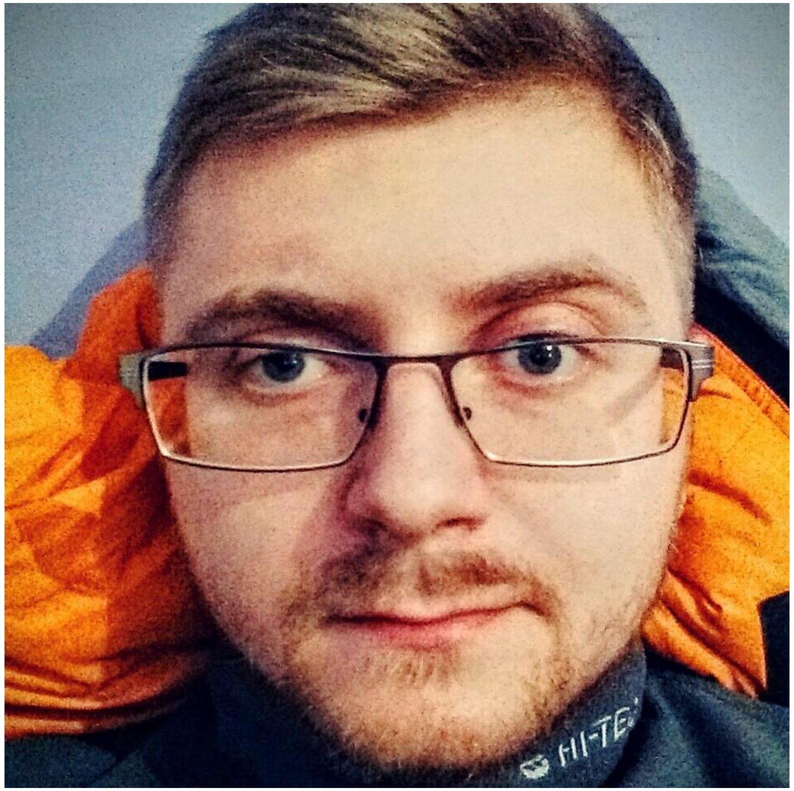
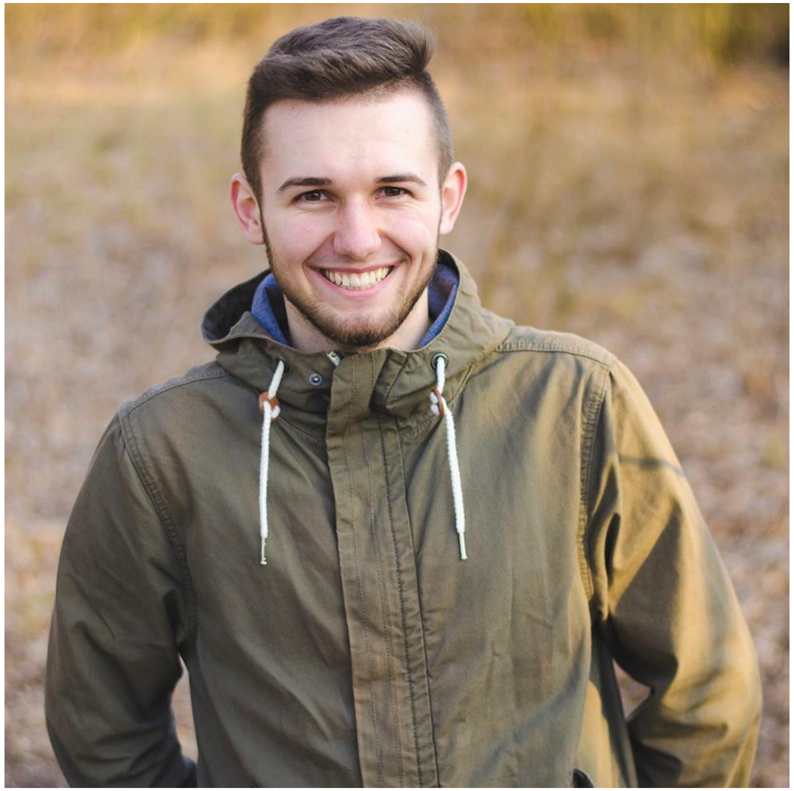
Custom loss functions for binary classification problems with highly imbalanced dataset using Extreme Gradient Boosted Trees

Bartosz Kolasa, Patryk Wielopolski

28 September 2019

About us





Data Science Team at DataWalk

Agenda

1. Motivation and problem statement
2. Theoretical aspect
3. Experiment
4. Implementation challenges
5. Results and conclusions

GitHub repository

Source codes and data repository:

https://github.com/pfilo8/WhyR-Presentation

Motivation and problem statement

Motivation

- Highly imbalanced datasets are very common case in insurance industry
- Preserving high precision of model predictions with respect to recall is very important case when dealing with fraud detection problems

Problem statement

Status quo:

- Binary classification algorithms often underperform in predicting positive values on highly imbalanced datasets

Goal:

- Improvement of "postive class friendly" performance measure for highly imbalanced dataset in binary classification problem

Theoretical aspect

XGBoost recall

Objective of XGBoost model at step t

$$\sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

- g_i - first derivative of loss function
- h_i - second derivative of loss function
- f_t - decision tree at step t
- Ω - regularization term

Custom Loss Functions

- Cross Entropy

$$L_{CE} = -ylog(\hat{y}) + (1 - y)log(1 - \hat{y})$$

- Weighted Cross Entropy

$$L_{WCE} = -Dylog(\hat{y}) + (1 - y)log(1 - \hat{y})$$

- Focal Loss

$$L_{FL} = -y(1 - \hat{y})^{\gamma}log(\hat{y}) - (1 - y)\hat{y}^{\gamma}log(1 - \hat{y})$$

- Bilinear Loss

$$L_{CE+B} = (1 - \alpha)[-ylog(\hat{y}) + (1 - y)log(1 - \hat{y})] + \alpha[yD + \hat{y} - y\hat{y}(1 + D)]$$

- Log Bilinear Loss (after transformations equal to Weigted Cross Entropy)

Experiment

Experiment description

Dataset:

- Fraud detection use case
- Real-world dataset from Insurance Industry
- 118 unnamed features generated by PCA
- Positive class fraction: 0.7%

Metric:

- AUCPR

Experiment:

- Best AUCPR for all proposed custom loss functions
- 5 fold stratified Cross Validation
- Hyperparameter tuning using MBO

Implementation challenges

Implementation

For implementation we used widely known R packages:

- xgboost
- dplyr
- mlr

Contribution

Our contributions:

- Providing essential derivatives for bilinear loss
- Implementation of custom loss functions
- Implementation of mlr wrapper for XGBoost with custom loss functions
- Implementation of mlr wrapper for AUCPR measure

Results and conclusions

Results

Performance.Measure	Cross.Entropy	Focal.Loss	Weighted.CE	Bilinear
AUCPR	0.0742625	0.0727012	0.0724779	0.0641854

Alternative results

Wang C., Deng C., Wang S. (August 2019) "Imbalance-XGBoost: Leveraging Weighted and Focal Losses for Binary Label-Imbalanced Classification with XGBoost"

- XGBoost with Weighted Cross Entropy and Focal Loss
- Improvement of F1 Score at Parkinson's Disease Dataset (3:1 ratio) compared to other classification algorithms
- No standard XGBoost results presented

Conclusions

- According to our experiment changing loss function doesn't increase AUCPR measure
- This method is not applicable for highly imbalanced datasets (100:1 ratio) or AUCPR measure is insensitive to changes of objective

Thanks for your attention!

Questions?

References

Bischl, Bernd, Jakob Richter, Jakob Bossek, Daniel Horn, Janek Thomas, and Michel Lang. 2017. “mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions.”

Chen, Tianqi, and Carlos Guestrin. 2016. “XGBoost: A Scalable Tree Boosting System.”

Davis, Jesse, and Mark Goadrich. 2006. “The Relationship Between Precision-Recall and Roc Curves.” In.

Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. “Focal Loss for Dense Object Detection.”

Resheff, Yehezkel S., Amit Mandelbom, and Daphna Weinshall. 2017. “Controlling Imbalanced Error in Deep Learning with the Log Bilinear Loss.” In.

Wang, Chen, Chengyuan Deng, and Suzhen Wang. 2019. “Imbalance-XGBoost: Leveraging Weighted and Focal Losses for Binary Label-Imbalanced Classification with XGBoost.”