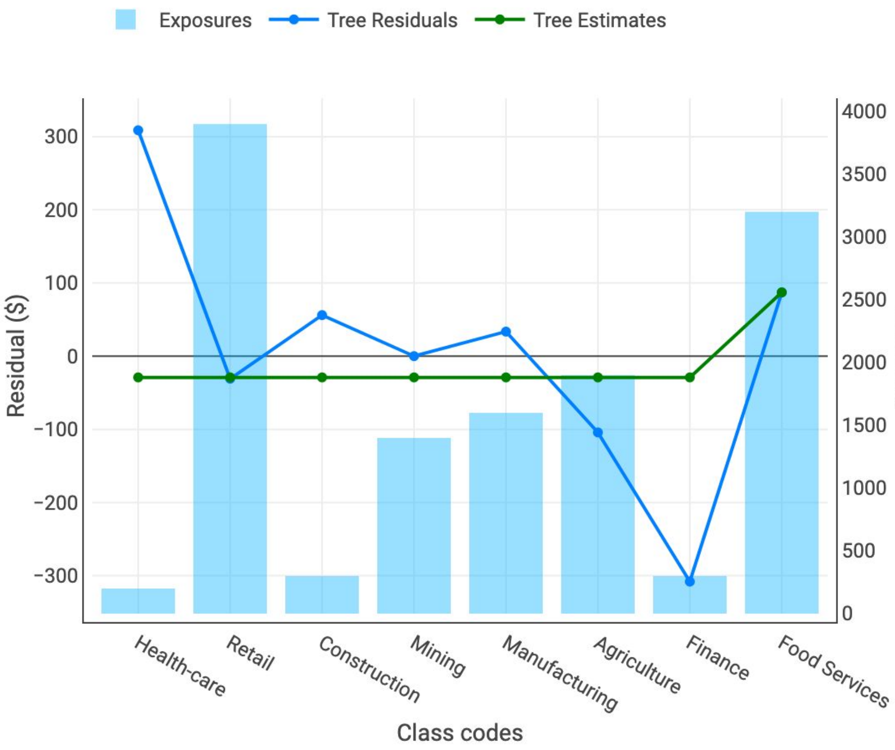
Application: Worker Compensation

The tree split the dataset between Mining and Not Mining, leading to two different predictions.

In a GBM, the **first** tree is the first step of the learning procedure: the **boosting.**

The boosting procedure consist of three steps:

1. Compute the **residuals**
2. Fit a **new tree**



Application: Worker Compensation

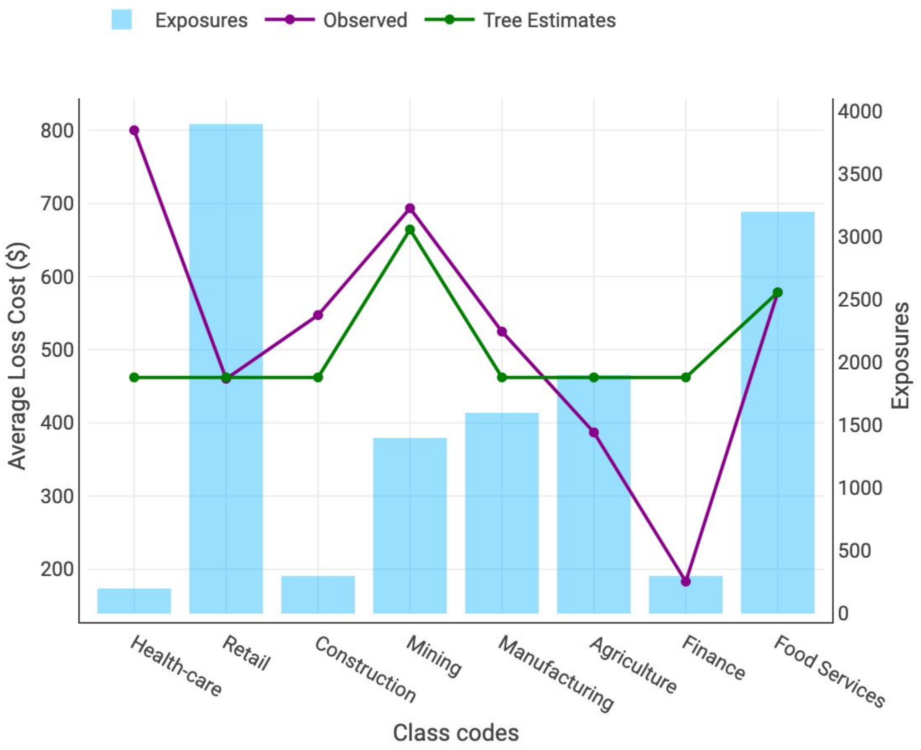
The tree split the dataset between Mining and Not Mining, leading to two different predictions.

In a GBM, the **first** tree is the first step of the learning procedure: the **boosting.**

The boosting procedure consist of three steps:

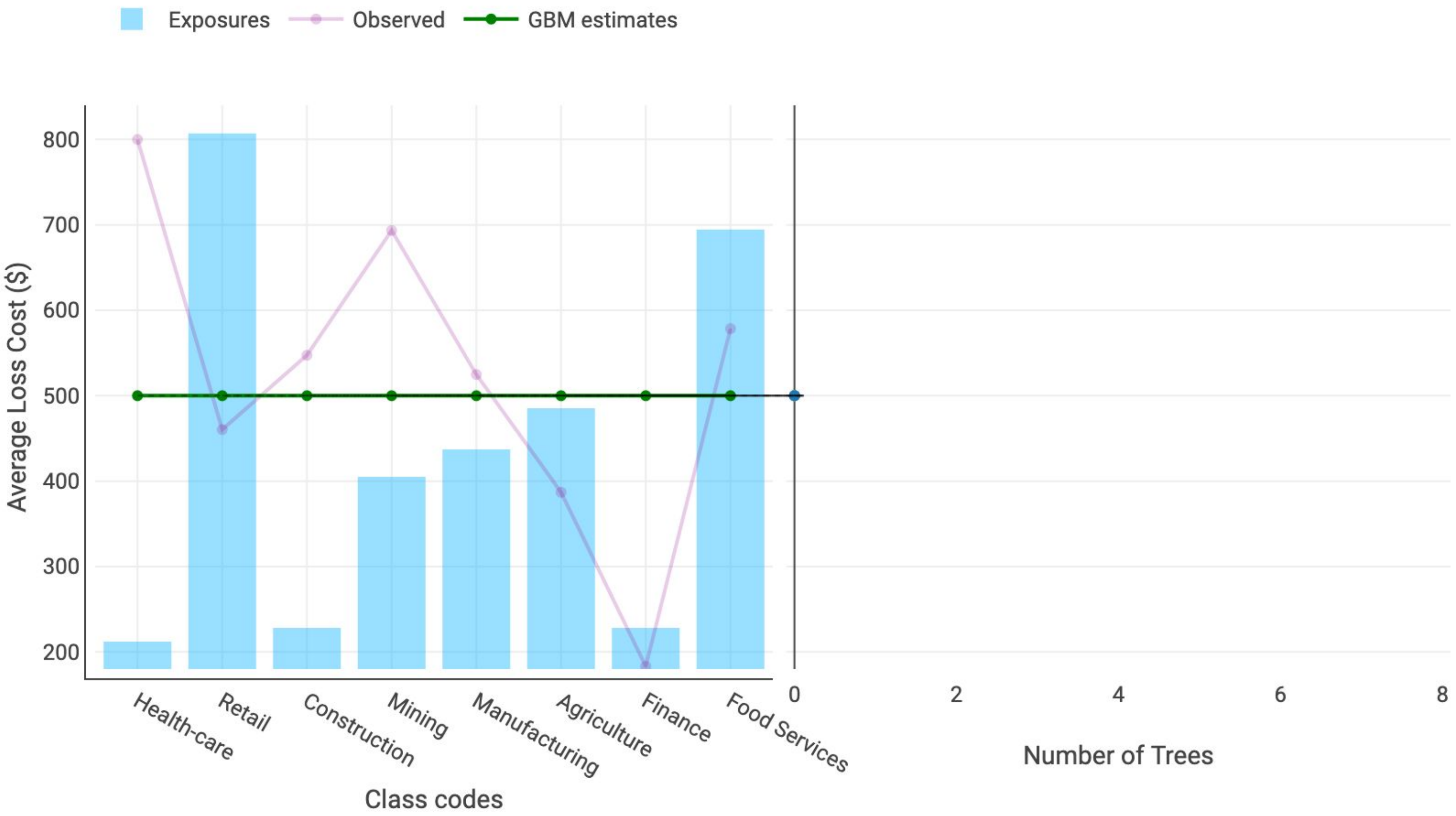
1. Compute the **residuals**
2. Fit a **new tree**
3. Compute the estimates by summing the previous trees

Estimate = Tree 1 + Tree 2 + …



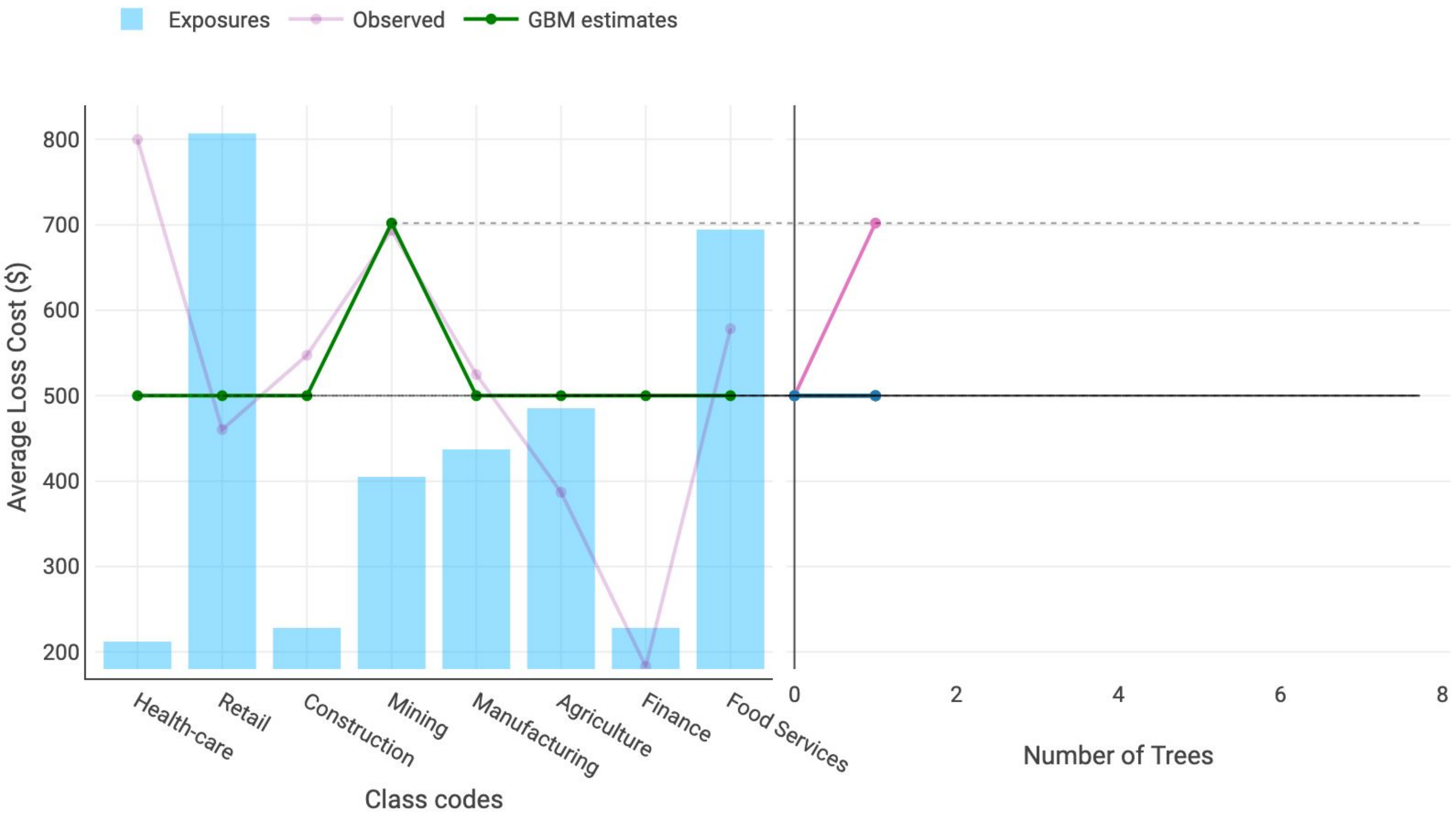
Coefficient path graph of a GBM

Workers Compensation example



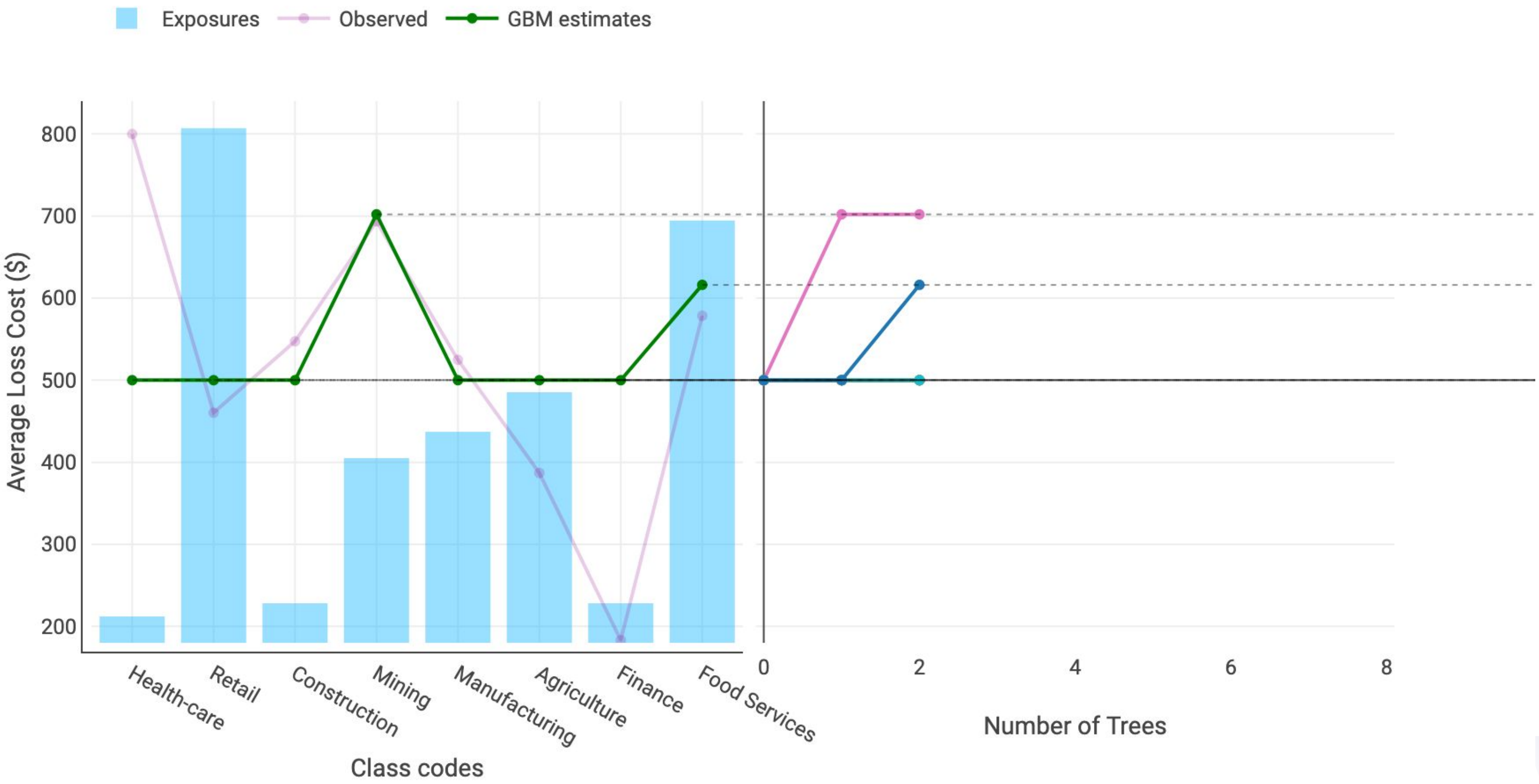
Coefficient path graph of a GBM

Workers Compensation example



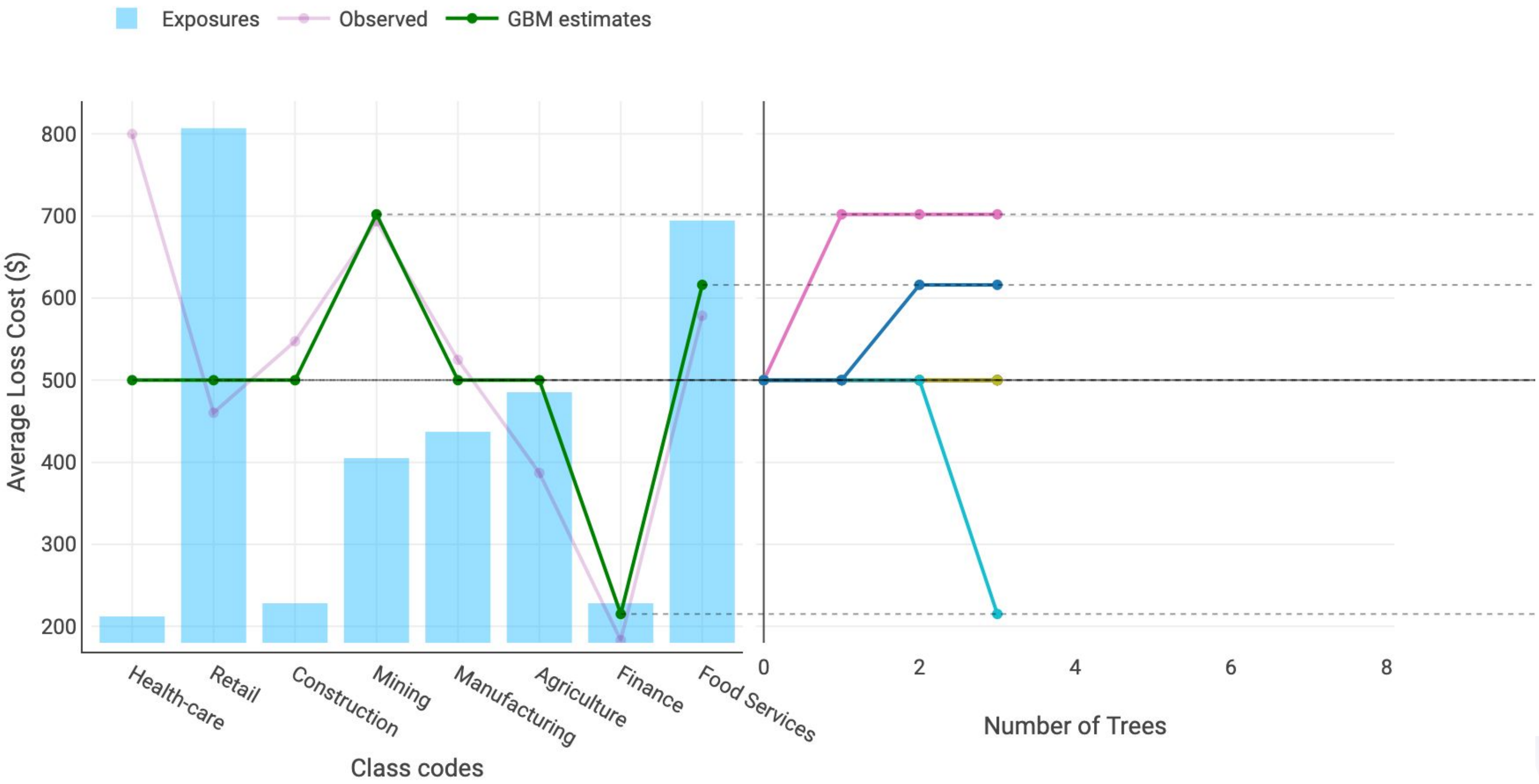
Coefficient path graph of a GBM

Workers Compensation example



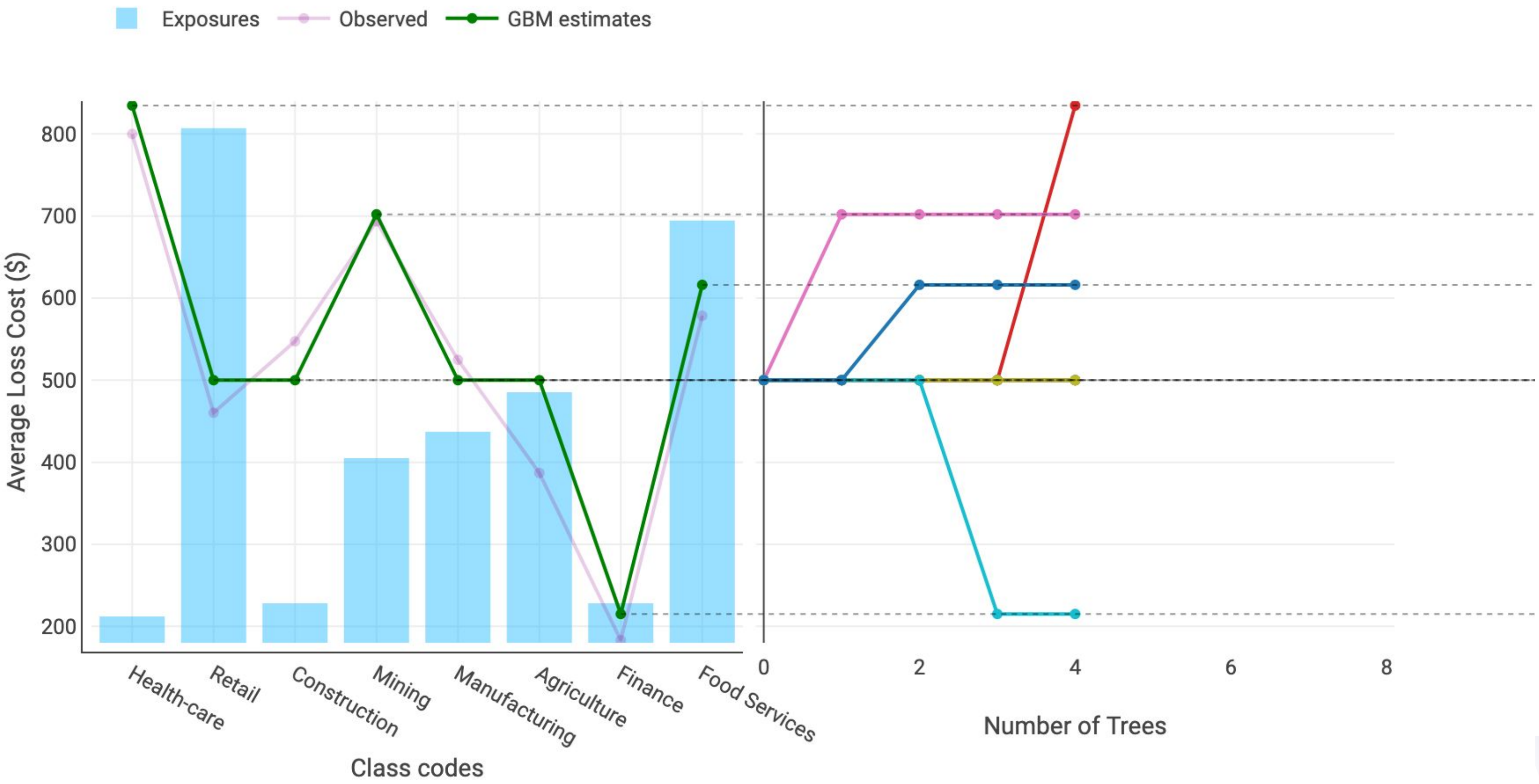
Coefficient path graph of a GBM

Workers Compensation example



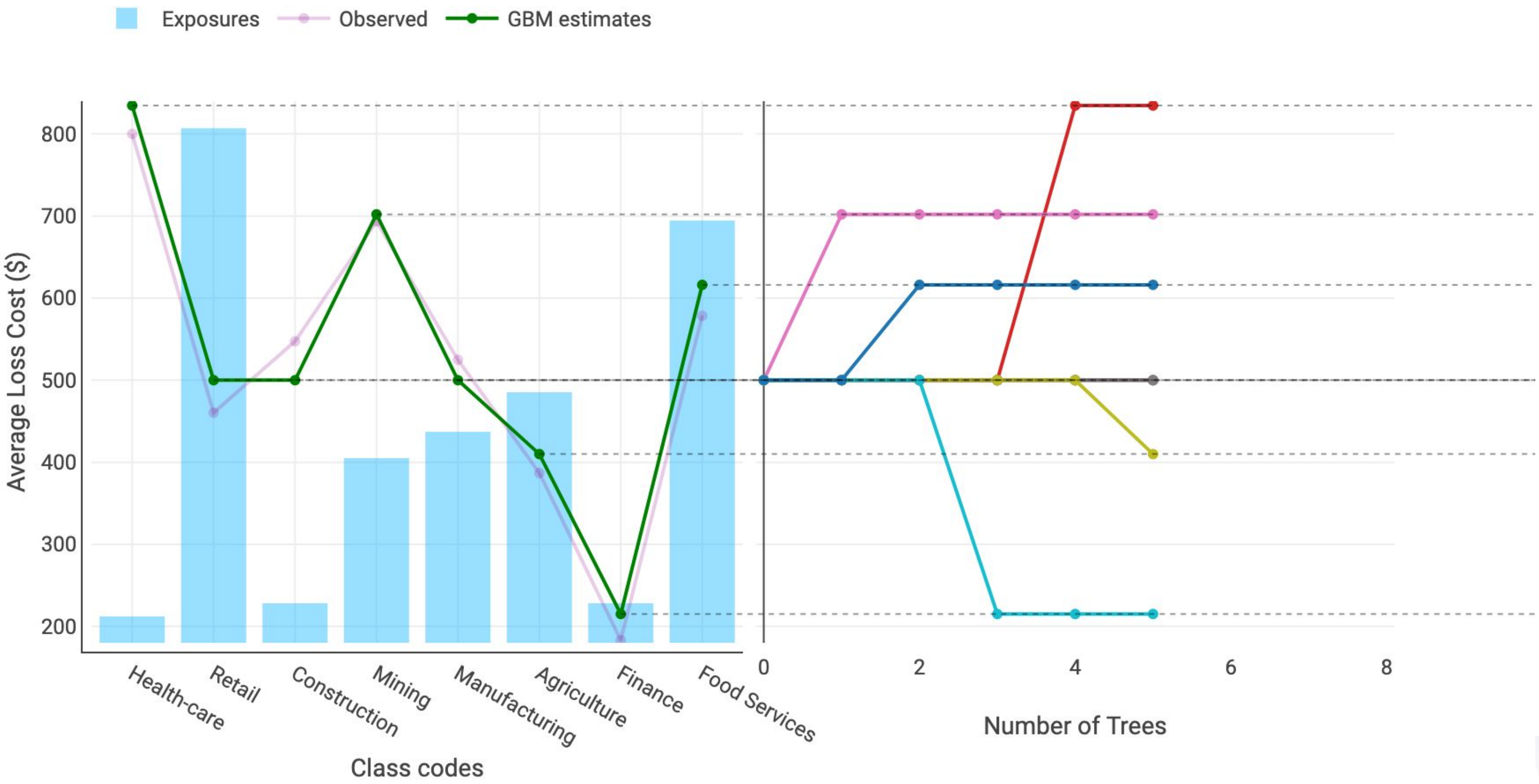
Coefficient path graph of a GBM

Workers Compensation example



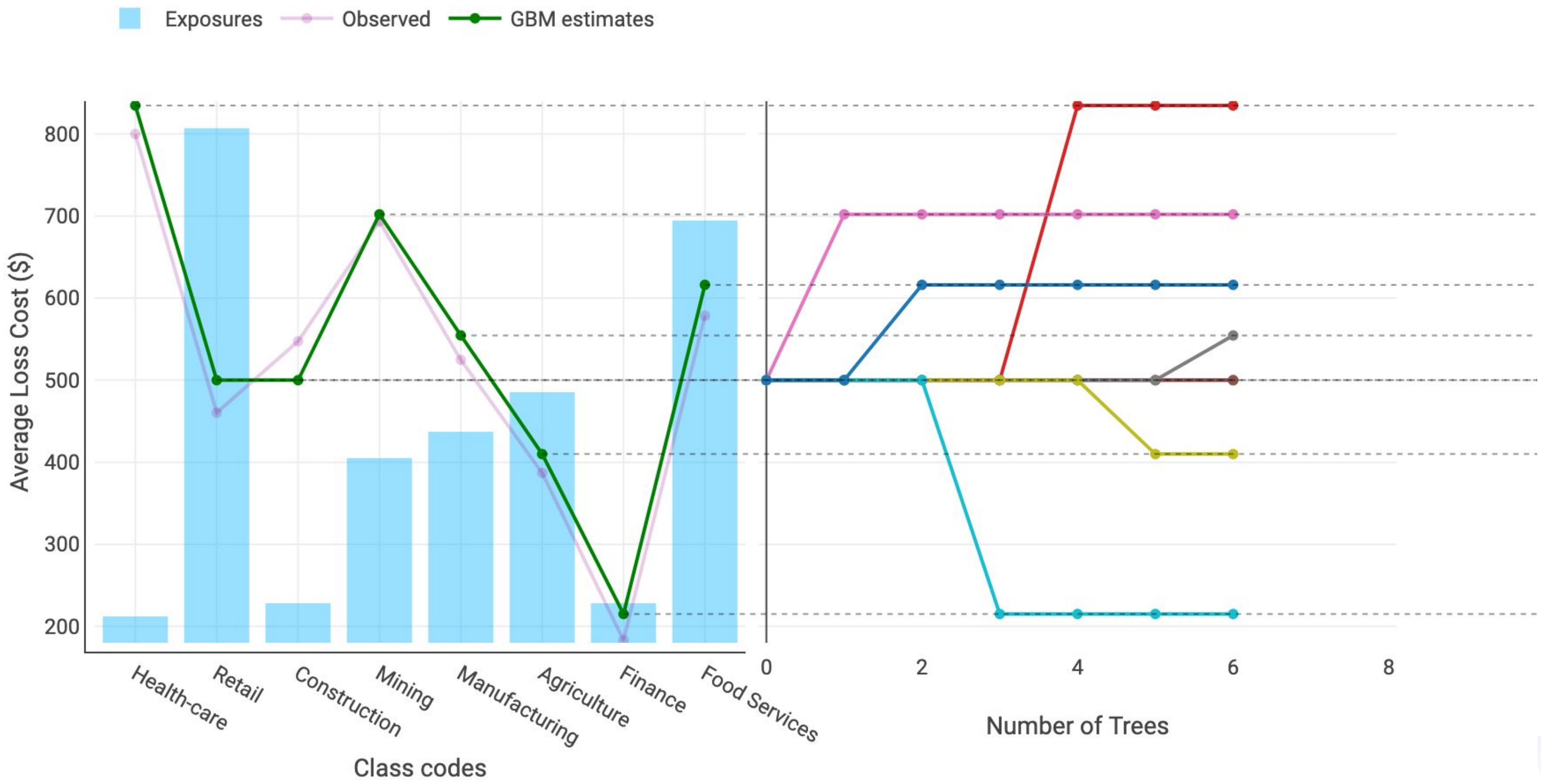
Coefficient path graph of a GBM

Workers Compensation example



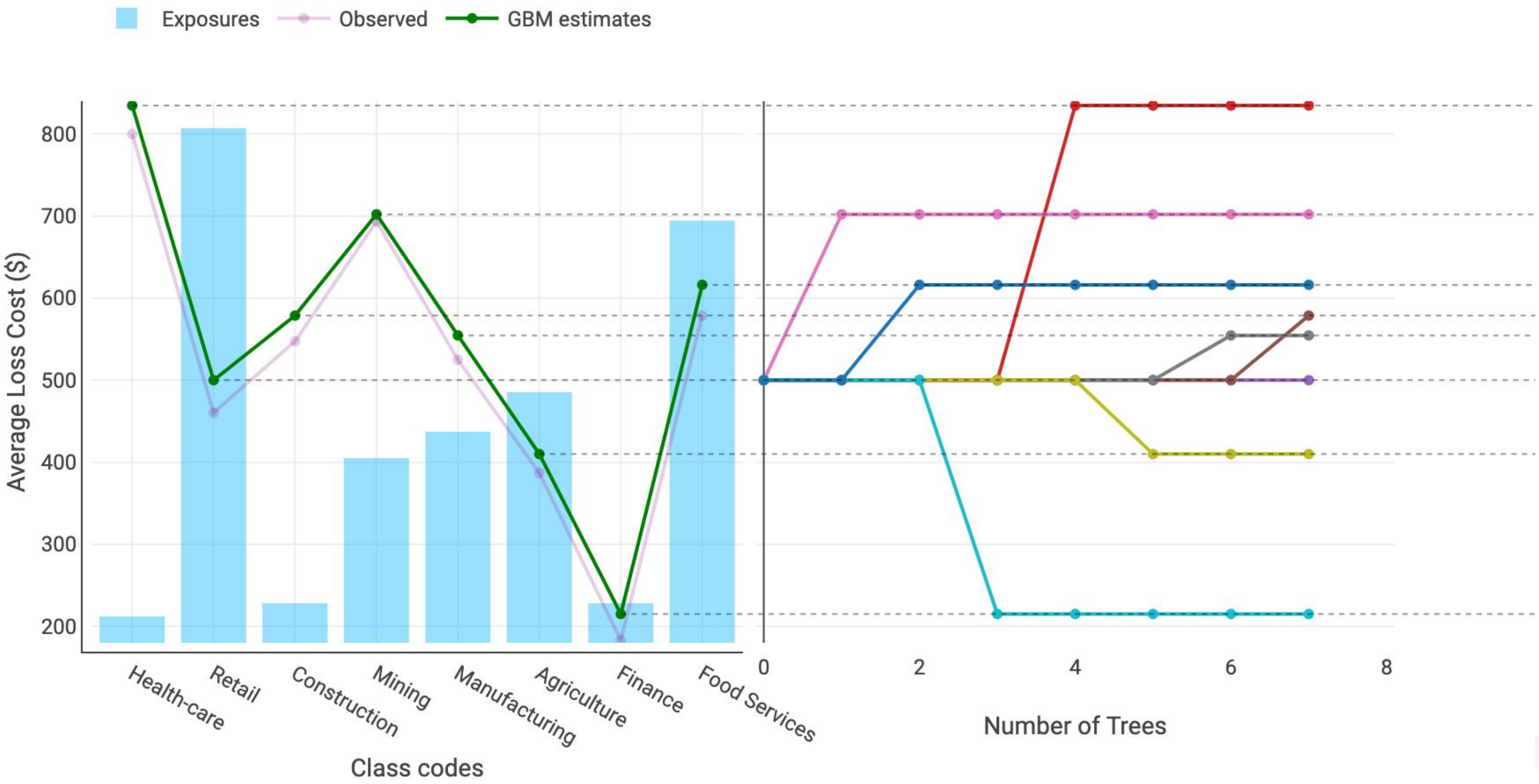
Coefficient path graph of a GBM

Workers Compensation example



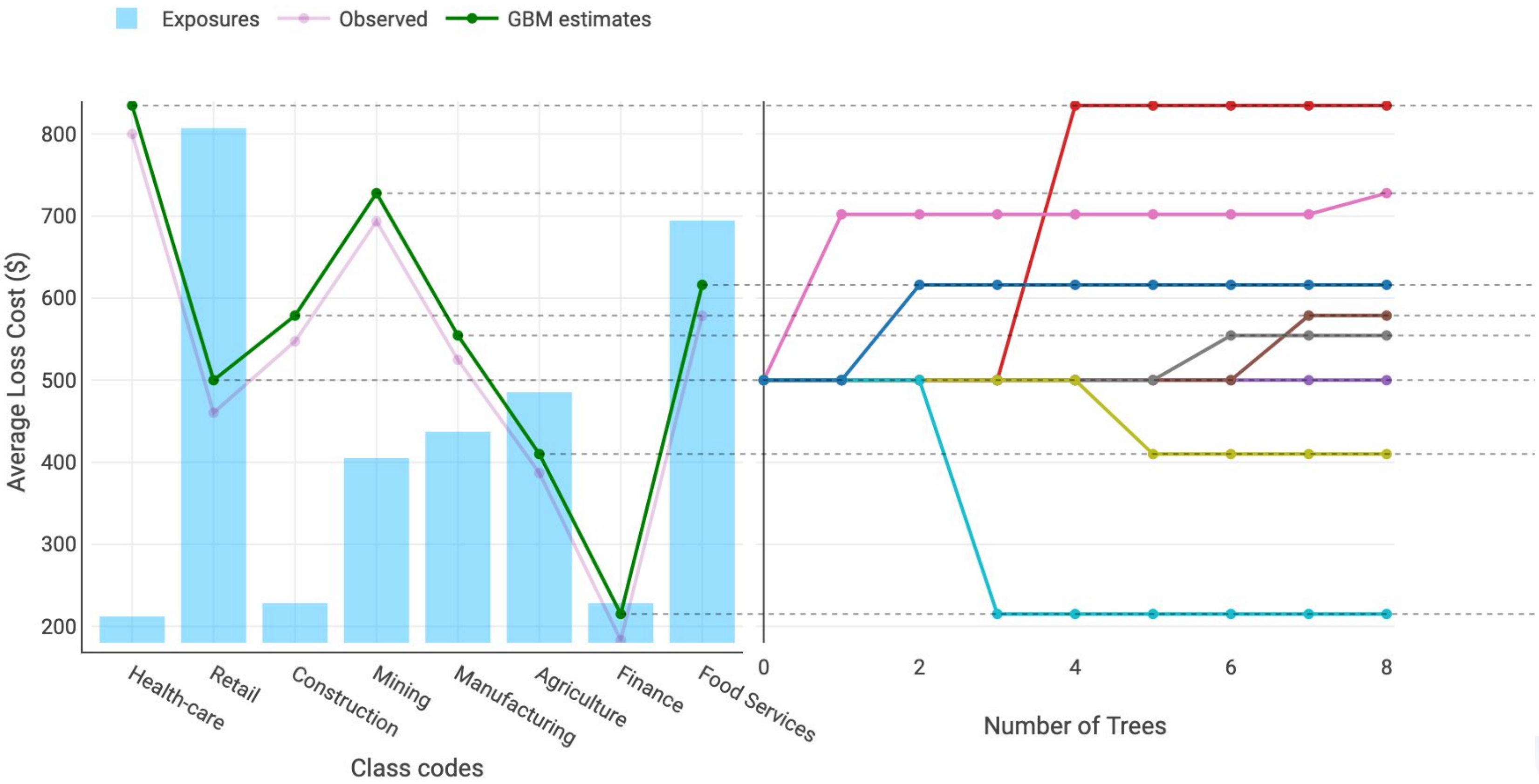
Coefficient path graph of a GBM

Workers Compensation example



Coefficient path graph of a GBM

Workers Compensation example



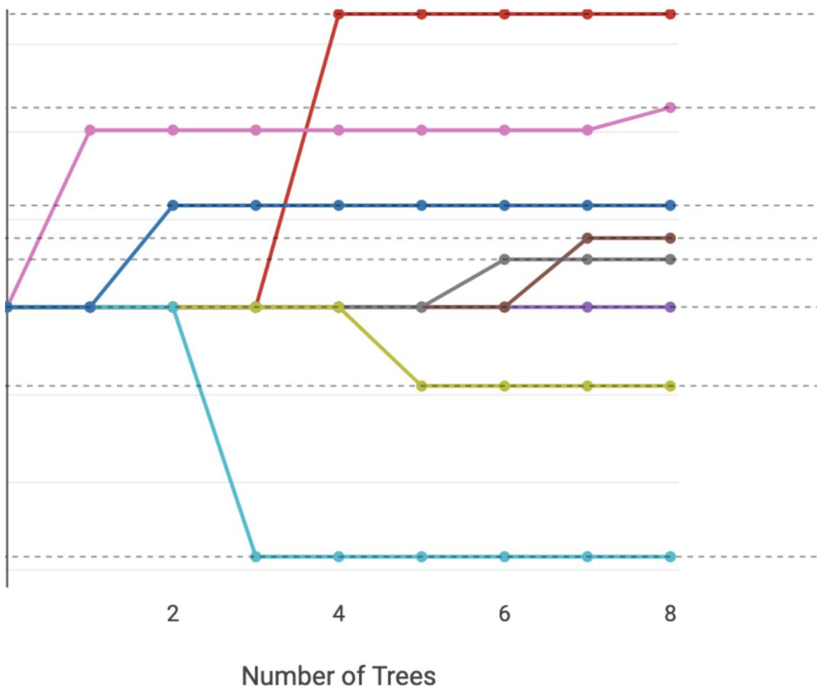
Boosting and stepwise learning

In the simple Worker Compensation example, the GBM learns as in a **forward stepwise procedure,** by iteratively

1. Selecting the most important feature.
2. Including (fitting) the effects.

Forward stepwise procedures work well in a very simple case like here, but they are known to **not handle correctly correlated variables.**

For a similar reason, **boosting procedures are always combined with a learning rate to improve the model's ability to generalize.**



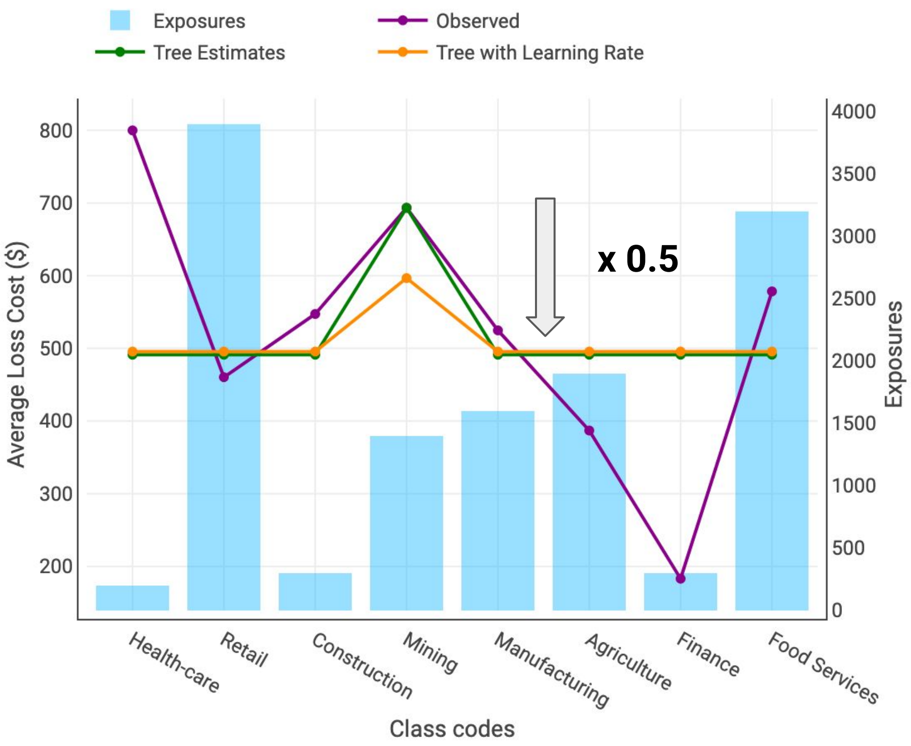
The learning rate

The learning rate is a constant between 0 and 1 that **mitigates** the contribution of an individual tree to the overall prediction.

For each step, the predictions of the tree will be multiplied by the **learning rate**.

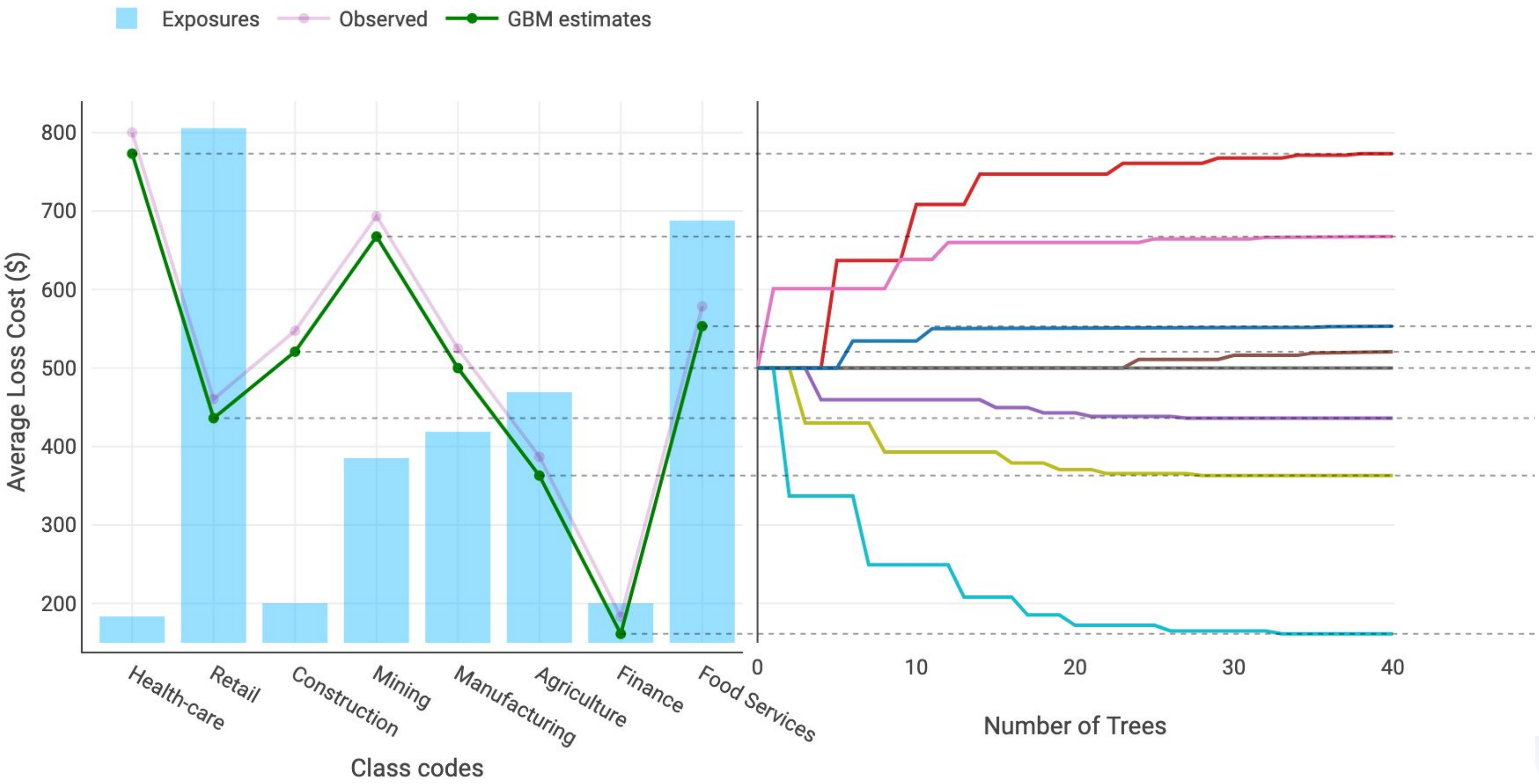
When the learning rate is **0.5** the GBM formula becomes

GBM estimate = **0.5** * Tree_1 + **0.5** * Tree_2 + **0.5** * …



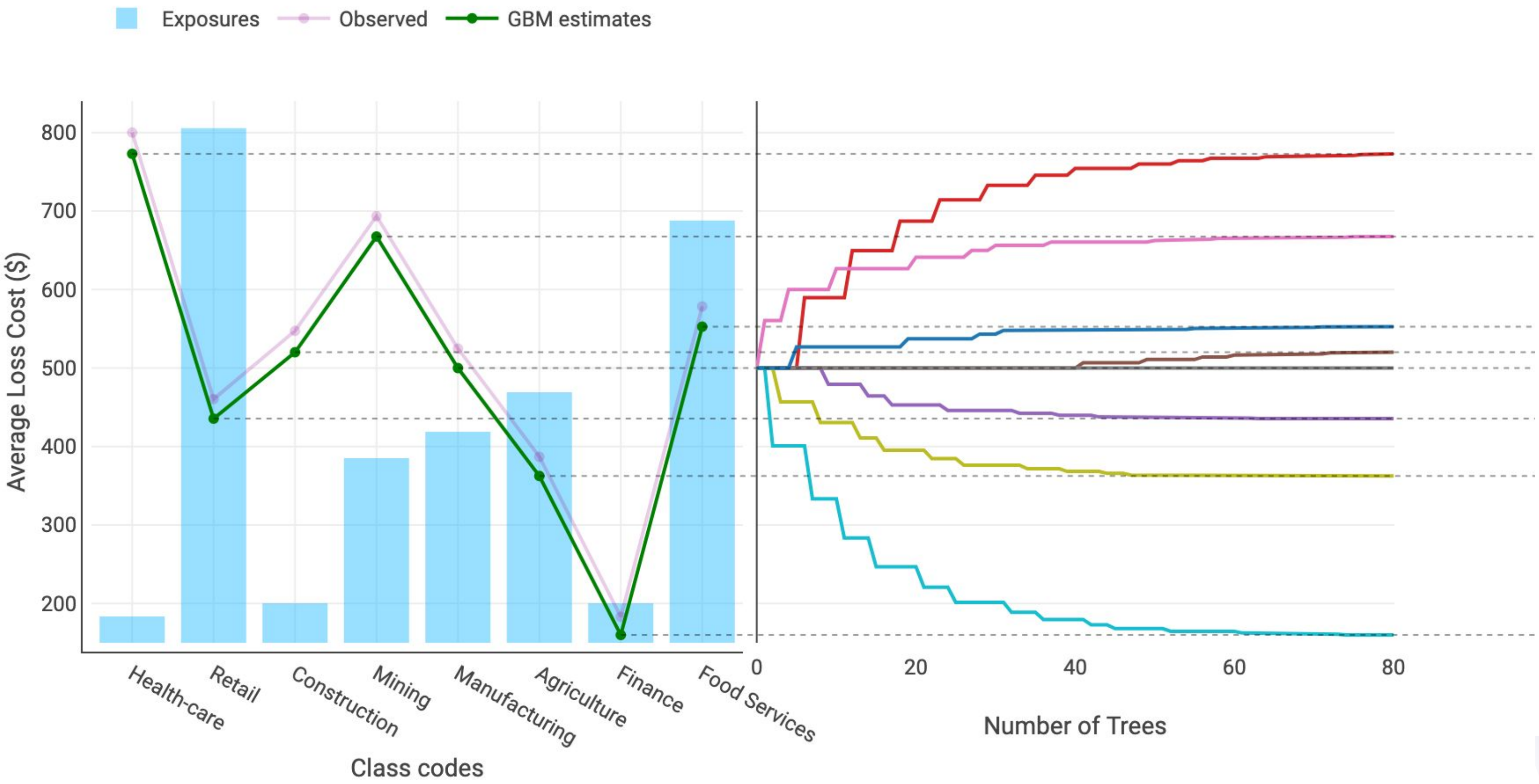
Learning rate = 0.5

Estimate evolution until **40** trees



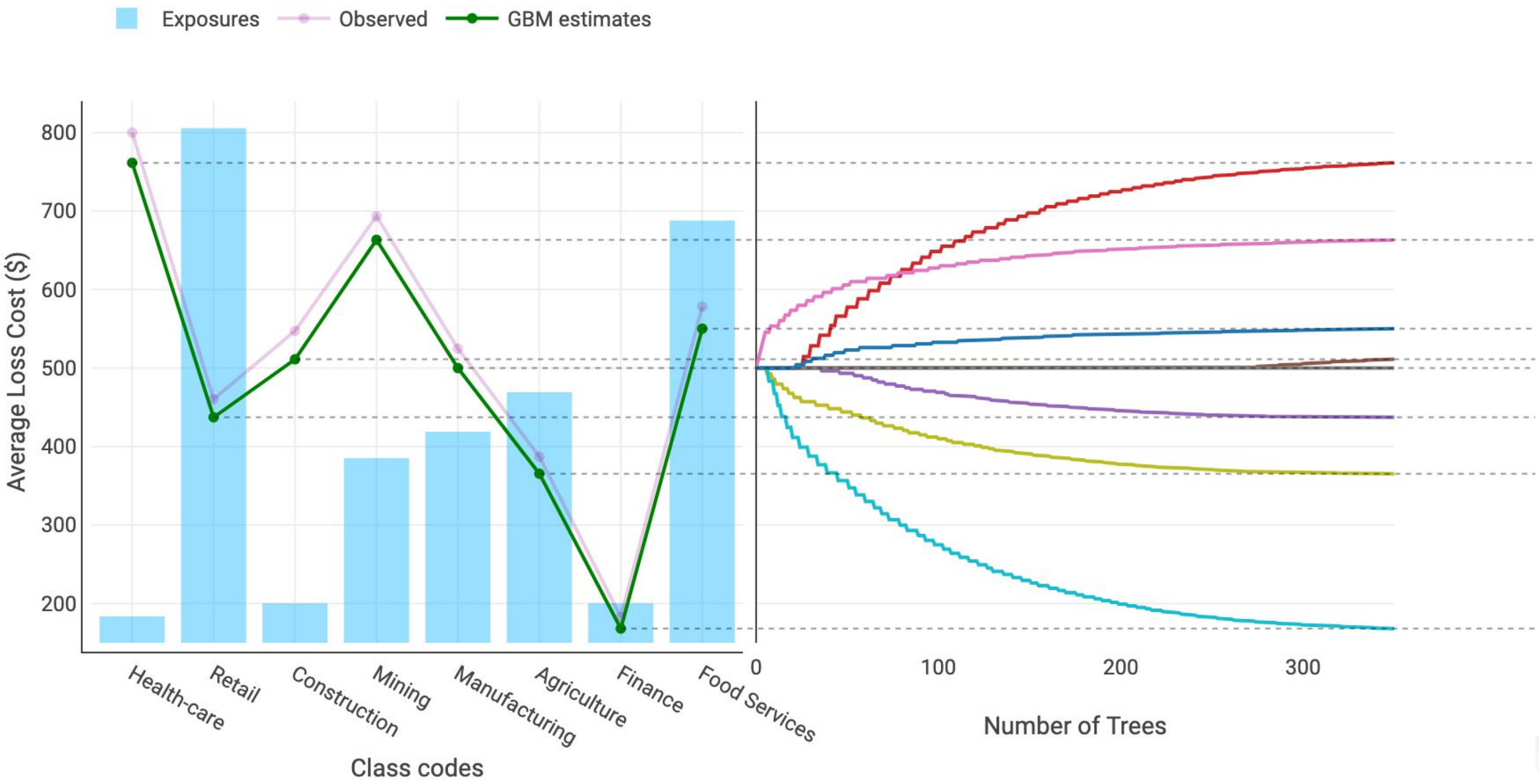
Learning rate = 0.3

Estimate evolution until **80** trees



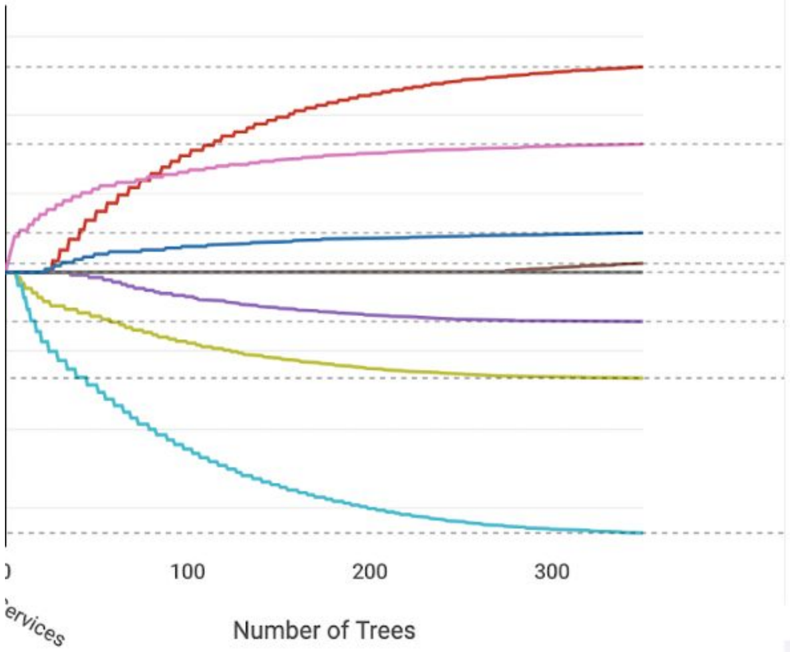
Learning rate = 0.05

Estimate evolution until **350** trees



Toward the coefficient path graph

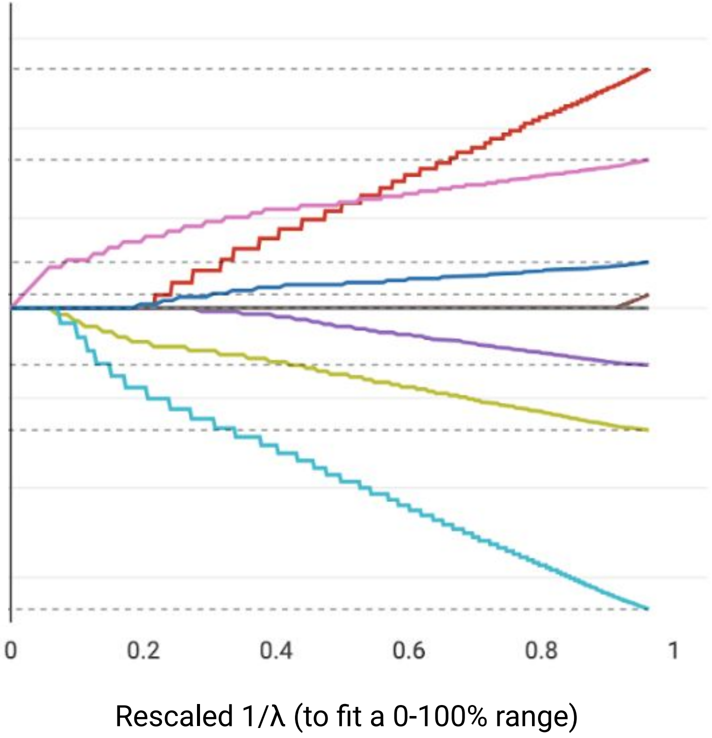
The graph on the right represents the evolution of the estimates **by the number of trees.**



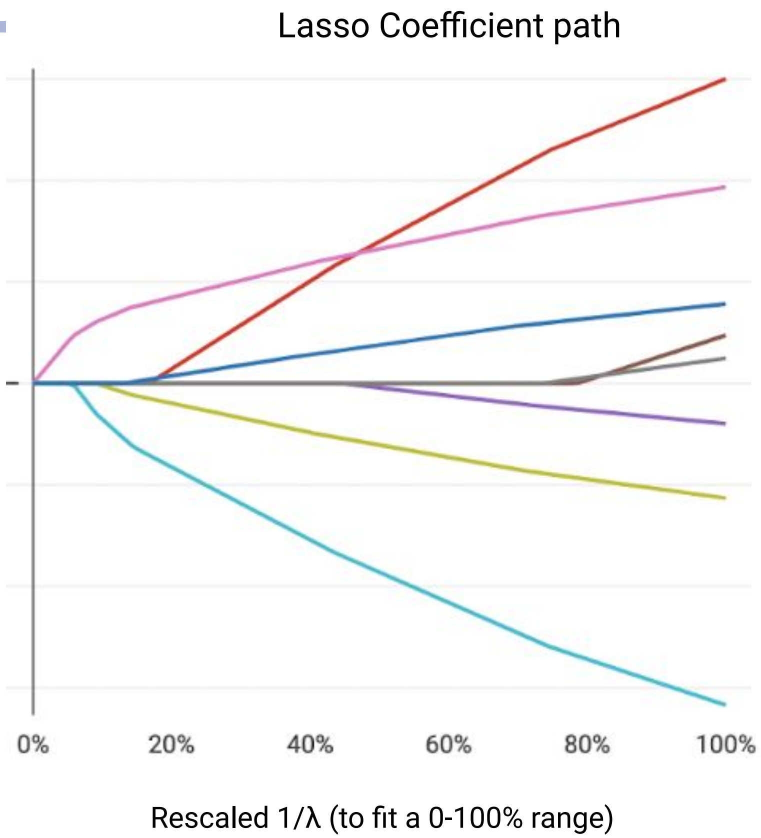
Toward the coefficient path graph

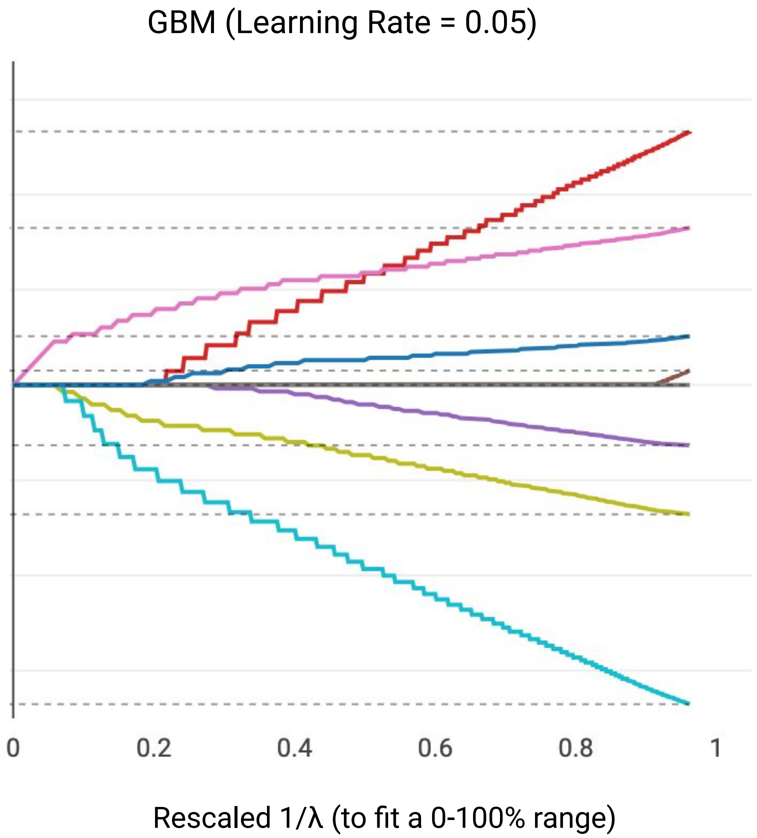
The graph on the right represents the evolution of the estimates **by the number of trees.**

The same graph can be represented by rescaling the x-axis in the same scale as in penalized regression (to fit a 0-100% range).



Comparing Lasso and GBM





Boosting converges to the Lasso

The convergence of boosting toward Lasso solution is a proven mathematical result.

1. *GBMs provide a good approximation of a Lasso regression*

2. *Both GBMs and Lasso allow to tune a parameter in order to control the training error and ability to generalise:*
   a. *GBMs via the combination of **number of trees - learning rate** (and **many** other tree-related parameter)*

   b. *Lasso via **the smoothness parameter***

ASTIN, AFIR/ERM and IACA Colloquia Innovation & Invention

Actuaries Institute

A discussion on credibility and penalised regression, with implications for actuarial work

The Annals of Statistics
2004, Vol. 32, No. 2, 407–499
© Institute of Mathematical Statistics, 2004

LEAST ANGLE REGRESSION

BY BRADLEY EFRON,[1] TREVOR HASTIE,[2] IAIN JOHNSTONE[3] AND ROBERT TIBSHIRANI[4]

Stanford University

The purpose of model selection algorithms such as *All Subsets*, *Forward Selection* and *Backward Elimination* is to choose a linear model on the basis of the same set of data to which the model will be applied. Typically we have available a large collection of possible covariates from which we hope to select a parsimonious set for the efficient prediction of a response variable. *Least Angle Regression* (LARS), a new model selection algorithm, is a useful and less greedy version of traditional forward selection methods. Three main properties are derived: (1) A simple modification of the LARS algorithm implements the Lasso, an attractive version of ordinary least squares that constrains the sum of the absolute regression coefficients; the LARS modification calculates all possible Lasso estimates for a given problem, using an order of magnitude less computer time than previous methods. (2) A different LARS modification efficiently implements Forward Stagewise linear regression, another promising new model selection method; this connection explains the similar numerical results previously observed for the Lasso and Stagewise, and helps us understand the properties of both methods, which are seen as constrained versions of the simpler LARS algorithm. (3) A simple approximation for the degrees of freedom of a LARS estimate is available, from which we derive a C_p estimate of prediction error; this allows a principled choice among the range of possible LARS estimates. LARS and its variants are computationally efficient: the paper describes a publicly available algorithm that requires only the same order of magnitude of computational effort as ordinary least squares applied to the full set of covariates.

1. Introduction. Automatic model-building algorithms are familiar, and sometimes notorious, in the linear model literature: Forward Selection, Backward Elimination, All Subsets regression and various combinations are used to automatically produce "good" linear models for predicting a response y on the basis of some measured covariates $x_1, x_2, \dots, x_m$. Goodness is often defined in terms of prediction accuracy, but parsimony is another important criterion: simpler models are preferred for the sake of scientific insight into the $x - y$ relationship. Two promising recent model-building algorithms, the Lasso and Forward Stagewise lin-

What about ordinal variables ?

What about Ordinal variables ?

The Worker Compensation example highlights the connection between GBMs and Lasso for **categorical variables.**

The main benefit of a GBM is its ability to natively fit **non-linear effect on ordinal variables.**

At a first glance, Penalized Regressions seem unable to natively fit non-linear effects.

We will show that, by analyzing how GBMs incorporate non-linearities, it is possible to incorporate the same learning procedures to Penalized regression.

GBM and Ordinal variables

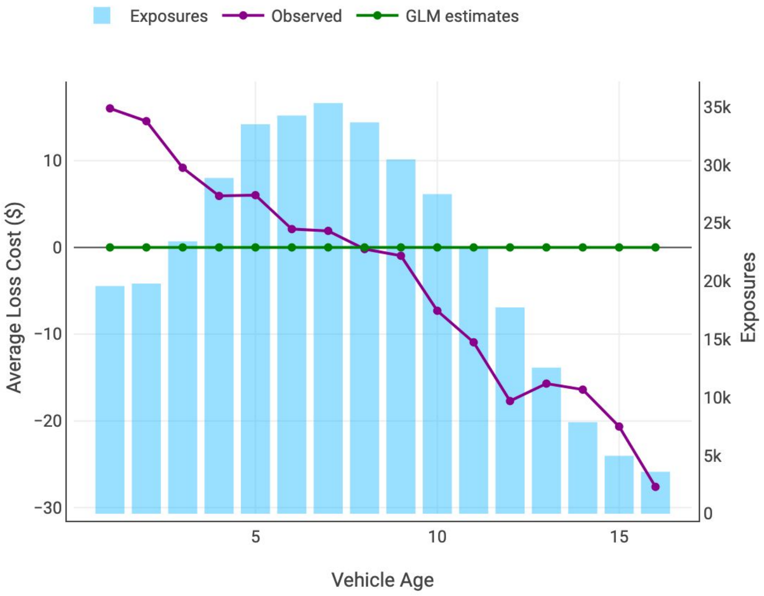
GBMs natively handles **non-linear effects** by combining

1. **Trees**

   Detects the location on where to split the ordinal variables in two region

2. **Boosting**

   Adaptively learns structure from the residuals / errors



GBM and Ordinal variables

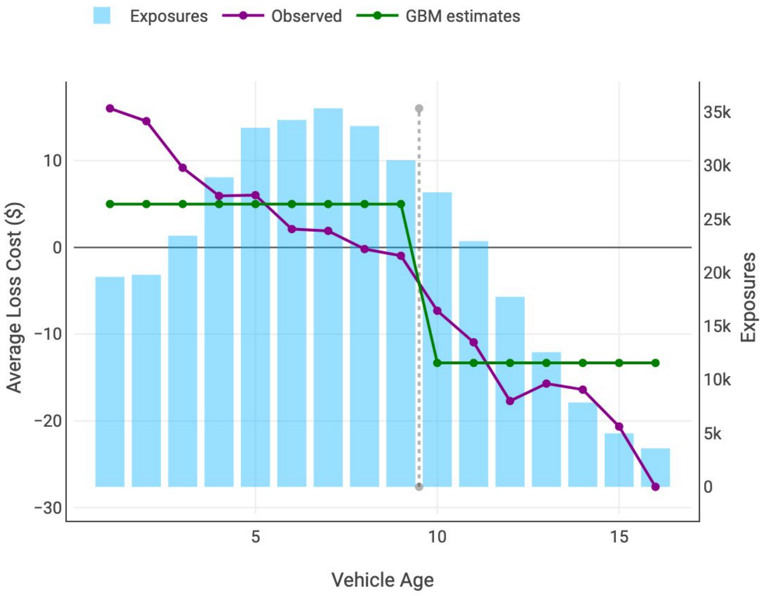
GBMs natively handles **non-linear effects** by combining

1. **Trees**

   Detects the location on where to split the ordinal variables in two region

2. **Boosting**

   Adaptively learns structure from the residuals / errors



Lasso and Ordinal variables

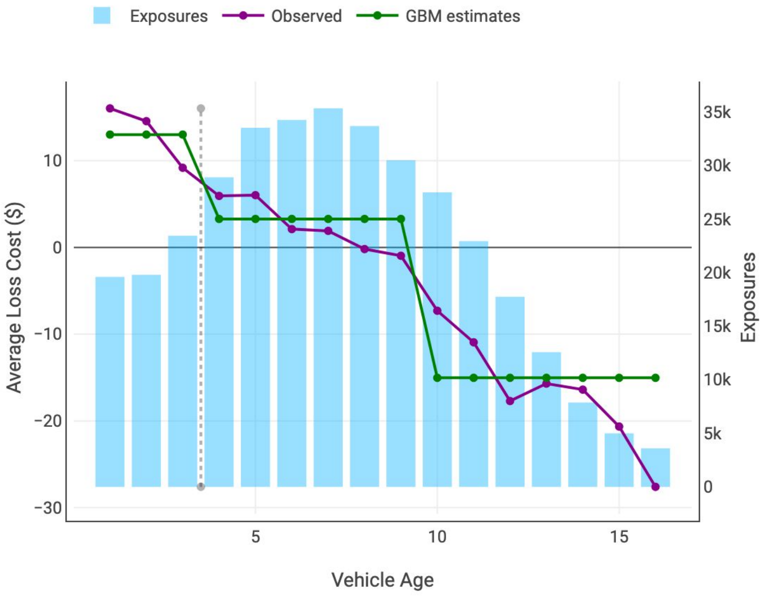
GBMs natively handles **non-linear effects** by combining

1. **Trees**

   Detects the location on where to split the ordinal variables in two region

2. **Boosting**

   Adaptively learns structure from the residuals / errors



Lasso and Ordinal variables

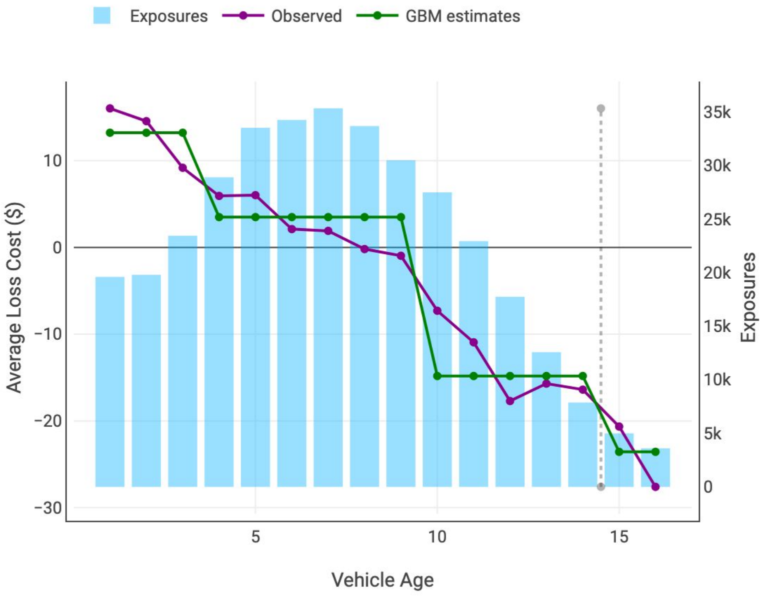
GBMs natively handles **non-linear effects** by combining

1. **Trees**

   Detects the location on where to split the ordinal variables in two region

2. **Boosting**

   Adaptively learns structure from the residuals / errors



Lasso and Ordinal variables

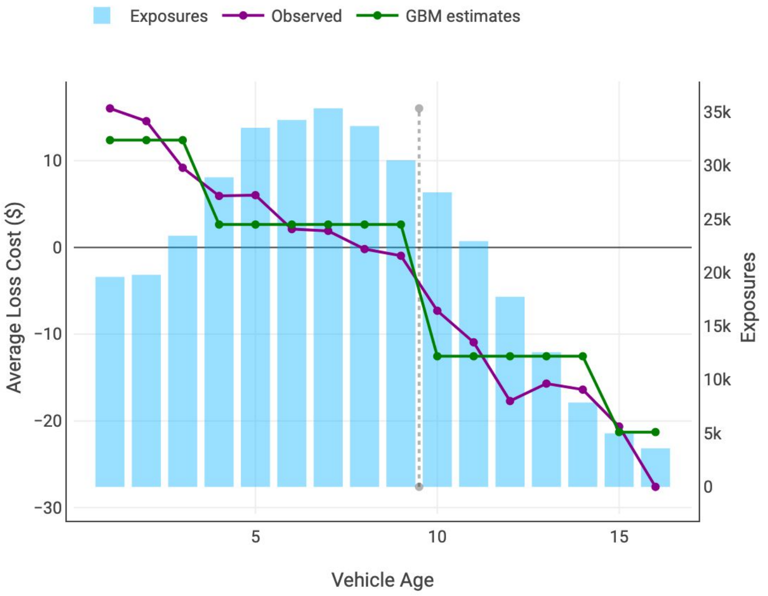
GBMs natively handles **non-linear effects** by combining

1. **Trees**

   Detects the location on where to split the ordinal variables in two region

2. **Boosting**

   Adaptively learns structure from the residuals / errors



Lasso and Ordinal variables

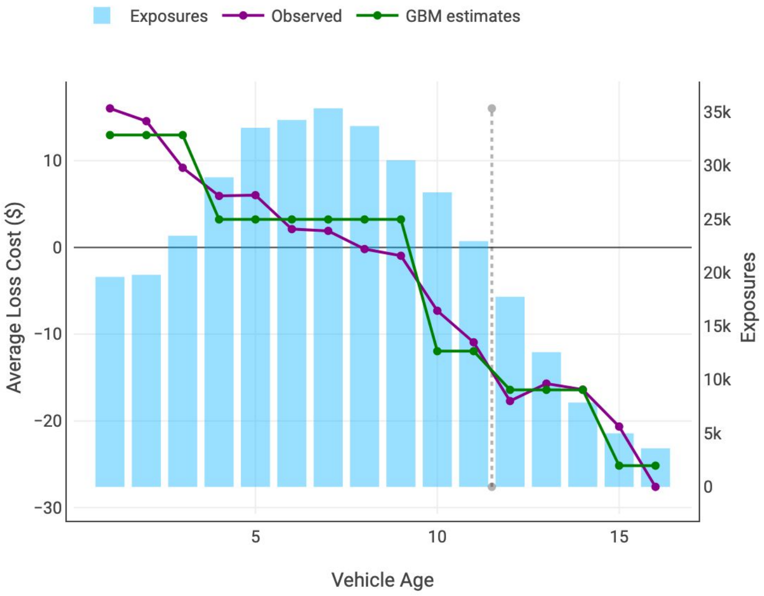
GBMs natively handles **non-linear effects** by combining

1. **Trees**

   Detects the location on where to split the ordinal variables in two region

2. **Boosting**

   Adaptively learns structure from the residuals / errors



Lasso and Ordinal variables

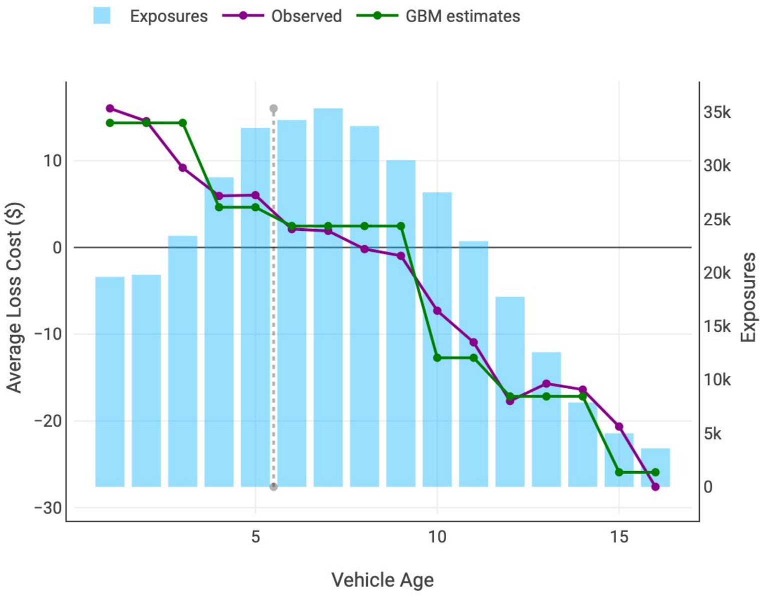
GBMs natively handles **non-linear effects** by combining

1. **Trees**

   Detects the location on where to split the ordinal variables in two region

2. **Boosting**

   Adaptively learns structure from the residuals / errors



Lasso and Ordinal variables

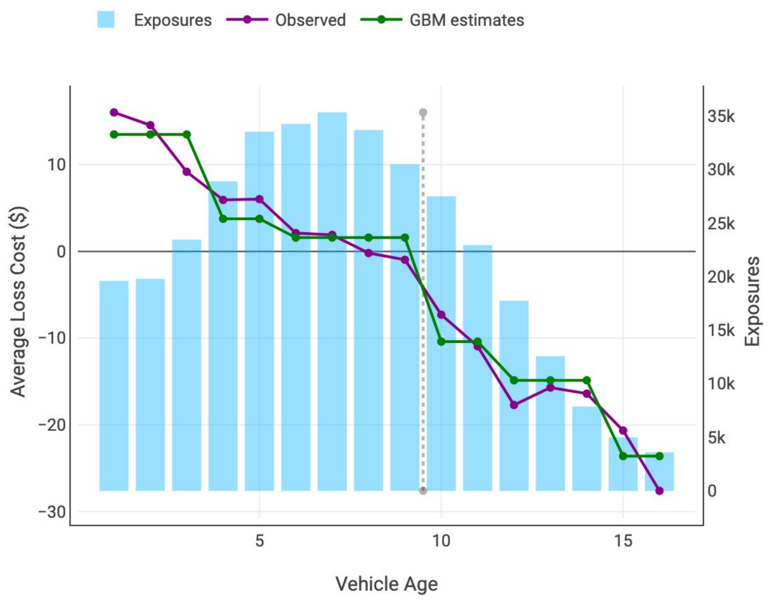
GBMs natively handles **non-linear effects** by combining

1. **Trees**

   Detects the location on where to split the ordinal variables in two region

2. **Boosting**

   Adaptively learns structure from the residuals / errors



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

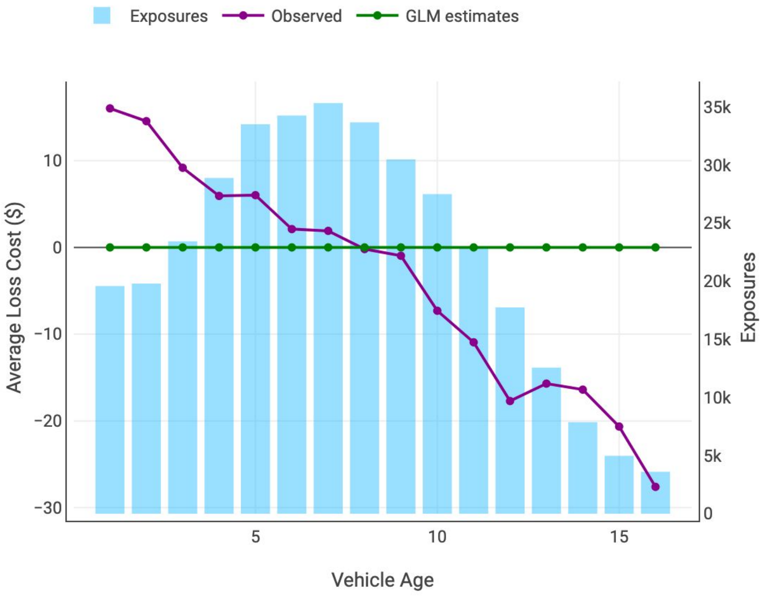
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model ‘smoother’ and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

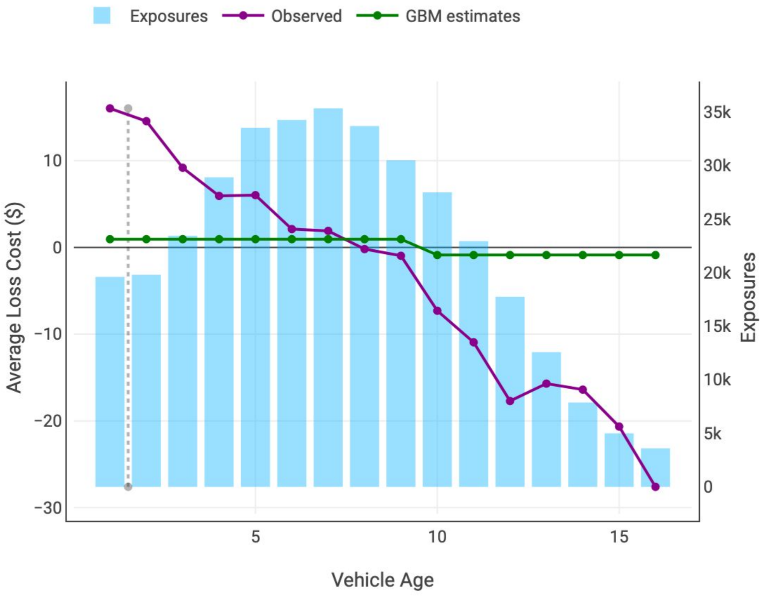
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model ‘smoother’ and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

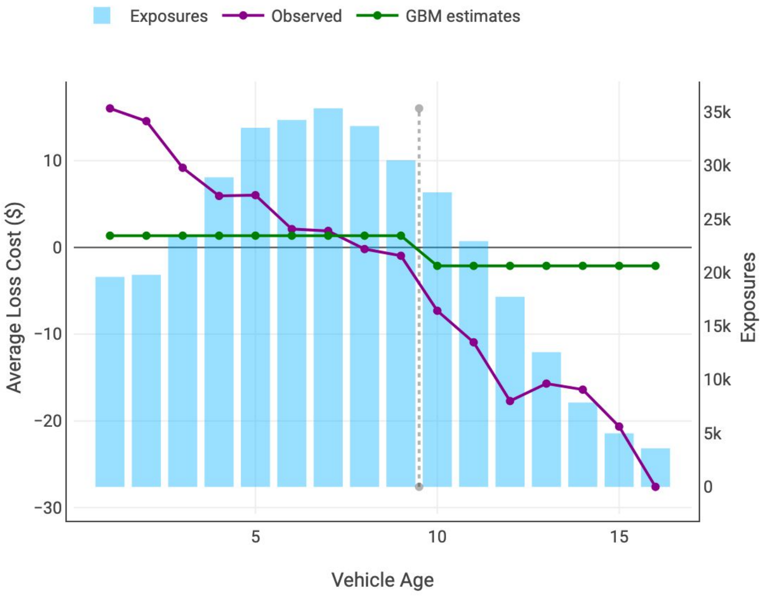
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model ‘smoother’ and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

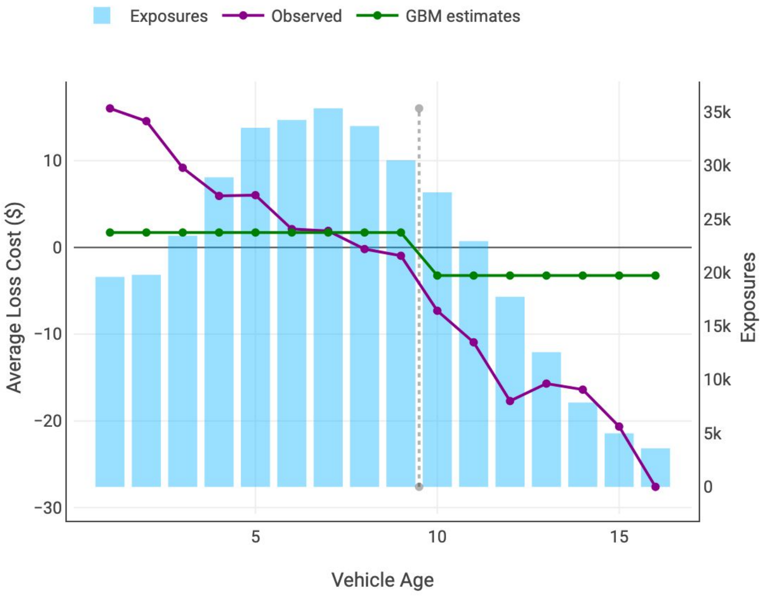
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model ‘smoother’ and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

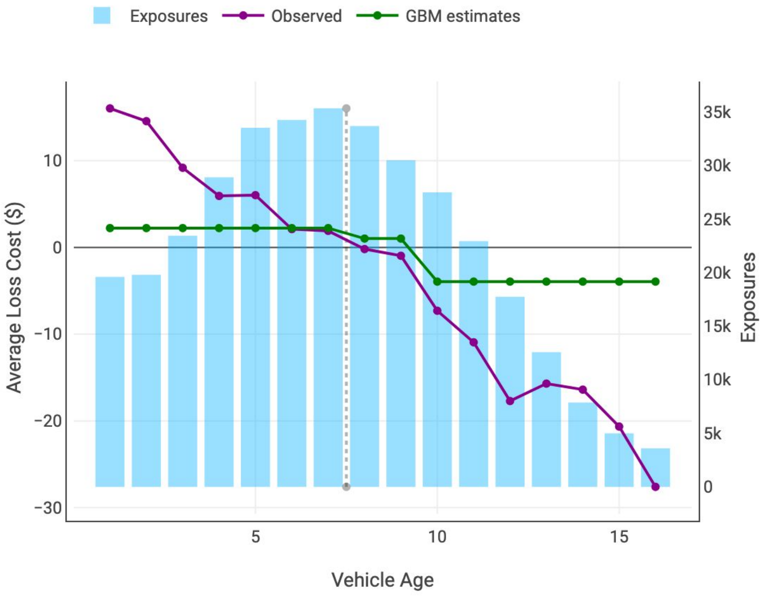
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model ‘smoother’ and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

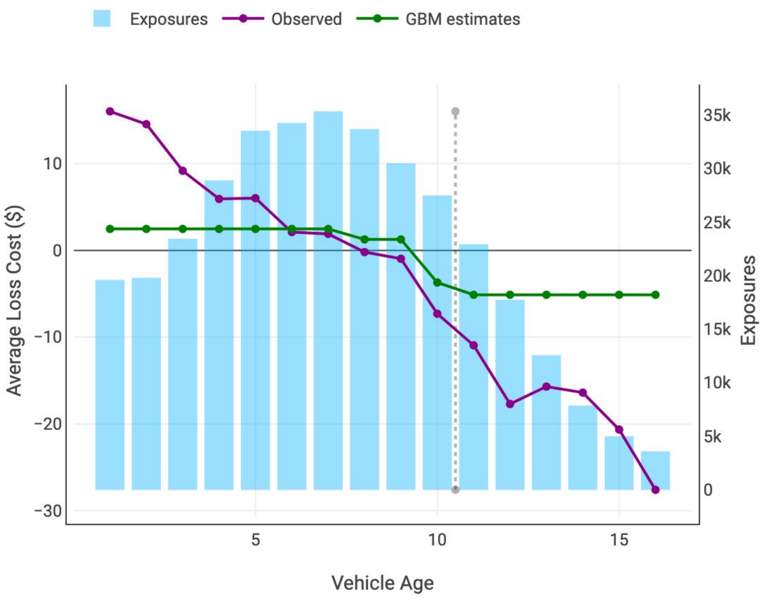
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model 'smoother' and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

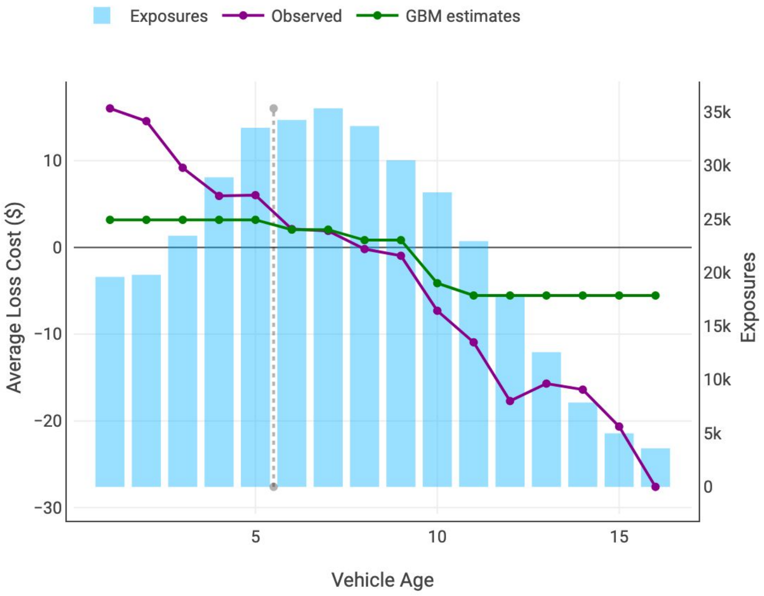
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model 'smoother' and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

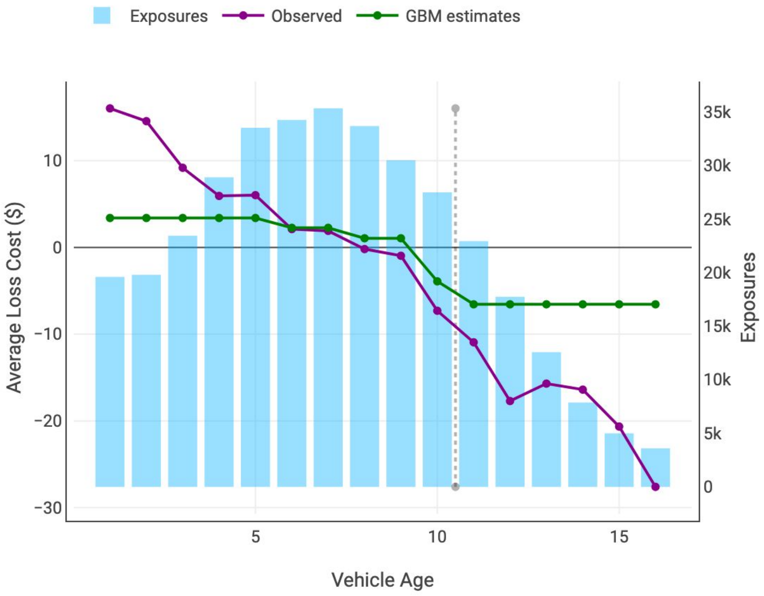
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model 'smoother' and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

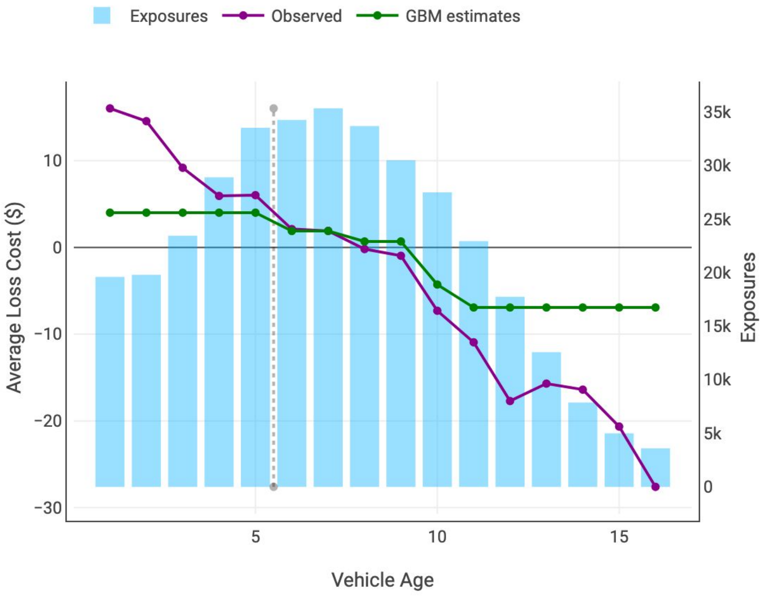
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model ‘smoother’ and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

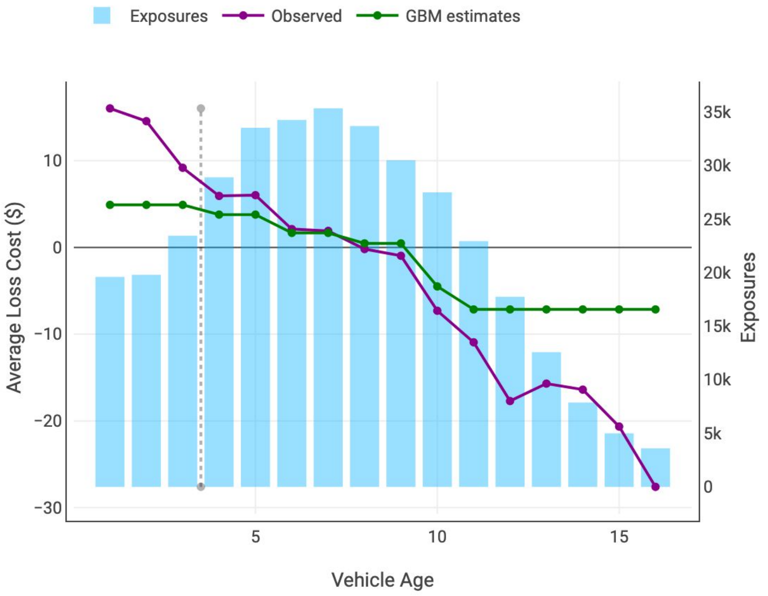
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model ‘smoother’ and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

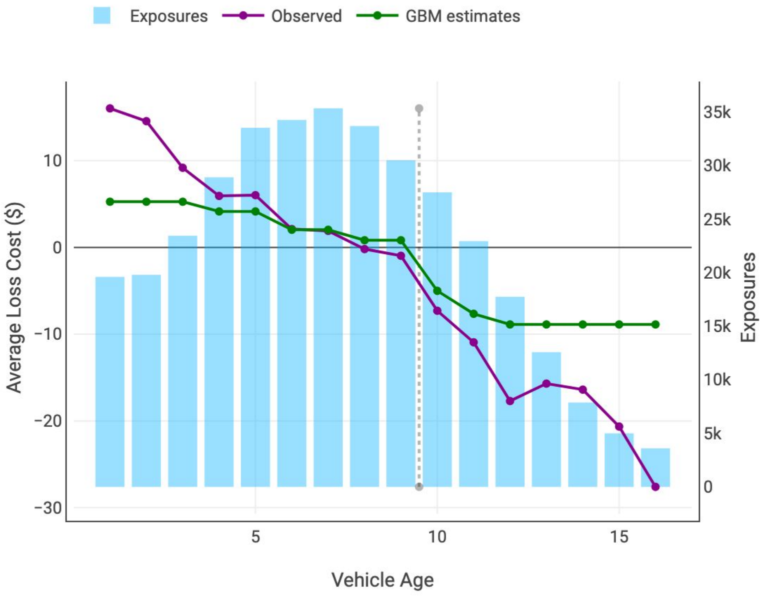
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model 'smoother' and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

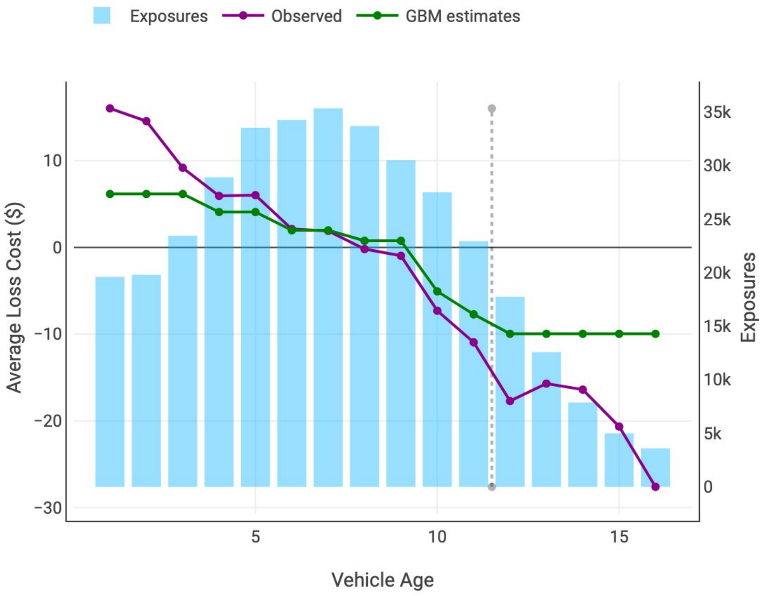
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model ‘smoother’ and more robust to correlations



The impact of the Learning Rate

GBMs natively handles **non-linear effects** by combining

1. Trees

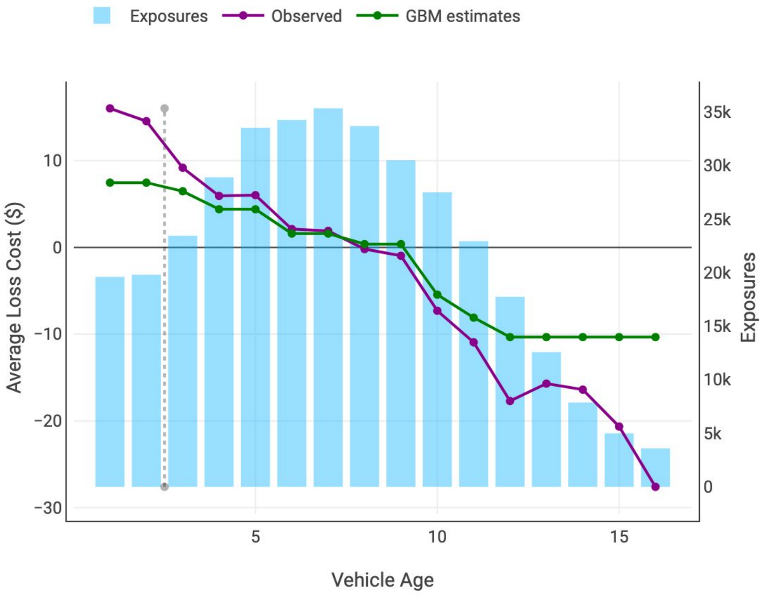
   Detects the location on where to split the ordinal variables in two region

2. Boosting

   Adaptively learns structure from the residuals / errors

3. **Learning Rate**

   Allows to incrementally adapt the trees to the signal, making the model ‘smoother’ and more robust to correlations



How GBMs ‘learn’ ordinal variables

These visual examples highlight **how** GBM effectively learn non-linearities:

1. The most significant split (the ‘derivative’) is computed.
2. The learning rate defines the amount of signal to be learnt (hence controlling for **smoothing**).
3. The number of trees defines the stopping point to prevent overfitting.

The same structure can be replicated in Penalized Regression by using an appropriate **prior distribution** (or **penalty**)

Creating new Priors and Penalties

Grouping is statistically equivalent to the assumption that the coefficients of two consecutive levels

- **are more likely to be close than far apart** if they are significantly different.a

- or **have the same coefficients** if they are not significantly different…

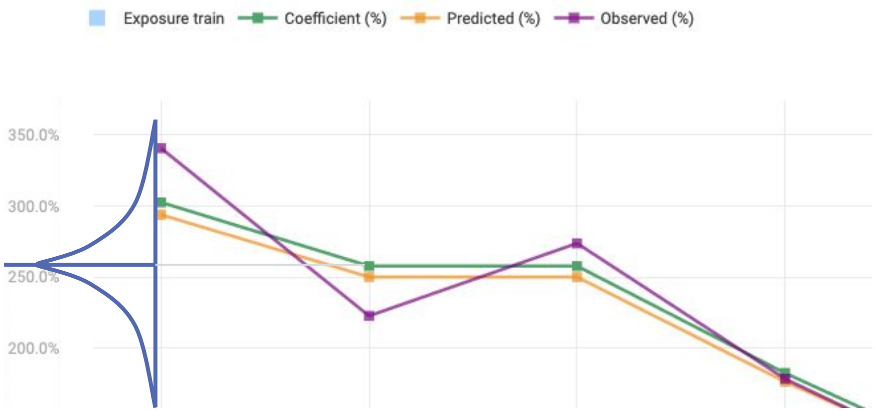
Creating new Priors and Penalties

This means that the **derivative of the (ordinal) variable) follows a Laplace distribution**:

As the values of the coefficients are discrete, the derivative can be written as:

$$p(\beta) \; \alpha \; e^{-\lambda \, |\beta_i - \beta_{i+1}|}$$

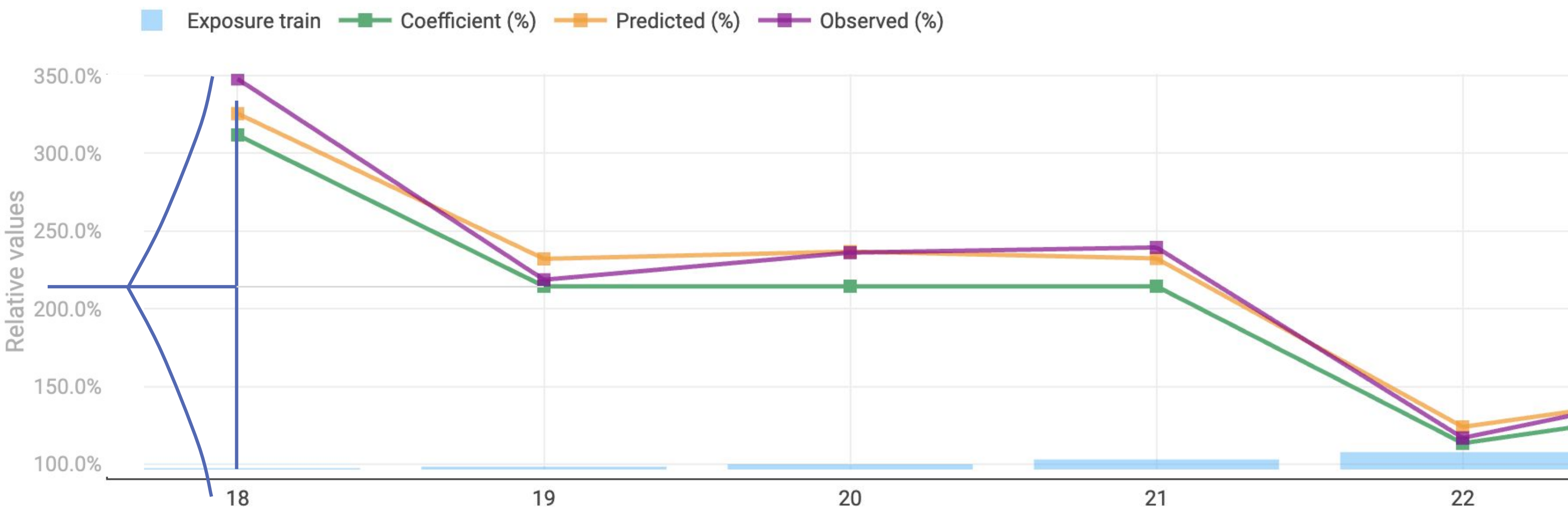
This distribution of probability is used as a prior when maximizing the likelihood to fit a model:



$$\beta^* = Argmax_\beta \;\; LL(x, y, \beta) - \lambda \, |\beta_i - \beta_{i+1}|$$

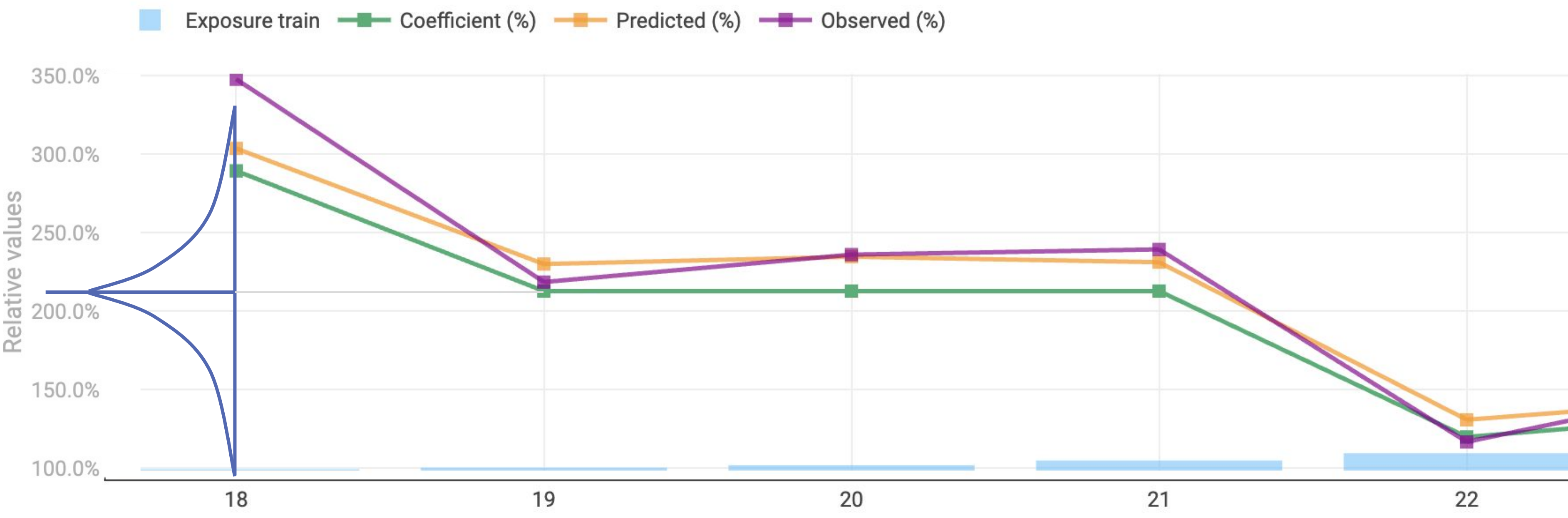
Weak Smoothness ⇔ Strong reliance on the observation

The prior has a very limited impact on the final model



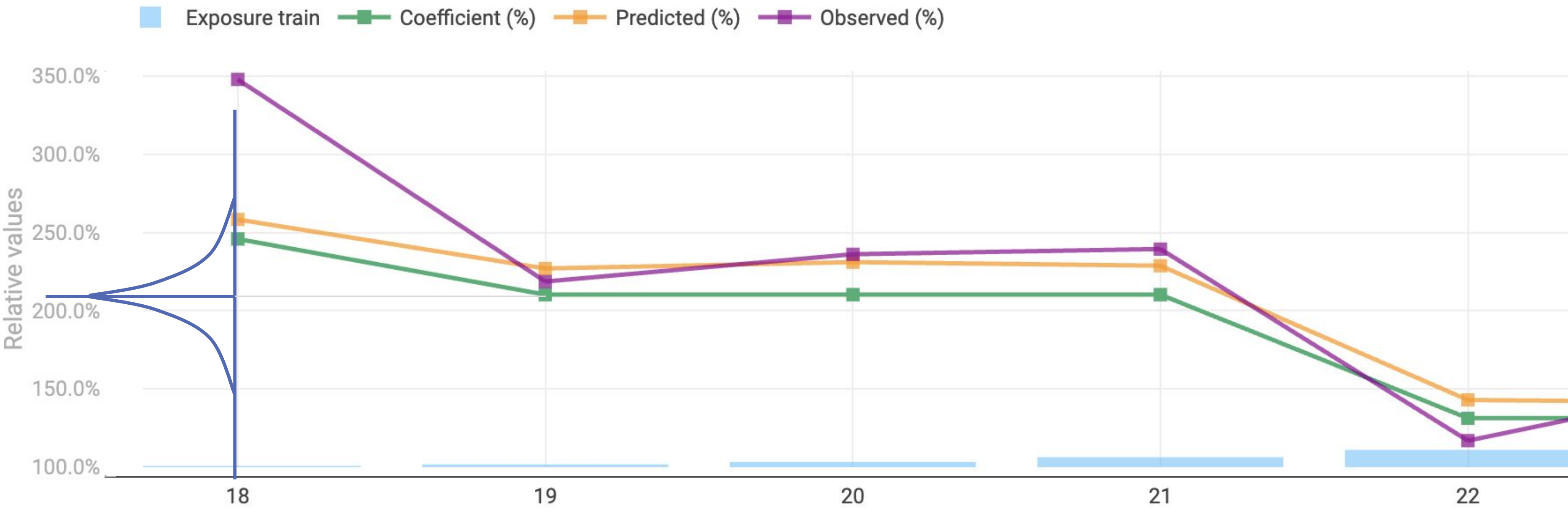
Stronger Smoothness ⇔ Weaker reliance on the observation

The final model is an average between the most likely coefficients according to the prior and the observations

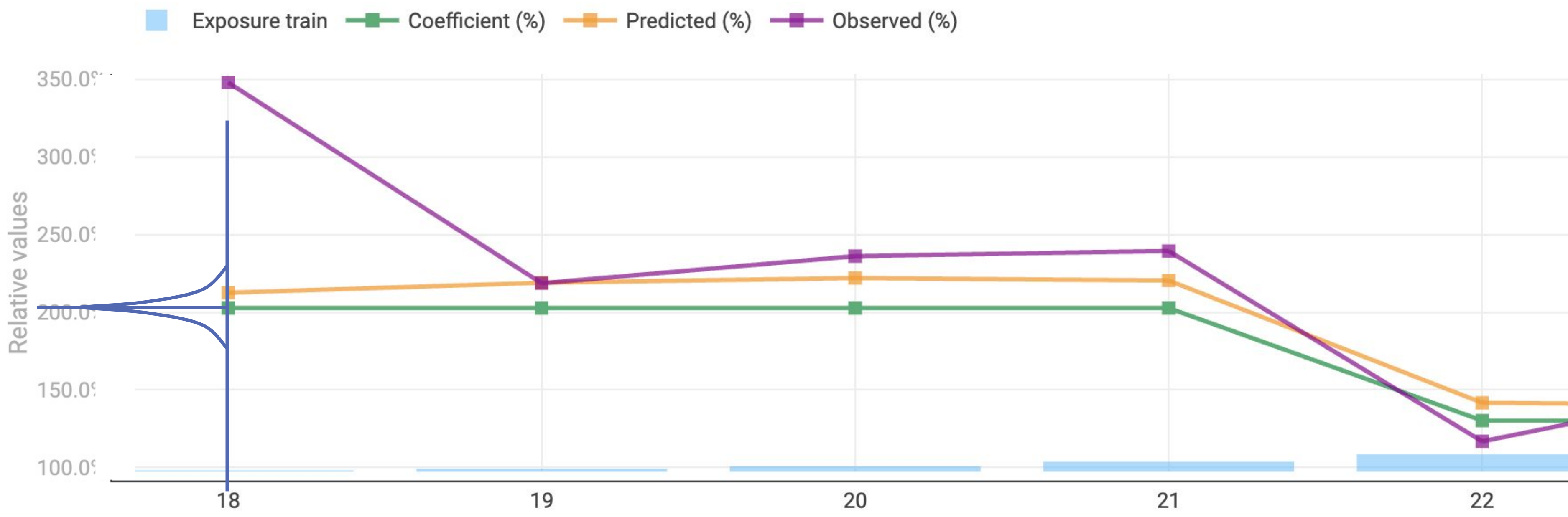


Strong Smoothness ⇔ Very weak reliance on the observation

The weight of the observation in the model is weaker than the priors



Very Strong Smoothness ⇔ Full reliance on the prior



Lasso and Ordinal variables

Under these **“Lasso”** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

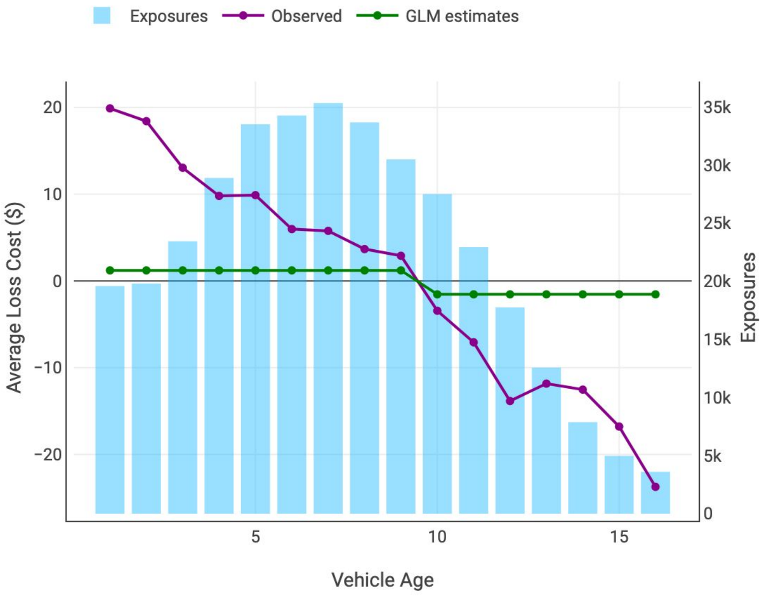
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

    - **number of trees**
    - **learning rate**
    - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

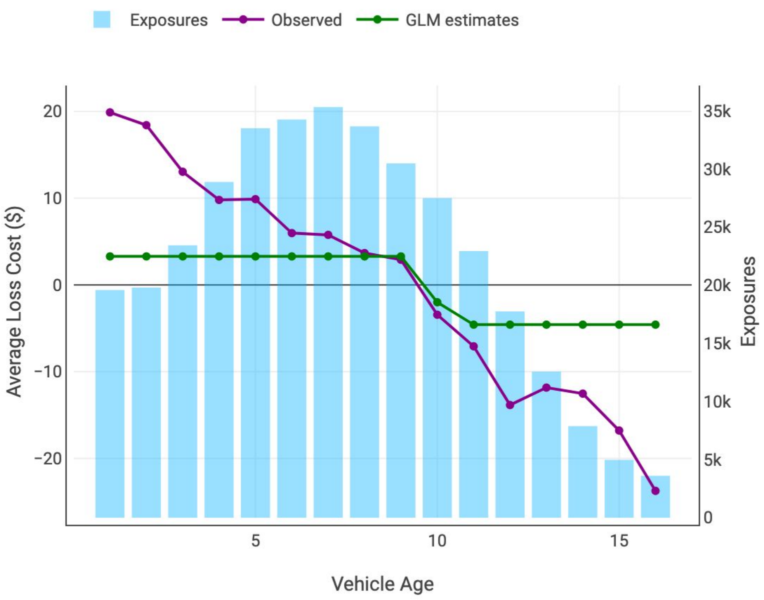
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

    - **number of trees**
    - **learning rate**
    - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

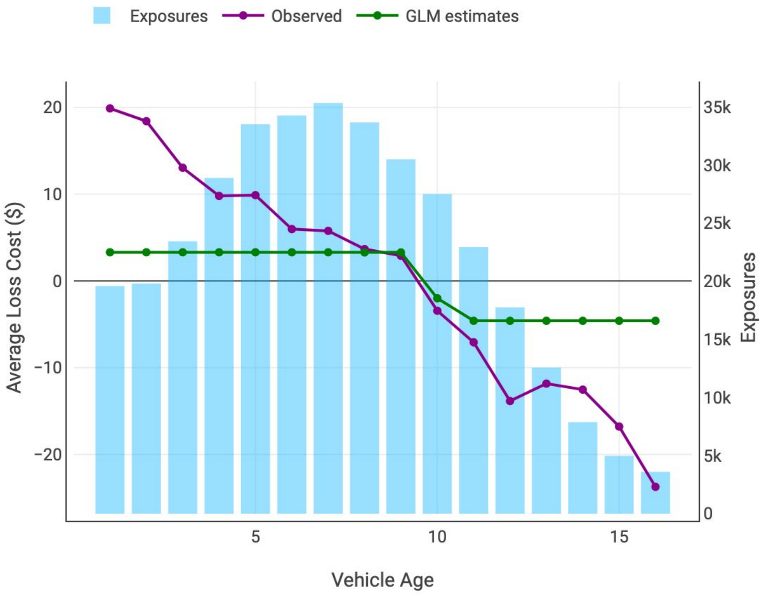
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

    - **number of trees**
    - **learning rate**
    - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

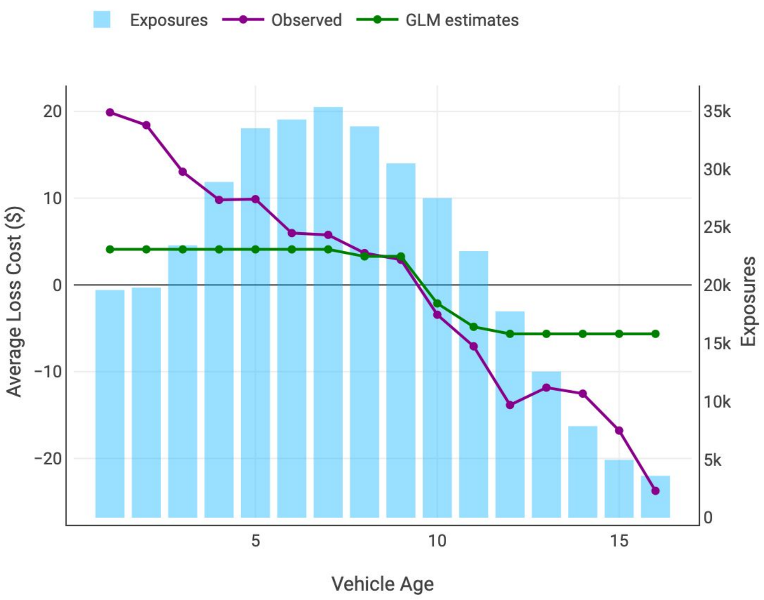
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

    - **number of trees**
    - **learning rate**
    - and other tree-related parameters



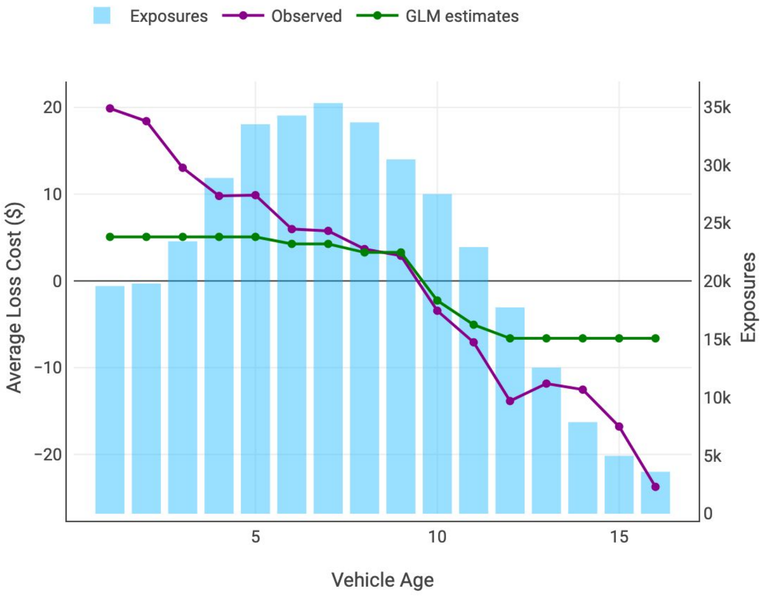
Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**
- GBMs require to determine the combination of **several parameters:**
    - **number of trees**
    - **learning rate**
    - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

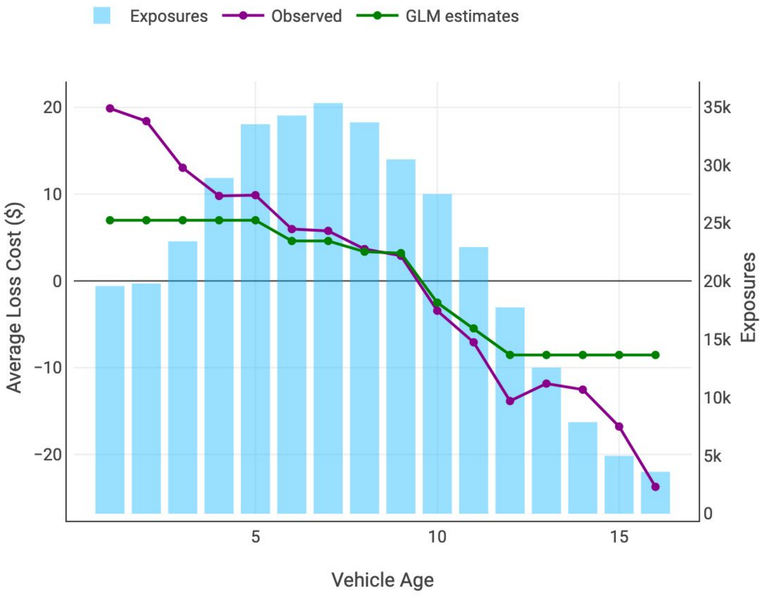
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

    - **number of trees**
    - **learning rate**
    - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

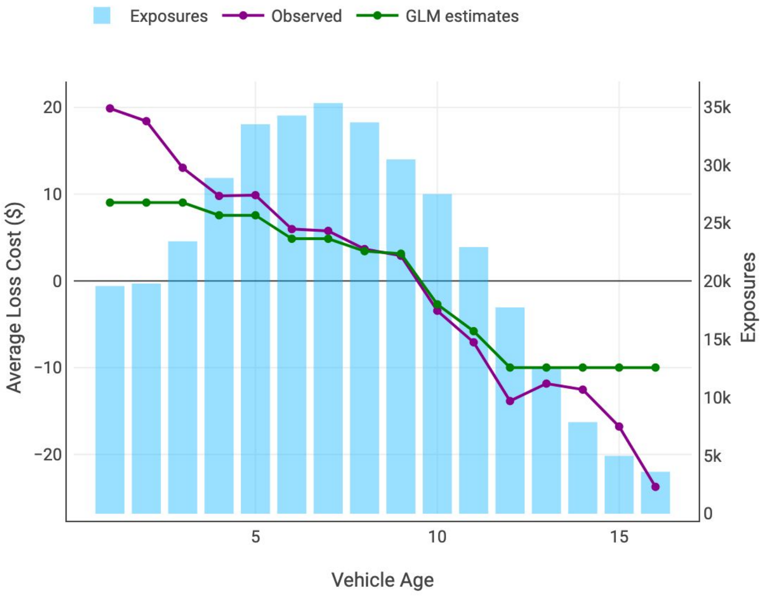
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

    - **number of trees**
    - **learning rate**
    - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

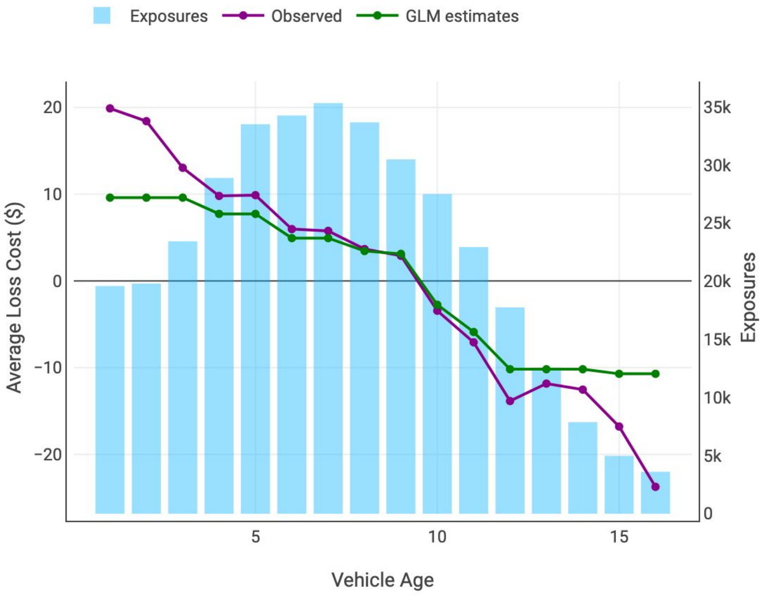
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

    - **number of trees**
    - **learning rate**
    - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

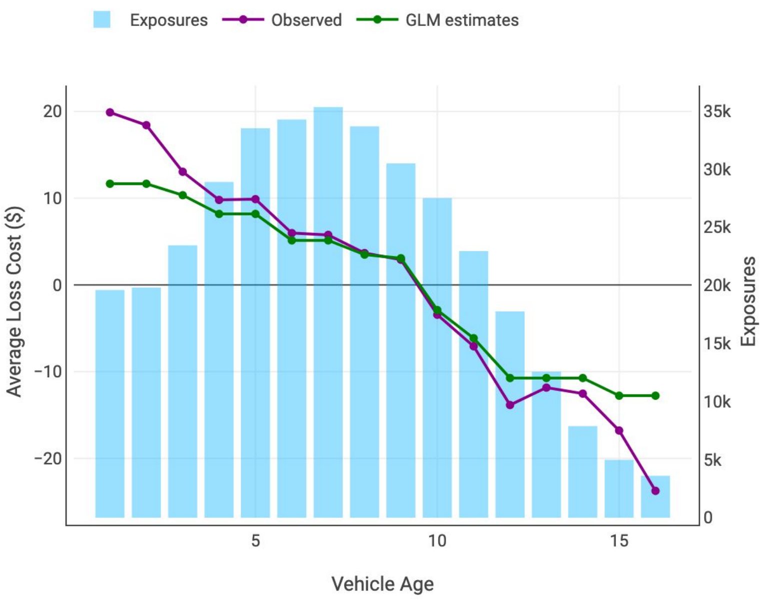
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

    - **number of trees**
    - **learning rate**
    - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

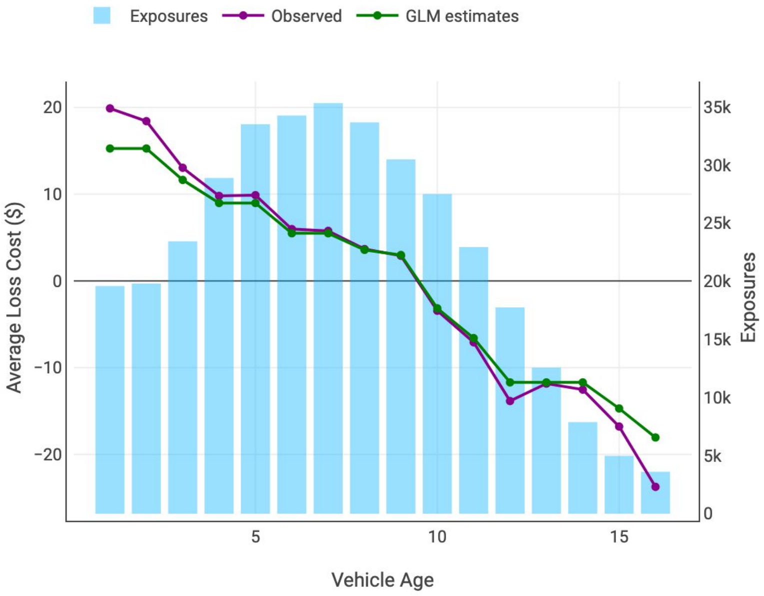
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

  - **number of trees**
  - **learning rate**
  - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

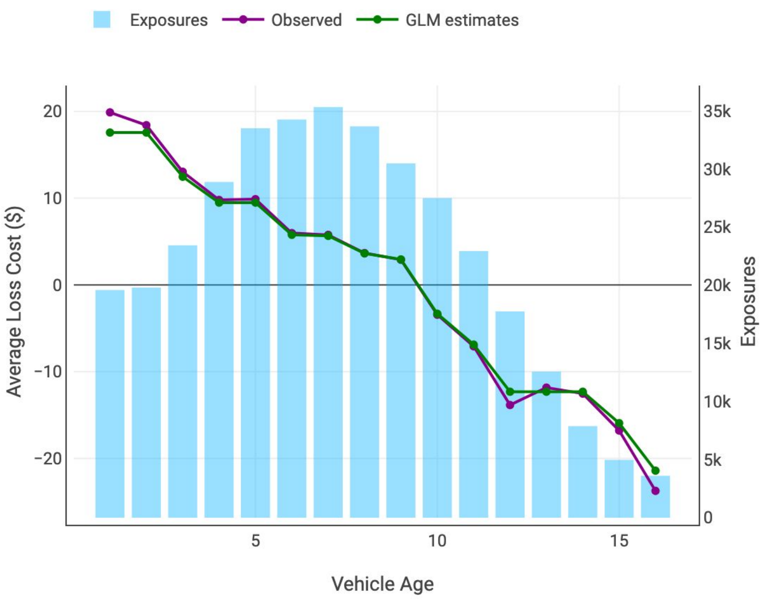
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

  - **number of trees**
  - **learning rate**
  - and other tree-related parameters



Lasso and Ordinal variables

Under these **"Lasso"** assumption on the **derivative**, penalized regression can **natively incorporate non-linear effects**.

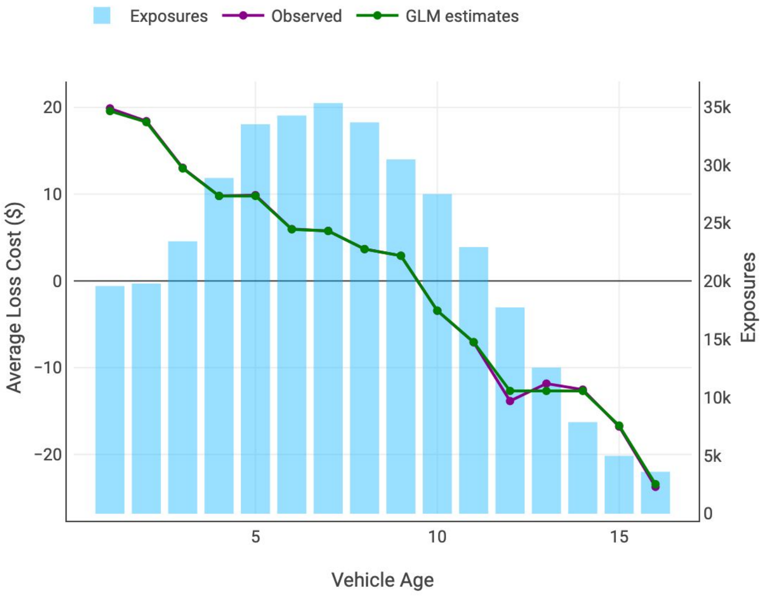
Furthermore, the convergence result between GBMs and Lasso is still valid.

To control the training error and ability to generalise:

- Penalized Regression require the definition of a **single parameter**: the **smoothness**

- GBMs require to determine the combination of **several parameters:**

  - **number of trees**
  - **learning rate**
  - and other tree-related parameters



Conclusion

Comparing GBM and Penalized Regression

	Lasso Regression	GBM	Derivative Lasso - AGLM
Control low-exposure segments to prevent overfitting	All the techniques presented today aim at controlling overfitting.		
Works for multivariate models	Yes ; apply the same priors / rules for all levels		
Creates transparent models (GLM or additive models)	Designed for the GLM framework	No - Output usually not transparent	Designed for the GLM framework
Natively manage non-linear effects	No - Requires non-linearities to be explicitly specified	Yes	

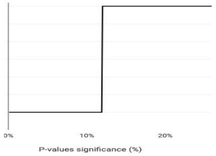
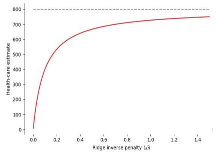
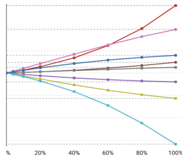
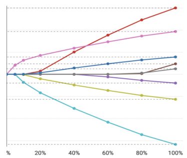
Conclusion

Penalized regression offers a **flexible and theoretically sound** framework to tackle and address the GLM's drawbacks.

It does so in an **accessible** way:

- Penalized regression require the choice of **only one parameter: the smoothness**
  - Smoothness relates to known credibility techniques
- Penalized regression require **little to no** investment cost
  - Inputs and outputs are equal to GLMs - adding penalizations to GLM is straightforward via software
- Potentially **unlock use-cases** not previously considered **for modeling**
  - Via complement of credibility, it is possible to gradually update current models to new ones
  - GLMs can be used as a data analysis alternative as modeling effort is reduced since non-linearities are natively handled.

The big picture

<table>
<tr><th></th><th>Levels Selection</th><th>Credibility</th><th>Ridge Regression</th><th>Lasso Regression</th><th>GBM</th><th>Derivative Lasso-AGLM</th></tr>
<tr><td>Control low-exposure segments to prevent overfitting</td><td colspan="6">All the techniques presented today aim at controlling overfitting.</td></tr>
<tr><td>Set coefficients of low-exposure segments at zero</td><td>Selection of effects</td><td colspan="2">No selection of effects</td><td colspan="3">Selection of effects, allowing binary decisions (if the effects are visualized - not always true for GBMs)</td></tr>
<tr><td>Shrink low-exposure segments</td><td>No</td><td colspan="5">This allows to tolerate segments with limited (yet usable) data</td></tr>
<tr><td>Work for multivariate models</td><td>Yes</td><td>No</td><td colspan="4">Yes ; apply the same priors / rules for all levels</td></tr>
<tr><td>Creates transparent models (GLM or additive models)</td><td colspan="4">Designed for the GLM framework</td><td>Usually, output not transparent</td><td>Additive models</td></tr>
<tr><td>Natively manage non-linear effects</td><td colspan="4">These techniques work on “pure GLM” (linear or categorical effects)</td><td colspan="2">Yes</td></tr>
<tr><td>Coefficient depending on the robustness parameter</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

THANKS

9 rue Fortuny, 75017 Paris
FRANCE

AKUR8

The coefficient path graph

How to 'rescale' the impact of the penalty

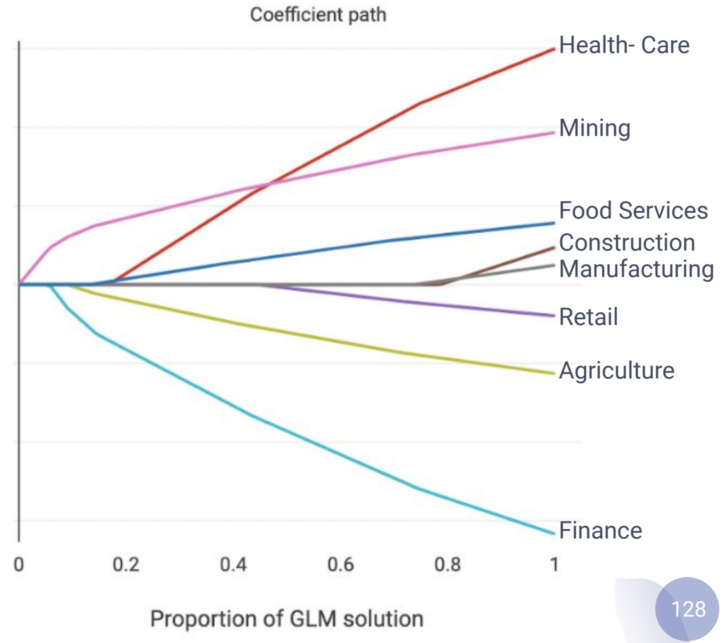
It is possible to generalize this graph, tracking the **impact of penalty on several levels** simultaneously.

The **'coefficient path graph'** allows to globally analyse how the estimates /coefficient evolve when the smoothness increases:

- Y axis represents the value of the estimates.
- X axis represents the 'Empirical Credibility' - which is a 'Proportion of the GLM solution)

$$\text{Empirical Credibility} = \sum_{i \in \text{Classes}} \frac{|\text{Predicted}_i - \text{Grand Average}|}{|\text{GLM}_i - \text{Grand Average}|} \%$$

- Empirical Credibility = 100 % - Estimates match the observed
- Empirical Credibility = 0 % - Estimates match the Grand Average (or complement of credibility)



Is the convergence result a desirable property ?

Smaller learning rate corresponds to better models, but at a cost

In GBMs the smaller the learning rate the better

1. Smaller learning rates lead to more performant and robust models - as they handle better correlations

2. Smaller learning rates require to build many more trees

The only limit of choosing a smaller learning rate in a GBM is the time required to build the models.

Lasso being equivalent to a very little learning rate is a desirable property.