Adjusting Manual Rates to Own Experience: Comparing the Credibility Approach to Machine Learning

Giorgio A. Spedicato, Ph.D FCAS FSA CSPA, Christophe Dutang, Ph.D, Quentin Guibert, Ph.D

21 giugno, 2021

Introduction

The use of market data as an aid for setting own rates has been a common practice in the Insurance Industry. External data, as provided by Insurance Rating Bureaus or Advisory Organizations, may supplement internal ones that may be scarce, unreliable because of a non-representative and/or a too short history, or, non-existing at all, e.g. when entering in a new business lines or territory. The importance of external data to both support adequate rates that preserve company solvency and in easing the entrance of new players has been historically recognized by regulator, e.g. granting a partial antitrust − law exception in the US jurisdiction (Danzon 1983).

When the Insurer takes into account its own experience in order to enhance the credibility of its rates, it need to assess if and in which manner its portfolio experience departs from the market one. The actuarial profession traditionally used techniques based on Bayesian statistics and non-parametric credibility to optimally combine the market and insurer's portfolio experience in the technical rates. These early models were not based on policyholders' ratemaking variables, yet some advanced regression credibility models have been proposed in the actuarial literature such as the Hachemeister model, see e.g. (Bühlmann and Gisler 2006). On the contrary, rates based on Generalized Linear Models (GLM), the current gold standard in personal rates pricing (Goldburd, Khare, and Tevet 2016), are only based on the impact of ratemaking factors, giving no credit to the individual policy experience. Nevertheless, mixed effects GLMs allow to incorporate policyholders' experience within GLM tariff structure ("Bayesian credibility for GLMs" 2018; Antonio and Beirlant 2007) but they are not widespread used.

The recent widespread/massive usage of Machine Learning has provided many more techniques to the practitioner actuaries' toolset. In particular, the Gradient Boosting Models (GBM) and the Deep Learning (DL) frameworks can be used in a manner that permits to "transfer" what the model has learnt on a much bigger data set (as the market data) to a smaller set (the portfolio data of the company). "Transfer learning" is typically in computer vision DL modes to fine tune standard architecture on specific recognition tasks. An "initial score" can be provided to GBMs to take into account the known effect of exposure or a-priori modeled estimate before "boosting" the prediction. As ML can easily handle complex non − linear relationship, it is reasonable to believe that they could assess the policyholders' risk more precisely than standard credibility or GLMs based approaches. The proposed research aims to contrast traditional methods to ML ones in the task of blending market data to individual portfolio experience. After a brief business and methodological introduction, we will apply on a (properly anonymized) data set comprised by both market and own portfolio experience relative to a European country non − life business line. The final comparisons will consider not only on the predictive performance, but also in the ease of practical application in term of computational request, ease of understanding and interpretability of the results.

Datasets structure

Two (anonymized) dataset were provided, one for the marketwide (`"mkt_anonymized_data.csv"`) and one for the company (`"mkt_anonumized_data.csv"`), henceforth MKT and CMP datasets. The datasets share the same structure, as each company provides its data in the same format to the Pool, that aggregates individual filings into a marketwide file, that is provided back to the companies. The dataset contains exposures and claims aggregated by some classification variables. Variable names, levels and numeric variable distribution have been masked and anonymized for privacy and confidentiality purposes.

maybe a graphic could be done?

The following variables are contained in the provided data set:

- *exposure*: the insurance exposure measures by classification group, on which the rate is filled (aggregated outcomes);
- *claims*: the number of claims by classification group (aggregated outcomes);
- *ID*: unique row number (helper variable);
- *zone_id*: territory (aggregating variable);
- *year*: filing year (aggregating variable);
- *group*: random partition of the dataset into train, valid and test set.
- *cat1*: categorical variable 1, available in the original file (aggregating variable);
- *cat2*: categorical variable 2, available in the original file (aggregating variable);
- *cat3*: categorical variable 3, available in the original file (aggregating variable);
- *cat4-cat8*: categorical variables related to the territory (joined to the original file by zone_id);
- *cont1-cont12*: numeric variables related to the territory (joined to the original file by zone_id);

Categorical and continuous variables have been anonymized by label encoding and scaling (calibrated on market data). In addition, the last available year (2008) has been used as test set, while data from precedent years have been randomly split between train and validation sets on a 80/20 basis.

Market data is available for 11 years, while company data for the last five one. Also, the number of exposures is widely dependent by

Modeling approach

The research aims to compare the predictive power of traditional and machine learning methods that use an initial estimate of loss costs, e.g. from market experience, to predict those of a smaller portion (a given company ones) in a subsequent period (the test set). Therefore, the modeling process aims to predict the losses on the last available year (the test set) training models on the data from the previous years (that have been eventually split into a train and validation test).

The empirical data available for the study regards a risk for which year to year volatility may be relevant due to the sensitivity on local weather. At this regards, the performance assessment has considered not only the discrepancy bet

The losses are the number of damaged units while the exposure are the number of insured units. Therefore only the frequency component has been modeling, choosing either a binomial or a Poisson loss function. Henceforth losses in this paper shall be considered as synonym of claim number.

We are using models that permits to use an initial estimate of losses performed on another set (transfer learning). While the paper explores the use of such approach applying ML methods, traditional GLM may be used as well. E.g. under a log-linear regression framework and initial log-estimate of either the frequency, the severity of the pure - premium may be set as an offset (Yan et al. 2009) for a subsequent model.

Machine learning techniques

ML methods have been acquiring increasing attention by actuarial practitioners especially. Beginning from the analysis of policyholders' behavior (Spedicato, Dutang, and Petrini 2018), several applications have sprung also for risk pricing. An application of boosting techniques to estimate the frequency and the severity of an MTPL dataset can be found in Noll (Noll, Salzmann, and Wuthrich 2020), while Schelldorfer and Wutrict presented a joint model that boosts GLM performance using Deep Learning (Schelldorfer and Wuthrich 2019). ML methods used in insurance pricing are strongly non - linear and are able to automatically find interactions among ratemaking factors and exclude non relevant features. In particular two techniques are acquiring widespread importance: Boosting and Deep learning. Both techniques allow the use of an initial estimate of loss / exposure to risk

Brief overview

Boosting techniques

The boosting approach can be synthesized by the following formula:

$$F_t\left(x\right) = F_{t-1}\left(x\right) + \eta * h_t\left(x\right)$$

that is, the prediction at the t-it step is given by the contribution, to the prediction of the previous step, of a weak predictor $h_t\left(x\right)$, properly weighted by a learning factor η, being x the covariate vector. The most common choice for the weak predictor $h_t\left(x\right)$ lies in the classification and regression trees family, from which the Gradient Boosted Tree (GBT) models. It can be shown that "boosting" weak predictors lead to very strong predictive models (Elith, Leathwick, and Hastie 2008). Almost all winning solutions of data science competitions held by Kaggle are at least partially based on XGBoost (Chen and Guestrin 2016), the most famous GBT model. More recent and interesting alternatives to be tested are: LightGbm (Ke et al. 2017), which is particularly renowned for its speed, and Catboost (Prokhorenkova et al. 2017), which has introduced an efficient solution for handling categorical data.

A set of hyperparameter defines a boosted model and even more define a GBT one. The core hyperparameters that influence the boosting part are the number of models (trees), $t = 1, 2, \ldots, T$ (typically between 100 and 1000) and the learning rate η, whose typical values lies between 0.2 and 0.001. $h_t\left(x\right)$ can be, when it belongs to the CART family, the maximum depth, the minimum number of observation in final leafs, the fraction of observation (rows or columns) that are considered when growing each tree. The optimal combination of hyperparameters is learn using either a grid search approach or a more refined one (e.g. bayesian optimization).

When applied to claim frequency prediction, they are fit to optimize a Poisson log-loss function. In addition, to handle uneven risk exposure, the log - measure of exposure risk is given (in log scale) as an init-score $(F_t\left(x\right))$ to initialize the learning process. The init-score (or base margin) in the boosting approach has the same role of the traditional GLM offset term (Goldburd, Khare, and Tevet 2016).

Deep Learning

An artificial network is a mathematical structure that applies a non linear function to a linear combination of inputs, say $\phi\left(\bar{x}_i^T * \bar{w} + \beta\right)$, being $\bar{w}$ and β the weights and bias respectively. A neural network consists in one o more layer of interconnected neurons, that receives a (possibly multivariate) input set and retrieves and output set. Modern Deep Neural Networks are constructed by many (deep) layers of neurons. Deep Learning has been knowing a hype in interest for a decade, thanks to the availability of huge amount of data, computing power (in particular GPU computing) and the development of newer approaches to reduce the overfitting that had halted the widespread adoption of such techniques in previous decades (Goodfellow, Bengio, and Courville 2016). Different architectures has reached state of the art performances in many fields; e.g. convolutionary neural networks achieved top performance in computer vision (e.g. image classification and object detection), while recurrent neural networks (e.g. Long Short Term Memory ones) provides excellent results in Natural language processing tasks like sequence-to-sequence modeling (translation) and text classification (sentiment analysis).

Simpler structure are needed for a claim frequency regression, the multi-layer perception (MLP) architectures that basically consist in stacked simple neurons layers, from the input one to the single output cell one. This structure is dealt to handle the relation between the relation between the ratemaking factors and the frequency (the structural part). To handle the different exposures, the proposed architecture is based on the solution presented by (Ferrario, Noll, and Wuthrich 2020; Schelldorfer and Wuthrich 2019). A separate branch collects the exposure, applies a log-transformation, then this exposure is added in a specific layer just before the final one (that has a dimension of one).

Training a DL model consists in providing batches of data to the network, evaluate the loss performance and updating the weights in the direction that minimize the training (backpropagation). The whole data set is provided to the fitting algorithms many times (epochs) split in batch. One of the common practice to avoid overfitting is to use a validation set where the loss is scored at each epoch. When it starts to systematically diverge, the training process is stopped (early stopping).

TODO

Credibility-based models

TODO

Notation

Bühlmann-Straub Credibility Model

Application

The analysis was performed on the two data sets, the CMP and MKT, preprocessed and split into train, validation and test set as previously discussed. Then, the models were fit on the train set and the predictive performance was assessed on the test set. The validation set was used in DL and BST models to avoid overfitting.

Finally, the models are compared in terms of predictive accuracy, using the (The actual / predicted ratio) and risk classification performance, using the Normalized Gini index (Frees, Meyers, and Cummings 2014). The latter index has become quite popular in the actuarial academia and practictioners to compare competing risk models.

ML techniques

BST

The Lightgbm model has been used to apply boosted trees on the provided data sets, minimizing Poisson deviance. As for most modern ML methods, a lightgbm model is fully defined by a set of many hyperparameter for which default values may not be optimal for the given data and there is no closed formula to identify the best combination for the given data.

Therefore an iperparameter optimization step was performed. For each hyperparameter a range of variation was set, then a 100-run trial was performed using a Bayesian Optimization approach performed by the hyperopt python library (Bergstra, Yamins, and Cox 2013). Under the BO approach, each subsequent iteration is perfomed toward the point that minimize the loss to be optimized, being the loss distribution by hyperparameter updated each iteration using a bayesian approach.

As suggested by boosting trees practitioners, the number of boosted models was not estimated under the BO approach but determined by early stopping. The loss was scored under the validation set and the number of trees chosen is that beyond which the loss stop to decrease and start diverging up.

The CMP and MKT model used the standard exposure (in logarithm base) as init score. The TRF model instead uses as init score the "a-priori" prediction of the MKT model on the CMP data.

Deep Learning

The chose DL architecture was set by several trials, based on previous experiments and symilar architecture found in the literature for tabular data analysis. Unfortunately, the hyperparameter space of a DL architecture is very vaste, comprising not only fitting level degrees of freedom (the optymizer, the number of epochs, the batch size) but the whole layers' architecture: the number of layers, the numbers of neuron within etc… At this regard it is common among practictioners to starts with a knowling working architecture in a symilar field and perform moderate changes. While more systemating DL architecture optimization approaches are being developed (e.g. the Neural Architecture Search) the use of such techniques was out of the scope of the paper

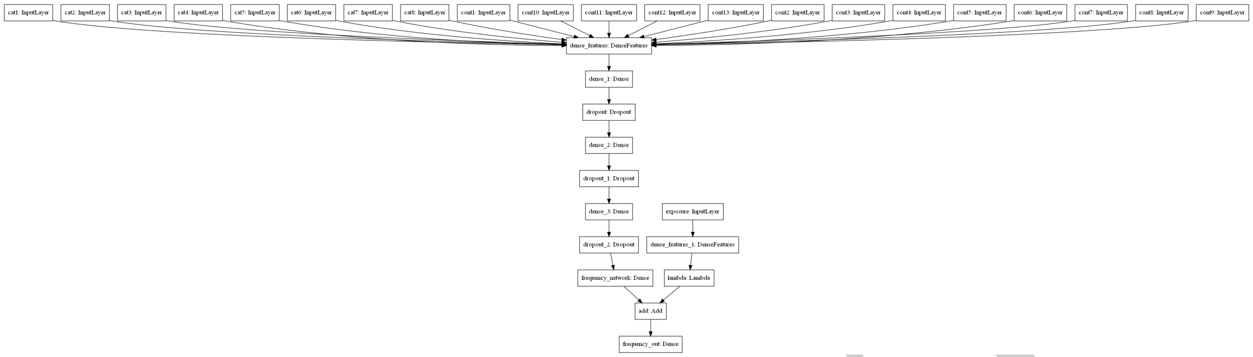

Figure 1: DL model structure

The same model shown above was used for both the CMP, MKT and TRF models. A dense layer collects the inputs, where the categorical variables have been handled using embedding. Three hidden layers perform the feature engineering and knowledge extraction from the input; Dropout layers have been added to robustify the process. As anticipated in the methodological section, the exposure part is separately handled in another branch and then merged in the final layer.

Overfitting was controlled using an EarlyStopping callback scoring the loss on the validation test and stop learning over next epochs if the loss did not improve for more than 20 epochs.

The TRF model has been build using the pre-trained weights calculated on the market data and continuing the traiing process on the CPN data.

Credibility

Comparison

The table below reports the predictive performance, evaluated on the company test set, for the DL and Bst models' families. The columns approach indicates whether the model was trained on market-only (mkt), company-only (cpy) or company data using a transfer learning approach (trf).

Table 1: Models comparison

model	approach	normalized_gini	actual_predicted_ratio
dl	mkt	0.921	0.924
dl	cpn	0.909	0.775
dl	trf	0.925	0.967
bst	mkt	0.939	0.975
bst	cpn	0.924	0.841
bst	trf	0.940	1.052

First, we see that the actual/predicted ratio is between 0.9 - 1.1 for all models, but company ones where it is somewhat worse. We anticipate that as the test set considers a year different from the train and validation pool, the predictions may be structurally biased as the insured risk depends by the year's climate and that frequency trending is not consider in the modeling framework at all. The transfer learning approach offers higher predictive accuracy when measured by the NG index and the predictive performance is the highest among all competitive approaches except for the boosting approach. Regarding the predictive accuracy, on the other hand, and expecially for DL models, we cannot rule out that the superiority of TL approaches holds for all possible MLP architectures. Also, it is likely to happen that as far as the company data increases, the advantage of transfer learning decreases.

TODO

Conclusion

We presented and application of TF that can be resembled to the traditional "credibility" approach to transfer the experience applied on a different, but similar, book of business to a newer one. We saw that ML approach may provides interesting results and may be worth to try with.

Finally, we performed our empirical analysis transferring loss experience from an external insurance bureau to a specific company portfoglio. This "transfer of experience" may be also performed within the same company for example when new products, taylored for niche books of business, are created. Initial losses estimates may be performed on the initial product and then applied as initial scores on the newer portfolio.

TODO

Appendix

Code

The modeling has been performed using both R (R Core Team 2021) and (Van Rossum and Drake 2009). The following files has been provided:

0. preprocess and anonymize.py: work on original data, anonymizing column and internal datasets and split data across train, validation and test set; the config_all.py file provides ancillary functions to perform this step;
1. analysis_neural_network.py: implement the MLP approach on the dataset;
2. analysis_lightgbm.py: apply the LightGbm on the dataset;
3. compare_predictions.R: contrasts the different predictive approaches on the company test data set,

Data Preparation and Anonymization

The market and company data file were loaded. An initial renaming of the variable has been performed, conventionally naming the continuous one as cont_x while the categorical one as cat_x, being x a number from one up to the number of variables of such category. The following criterion have been used to filter out anomalous observations: presence of missing values in any of the observations, zero exposures.

Then, the available data has been split threefold: the last available year has been set to the test set, while the remaining years have been split into a train / validation set using a 80/20 ratio. Therefore we have available three dataset for the marked data, and another three for the company one.

References

Antonio, Katrien, and Jan Beirlant. 2007. "Actuarial Statistics with Generalized Linear Mixed Models." *Insurance: Mathematics and Economics* 40 (1): 58–76.

"Bayesian credibility for GLMs." 2018. *Insurance: Mathematics and Economics.* https://doi.org/10.1016/j.insmatheco.2018.05.001.

Bergstra, James, Daniel Yamins, and David Cox. 2013. "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures." In *International Conference on Machine Learning*, 115–23. PMLR.

Bühlmann, Hans, and Alois Gisler. 2006. *A Course in Credibility Theory and Its Applications*. Springer Science & Business Media.

Chen, Tianqi, and Carlos Guestrin. 2016. "Xgboost: A Scalable Tree Boosting System." In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–94.

Danzon, Patricia Munch. 1983. "Rating Bureaus in U.S. Property Liability Insurance Markets: Anti or Pro-competitive?" *The Geneva Papers on Risk and Insurance - Issues and Practice* 8 (4): 371–402. https://doi.org/10.1057/gpp.1983.42.

Elith, J., J. R. Leathwick, and T. Hastie. 2008. "A working guide to boosted regression trees." https://doi.org/10.1111/j.1365-2656.2008.01390.x.

Ferrario, Andrea, Alexander Noll, and Mario V Wuthrich. 2020. "Insights from Inside Neural Networks." *Available at SSRN 3226852*.

Frees, Edward W. (Jed), Glenn Meyers, and A. David Cummings. 2014. "Insurance Ratemaking and a Gini Index." *The Journal of Risk and Insurance* 81 (2): 335–66. http://www.jstor.org/stable/24546807.

Goldburd, Mark, Anand Khare, and Dan Tevet. 2016. *Generalized Linear Models for Insurance Rating*. 5. https://doi.org/10.2307/1270349.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Adaptive Computation and Machine Learning. MIT Press.

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. "Lightgbm: A Highly Efficient Gradient Boosting Decision Tree." *Advances in Neural Information Processing Systems* 30: 3146–54.

Noll, Alexander, Robert Salzmann, and Mario V Wuthrich. 2020. "Case Study: French Motor Third-Party Liability Claims." *Available at SSRN 3164764*.

Prokhorenkova, Liudmila, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2017. "CatBoost: Unbiased Boosting with Categorical Features." *arXiv Preprint arXiv:1706.09516*.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Schelldorfer, Jürg, and Mario V Wuthrich. 2019. "Nesting Classical Actuarial Models into Neural Networks." *Available at SSRN 3320525*.

Spedicato, Giorgio Alfredo, Christophe Dutang, and Leonardo Petrini. 2018. "Machine Learning Methods to Perform Pricing Optimization. A Comparison with Standard GLMs." *Variance* 12 (1): 69–89.

Van Rossum, Guido, and Fred L Drake. 2009. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

Yan, Jun, James Guszcza, Matthew Flynn, and Cheng-Sheng Peter Wu. 2009. "Applications of the Offset in Property-Casualty Predictive Modeling." In *Casualty Actuarial Society E-Forum, Winter 2009*, 366.