Comparative Analysis of Recent Implementations of Gradient Boosting for Decision Trees

Dominik Chevalier and Marie-Pier Côté

École d'actuariat, Université Laval

From GLMs to gradient boosting

GLMs have been used for many years for pricing in general insurance.

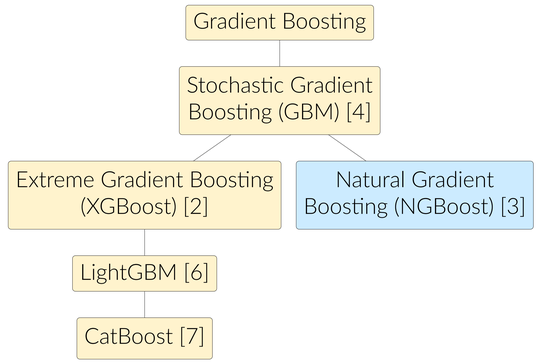
Motivations :

- [5] have illustrated that boosted regression trees could outperform GLMs;
- Implementations aim at improving gradient boosting for decision trees [4].

GLMs	Boosting
▪ are readily interpretable;	▪ is flexible (loss function and interaction depth);
▪ can be globally unbiased.	▪ enhances variable lift.

Objective : To compare the recent implementations of gradient boosting for decision trees on the basis of performance and of efficiency.

Recent implementations of gradient boosting



Notation

- $\mathcal{D}$: n-observation dataset of target variable y_i and features $\mathbf{x}_i$
- $\mathbf{x}_i$: k-feature vector for observation i
- $f_{model}(\mathbf{x})$: prediction function, which links $\mathbf{x}_i$ to y_i
- M: number of iterations
- f_0: initial prediction
- $I_{\mathcal{L}}(\mathbf{p})$: Fisher information matrix with distribution parameters $\mathbf{p}$
- $h(\mathbf{x})$: $f_{tree}(\mathbf{x}_i) = \sum_{j=1}^{J} \bar{y}_j \mathbf{1}(\mathbf{x}_i \in R_j)$
- R_j: region of a regression tree $j \in \{1, \dots, J\}$
- $\mathcal{L}\{y, f_{model}(\mathbf{x})\}$: loss function
- d: tree depth
- λ: learning rate
- δ: line sampling proportion
- $\mathbf{p}$: k-dimensional parameters vector

References

[1] Chavez-Demoulin, V., Embrechts, P., and Hofert, M. (2016). An extreme value approach for modeling operational risk losses depending on covariates. *Journal of Risk and Insurance*, 83(3):735–776.

[2] Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794.

[3] Duan, T., Anand, A., Ding, D. Y., Thai, K. K., Basu, S., Ng, A., and Schuler, A. (2020). NGBoost: Natural gradient boosting for probabilistic prediction. In *International conference on machine learning*, pages 2690–2700. PMLR.

[4] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 29:1189–1232.

[5] Henckaerts, R., Côté, M.-P., Antonio, K., and Verbelen, R. (2021). Boosting insights in insurance tariff plans with tree-based machine learning methods. *North American Actuarial Journal*, 25(2):255–285.

[6] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3147–3155.

[7] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31:6639–6649.

Gradient boosting algorithm for point prediction

We have hyperparameters $\mathcal{D}$, δ, M, λ and d.

1. **Initialization :** $f^0_{GBM}(\mathbf{x}_i) \leftarrow \operatorname{argmin}_b \left\{ \sum_{i=1}^{n} \mathcal{L}(y_i, b) \right\}$
2. For $m = 1, \dots, M$,
   - **Sampling :** $\mathcal{D}' \leftarrow$ Sample of size $\delta \times n$ observations without replacement from dataset $\mathcal{D}$
   - **Evaluation :** $g_{m,i} \leftarrow -\nabla_{f(\mathbf{x}_i)} \mathcal{L}(y_i, f(\mathbf{x}_i))$ evaluated at $f(\mathbf{x}_i) = f^{m-1}_{GBM}(\mathbf{x}_i), \forall\, i \in \mathcal{D}'$
   - **Learning :** $R_{j,m} \leftarrow j$th region of the CART tree (of depth d) fitted to pseudo-residuals $g_{m,i}$, for $j \in \{1, \dots, J_m\}$
   - **Optimization :** $\hat{b}_{j,m} \leftarrow \operatorname{argmin}_b \left[\sum_{i:\mathbf{x}_i \in R_{j,m}} \mathcal{L}\{y_i, f^{m-1}_{GBM}(\mathbf{x}_i) + b\} \right]$, for $j \in \{1, \dots, J_m\}$
   - **Update :** $f^m_{GBM}(\mathbf{x}_i) \leftarrow f^{m-1}_{GBM}(\mathbf{x}_i) + \lambda \sum_{j=1}^{J_m} \hat{b}_{j,m} \mathbf{1}(\mathbf{x}_i \in R_{j,m})$
3. **Prediction :** $f^M_{GBM}(\mathbf{x}_i)$

Each implementation adds new features to GBM

Analyzed implementations tackle two main drawbacks of gradient boosting :

1. All splits are considered in the trees ⇒ demanding and risk of overfitting,
2. Loss function is not approximated ⇒ demanding or impossible optimization.

- **XGBoost:** Taylor approximation, complexity control, split finding algorithm;
  + Flexible and quick, default tree direction for special cases.
  - One-hot encoding required, additional step for default direction selection.
- **LightGBM:** Gradient-based one-side sampling, exclusive feature bundling;
  + Training of underfitted regions, overfit prevention, dimensional shrinkage.
  - Less interpretable categorical treatment, sorting gradients is demanding.
- **CatBoost:** Target statistics for categorical features, ordered boosting;
  + Enhanced generalization, dimensional shrinkage for categorical treatment.
  - Less effective for larger datasets, stocking permutations is demanding.
- **NGBoost:** Probabilistic regression with natural gradient;
  + Distributional information, interpretation of each parameter.
  - Larger training time, demanding optimization for $\hat{\rho}$.

There is no one-size-fits-all implementation

Table 1. RMSE of predictions with squared error loss (log scale).

Dataset	GLM LN	GBM LN	XGBoost LN	LightGBM LN	CatBoost LN	NGBoost LN
Emcien	0,46	0,48	**0,46**	0,46	0,46	**0,53**
freMTPL	1,24	1,24	1,24	**1,26**	1,23	**1,22**
freMPL	**1,31**	1,30	1,26	1,28	1,26	**1,23**
Allstate	0,64	0,65	0,63	0,70	**0,62**	**0,72**
Disability	**1,14**	1,11	**1,09**	1,10	1,09	1,11

Table 2. Training time (seconds).

Dataset	XGBoost Gamma	LightGBM Gamma	GBM LN	XGBoost LN	LightGBM LN	CatBoost LN	NGBoost LN
Emcien	5,12	**4,64**	**21,65**	4,83	**4,41**	15,78	20,87
freMTPL	8,72	**1,72**	37,46	3,77	**3,69**	48,25	**57,74**
freMPL	6,87	**1,43**	13,62	**3,03**	3,49	**59,37**	41,75
Allstate	10,31	**6,15**	42,91	5,77	**5,47**	56,41	**102,60**
Disability	6,18	**4,64**	14,56	**2,60**	4,11	46,10	**251,40**

NGBoost algorithm for parameter estimation

We have hyperparameters $\mathcal{D}$, M, $\mathbf{p}$, λ and d. The natural gradient is

$$\tilde{\nabla}\mathcal{L}(y, \mathbf{p}) \propto I^{-1}_{\mathcal{L}}(\mathbf{p}) \nabla_{\mathbf{p}} \mathcal{L}(y, \mathbf{p}).$$

1. **Initialization** at MLE: $\mathbf{f}^0_{NGBoost}(\mathbf{x}_i) \leftarrow \operatorname{argmin}_{\mathbf{p}} \left\{ \sum_{i=1}^{n} \mathcal{L}(y_i, \mathbf{p}) \right\}$
2. For $m = 1, \dots, M$,
   - **Evaluation :** $\mathbf{g}_{m,i} \leftarrow \tilde{\nabla}\mathcal{L}(y, \mathbf{p})$ evaluated at $\mathbf{p} = \mathbf{f}^{m-1}_{NGBoost}(\mathbf{x}_i), \forall\, i \in \mathcal{D}$
   - **Learning :** $\mathbf{h}^m(\mathbf{x}_i) \leftarrow k$ trees (of depth d) fitted to pseudo-residuals $\mathbf{g}_{m,i}$
   - **Optimization :** $\hat{\rho}^m \leftarrow \operatorname{argmin}_{\rho} \left[\sum_{i=1}^{n} \mathcal{L}\{y_i, \mathbf{f}^{m-1}_{NGBoost}(\mathbf{x}_i) + \rho \mathbf{h}^m(\mathbf{x}_i)\} \right]$
   - **Update :** $\mathbf{f}^m_{NGBoost}(\mathbf{x}_i) \leftarrow \mathbf{f}^{m-1}_{NGBoost}(\mathbf{x}_i) + \lambda \hat{\rho}^m \mathbf{h}^m(\mathbf{x}_i)$
3. **Prediction :** $\mathbf{f}^M_{NGBoost}(\mathbf{x}_i)$

Are predictions appropriate for the assumed distribution?

Predictions from point regression overfit the expected value with squared error loss. Predictions from probabilistic regression are appropriate for the distribution. This duality is absent when the loss function is the gamma deviance. The method used to transform the quantiles is inspired by [1].

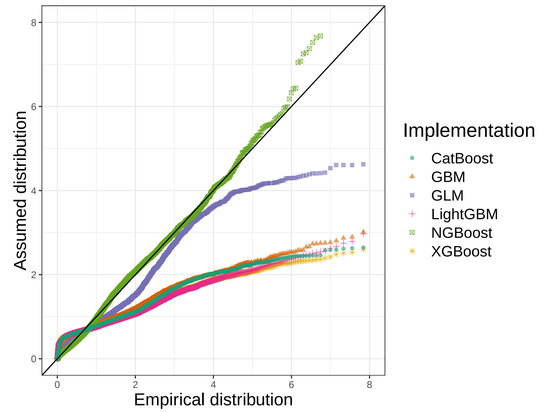


Figure 1. Transformed quantiles with predictions under lognormal distribution.

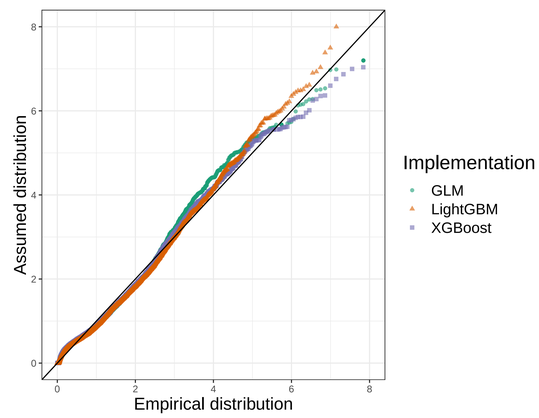


Figure 2. Transformed quantiles with predictions under gamma distribution.

dominik.chevalier.1@ulaval.ca