Main Topic

Yusheng, Ni
Yang, Xia

Shanghai University of Finance and economic

December 3, 2018

Overview

Problem overview

- In credit scoring, application selection methods have naturally-occurring selection bias with model based on accepted population with known performance. The credit prediction for rejected populations generates the problem of "Missing not at random".
- In the data, credit scoring with sample unbalanced amounting to rate of 15:1 and high dimensional features. Dimension of features amount to 6500, interfering with robustness of prediction.

Difficulties

- The two classes are imbalanced and the prior of the positive classes is small itself.
- The training set may have large sample bias.
- The dimension of feature space is ultra high and the data is quite dirty.

Approach

1. Establish baseline
2. Variable selection(Dimension reduction)
3. Fine tune model
4. Model ensemble
5. Apply reject inference related methods

Techniques on Variable selection

SIS+Adaptive Lasso

Sure independent screening(SIS) is a method to reduce dimensionality from ultra high to a moderate scale which guarantees the important variables survive after applying a variable screening procedure with probability tending to 1.

This method is always be used with Danzig Selector, Adaptive Lasso and SCAD.

Note that Lasso variable selection is inconsistent. Adaptive Lasso,as a refined version,fixes this problem and be proved to bear oracle property.

Techniques on Variable selection

Boosting variable selection

Although the SIS+Adapative Lasso procedure has theoretically good properties, they are restricted only to generalized linear models. Our experiments show that in practice this procedure is not suitable for tree based algorithms.

On the other hand, Boosting algorithms can calculate the feature importance during training. It has widely been used in competitions like Kaggle. And also this is a good way to add and then select useful manually made features.

Notes on paramter tuning

1. first decide the approximate model complexity for this task
2. smaller learning rate will lead to better accuracy
3. inspect the performance both on training and validation set pay close attention to overfitting (the gap between performance on training set and validation set is to large).
4. $\ell 1$-penalty leads to better generalization than $\ell 2$ under same model complexity but the trade-off is that it may require bigger model complexity.
5. Over-sampling or scaled loss for positive class doesn't always work well in the imbalanced case especially when the prior of the positive class is small itself.

Reject inference

Methods

1. Fuzzy Augumentation
2. Parcelling
3. Re-weight
4. Re-classification

The two core ideas are :

Use heuristic ways to give 'unlabelled data' labels.

Use the prior knowledge that the labelled data is selected by a classifier and try to replicate the classifier and use its information.

Reference: http://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2731-2018.pdf

Model Ensemble

Weighted Average

Train 4 models with different generalization abilities by 4-fold cross validation and average prediction results

Bagging

Conduct slight modification with model coefficients, train 30 models and take average of the final prediction results

Stacking

Train 3 models with different generalization abilities by 4-fold cross validation and compile the base models with Logistic Regression to get final prediction

Experiment Results

	Accuracy	Precision	Recall	F1 score	AUC score
LR	0.7248	0.11	0.5	0.1922	0.688
LR(2179)	0.767	0.170	0.61	0.2666	0.777
Xgboost	0.932	0.57	0.080	0.141	0.814
Ensemble	0.940	0.89	0.158	0.269	0.843

Table: Results

Conclusion

- Xgboost is a powerful tool!
- Although xgboost perform well its Recall score is small which is however important in real cases with sensitive costs.
- Things like Xgboost and DNN can't avoid overfitting sometimes. (Too large penalty will cause too much bias in your model)
- Logistic Regression also has potential to improve.
- Traditional Reject Inference methods don't work well as we expect.

Discussion

Although final result is not bad, there are several possible ways to get more improvements.

- The delicate use of semi-supervised algorithms
- Smarter feature engineering such as feature crosses
- Is it possible for low dimension features to get the same result?(and hence more interpretability?)
- How much can we reduce the effect of sample imbalance and make the tradeoff between 'FN' error and 'FP' error in the cost sensitive cases.

Thank you!