A Volume Based Approach to Improve Default Prediction Model

Cong Zhang[1,a,*]

[1]*Alibaba, Shenzhen, China*
a. zcistc123@gmail.com
**corresponding author*

Keywords: default risk, machine learning model, GBDT, CVA

Abstract: In recent years, small loan businesses are growing rapidly in emerging markets. And due to lack of traditional customer risk related information and development of big data related techniques, more and more financial loan companies apply machine leaning models to manage loan risks. However, traditional machine learning algorithms do not financially optimize or evaluate models and thus the optimal models we get may not be the best ones in term of financial perspective. In this paper, the author adds CVA components into model objective functions on three popular GBDT-based machine learning models: Xgboost, LightGBM, Catboost; and applies those models on Kaggle's European credit card fraud detection dataset and Lending Club data loan dataset to verify whether this technique can lead to a financially preferred result. As a result, it is found that using CVA components to adjust model objective functions during training process will enable models to predict more accurately for the loans that are more likely to make large gains/losses, and thus give us financially optimized models.

1. Introduction

Small installment loan and cash loan businesses have been growing rapidly in emerging markets in recent years. Comparing with traditional credit card business in more developed markets, small loan businesses in emerging markets are facing a lot of new challenges. For example, verifying customers' IDs could be a difficult problem here as those countries don't have developed identity registration and investigation systems, so synthetic-identity fraud has quickly become a common problem in these markets. Also, some types of customers' information that are widely used in traditional credit card business, such as educational level, occupation, personal income, and bank account balance, are almost impossible to access and be verified in those countries. As a result, the traditional risk control framework does not work well in these markets and the loss from default is much higher compared with those in developed markets. In the past several years, loan companies usually use high interest rate to compensate the loss from default cases. But in recent years, governments in those countries begin to regulate usurious loan; and thus, how to control loan default risks within an acceptable level become very critical for loan companies.

As a consequence, companies try to utilize multiple data sources together, such as address books and call logs on cellphones, network at social APPs, action logs on software and webpages to get useful information. And with the explosive development of big data-based machine learning techniques, a variety of machine learning based models have also been employed by companies to predict loan default risks. For example, Decision Tree model is a popular machine learning model to solve classification problem since 1980s. It was originally based on information theory (e.g. ID3 and ID4.5 or CART), which means we would use information entropy or Gini Index to find best point to split data and make multiple binary trees (in binary classification cases) together into a decision tree model. Then early in this century, Friedman proposed an updated model called GBDT model which used techniques called gradient boosting. It solves classification or regression problem by optimizing loss function based on gradient descent (or inversely, gradient ascent) and making multiple rounds of decision building, both of which put more focuses on the wrong cases from last round(the boosting technique). This model has been widely used, due to its efficiency, accuracy, and interpretability. Based on GBDT framework, several developed algorithms have then been put forward recently and widely used in industry. For instance, XGBOOST uses regularized learning objective that enables clever penalization trees to prevent overfitting, and also uses Newton boosting to accelerate gradient descent; LIGHTGBM uses techniques like gradient based one-side sampling (GOSS) during boosting phrase as well as exclusive feature bundling to process high-dimensional sparse data, in order to improve boosting efficiency while keeping accuracy for learning decision trees; and CATBOOST focuses on using specific techniques to process category feature to better help decision-making.

However, there are also some problems in those models. Traditional loan default risk measurements, like CVA, measure the risk according to its probability, loan amount and all factors that could affect actual loss in a default event. However, machine learning models are usually evaluated by confusion matrix-based evaluation metrics like KS and AUC , and focus on predicting default probability of a single default event, regardless of the loss amount from a default event, profits from a non-default event and the tradeoff between them. From financial perspective, thus, the traditional loan default risk measurements could under- or overestimate the real loss for the default risk of the whole loan portfolio.

In this paper, the author will use Kaggle's European credit card fraud detection dataset to illustrate why traditional model evaluation metrics may not lead to a financially optimized result, and by adjusting model loss function, the model prediction result can be improved in the preferred direction.

The following content will be presented in three parts. First, the paper will expound how to modify the algorithm by adding CVA and profits components into loss function, enabling the model to go into the direction we want. Second, the author will do data cleansing, process original data into features and use three models: XGBOOST, LIGHTGBM and CATBOOST to make prediction and get results by using both the original and new loss functions. At last, the paper will compare the results of the new and the old models, verifying that a modified loss function will generate a financially optimized result.

2. Definition of the Objective Function

If we treat loan default problem as a traditional binary classification problem which only focuses on estimating the probability of the event rather than the influence of the whole event, we are not likely to get a desired model that can minimize the loss from the default event, as we do not take the different money amount of a default loss or gain from non-fault loans into consideration. Also, if

we evaluate our model by using confusion matrix-based metrics, we may not get a fair result as those metrics treat all samples equal weight and ignore the different amount of gain and loss from loans.

Thus, here the author will modify the loss function of the models and enable them to consider the actual value of loans, and then evaluate models from both confusion matrix-based metrics and volume-based metrics to see how modified loss function can improve models from the financial perspective.

2.1. For Single Loan

Suppose our loan is a zero coupon loan, and the principal, interest and loss given default (LGD) are fixed, then for a default event, under CAV framework, the loss will be:

$$Loss=LGD*EAD \quad (1)$$

At no default condition, the gain from loan will be:

$$Gain=principal*interest\ yield \quad (2)$$

Suppose y is a Boolean value that represents the loan event at maturity and is independent from loan principal and interests yield. And 1 present default while 0 present non-default, then the value of the loan can be presents as:

$$\begin{aligned} Value\ of\ loan &= (1-y)*principal*interest\ yield - y*LGD*EAD \\ &= principal*interest\ yield - y*(principal*interest\ yield + LGD*EAD) \end{aligned} \quad (3)$$

As small loan usually has no collateral and will lose all of the principal during default, here we suppose EAD equals to principal and the above equation will be:

$$\begin{aligned} Value\ of\ loan &= principal*interest\ yield - y*(principal*interest\ yield + LGD*principal) \\ &= loanprofits - (princiapl*(interest\ yield + LGD))*y \end{aligned} \quad (4)$$

During the model prediction process, suppose principal, interest yield and LGD are all fixed, only default event y will be replaced by estimated default probability, $\hat{y}$.

$$Estimated\ value\ of\ loan = loan\ profits - (princiapl*(interest\ yield + LGD))*\hat{y} \quad (5)$$

And to estimate default probability, we use information entropy log loss as our loss function, which would be:

$$L(y) = y*\ln\hat{y} + (1-y)*\ln(1-\hat{y}) \quad (6)$$

And since loan profits part are fixed, for the whole loan value estimation, the loss function will be:

$$L(value) = (princiapl*(interest\ yield + LGD))*L(y) \quad (7)$$

2.2. For Loan Portfolio

The total value of loan is the sum up of values from all single loans:

$$\begin{aligned}\text{Total value} &= \sum_{i=1}^{N} (\text{loan profits}_i - y_i * (\text{principl}_i * (\text{interest yield}_i + \text{LGD}_i))) \\ &= \sum_{i=1}^{N} \text{loan profits}_i - \sum_{i=1}^{N} y_i * (\text{principl}_i * (\text{interest yield}_i + \text{LGD}_i)) \\ &= \text{total profits} - \sum_{i=1}^{N} y_i * (\text{principl}_i * (\text{interest yield}_i + \text{LGD}_i)) \end{aligned} \quad (8)$$

Since total profits are fixed and independent from y, we will remove it and the loss function to estimate the whole loan portfolio will be:

$$L(\text{value}) = \sum_{i=1}^{N} L(y_i) * (\text{principl}_i * (\text{interest yield}_i + \text{LGD}_i)) \quad (9)$$

By dividing the loss function (9) by (10), the problem can be simplified into a binary classification problem with different sample weight:

$$\sum_{i=1}^{N} (\text{principl}_i * (\text{interest yield}_i + \text{LGD}_i)) \quad (10)$$

And then, our model objective function will become:

$$\text{New loss function} = \sum_{i=1}^{N} L(y_i) * w_i \quad (11)$$

Where

$$L(y_i) = y_i * \ln \hat{y}_i + (1 - y_i) * \ln(1 - \hat{y}_i) \quad (12)$$

And

$$w_i = \frac{(\text{principl}_i * (\text{interest yield}_i + \text{LGD}_i))}{\sum_{i=1}^{N} (\text{principl}_i * (\text{interest yield}_i + \text{LGD}_i))} \quad (13)$$

3. Model Methodology

Kaggle's credit card fraud detection dataset contains transactions made by credit cards, which occurred in two days in September 2013 by European cardholders. In the dataset, there are 285297 samples in total with 284807 normal transactions and 492 fraud transactions.

The author will randomly split the data into train and test sets with ratio 7:3. After split, the data distribution of the sample is:

Table 1: Sample set description.

Sample set description	Train Set	Test Set	Total
number of normal transactions	199,032	85,283	284,315
Num of fraud transaction	332	160	492
Money amount of normal transaction	17,604,730	7,497,732	25,102,462
Money amount of fraud transaction	42,121	18,007	60,128

The following steps are used to process original data into features for model training:

A. Conduct feature processing, converting time into hour of date.

B. Randomly select 70% of the data as test set.

C. Use GBDT feature importance method on test set, and select the features that with importance larger than zero.

D. As the data are imbalanced, we use SMOTE method to oversampling fraud samples to make a balanced training dataset.

E. Train test data with three models: XGBoost, LightGBM and Catboost.

F. Evaluate three models by using data from test set and check how they perform under confusion matrix-based evaluation metrics and volume-based metrics.

G. Add CAV weights into models training process to make new models and evaluate models again. Here, the interest yield is set in Eq.12 as 0.05 as assumption.

4. Model Result

In this part, the author will show how models perform differently without/with CVA weights into training process. The results will be illustrated from two aspects: traditional model evaluation metrics. (based on confusion matrix) and money loss-based metrics.

Without CVA weights added, the KS and AUC of three models are:

Table 2: Evaluation Metrics without weights added into model training.

	Xgboost	LightGBM	Catboost
AUC	0.95	0.94	0.89
KS	0.9	0.9	0.9

We can see all three models get high AUC and KS, and the XGboost model get the highest AUC (0.95) while KS of three models are the same.

And if we take 0.5 as threshold, confusion matrices for three models are:

Table 3: Confusion Matrix without weights added into model training.

	Xgboost	LightGBM	Catboost
True Positive	145	143	133
True Negative	84218	84272	80242
False Positive	1,065	1,011	5,041
False Negative	15	17	27

We can see that the Xgboost model has the least false samples while Catboost has the most. And the percentage of false positive in total negative samples is much less than false negatives in total positives, indicating that all three model predicts no-fraud samples better than fraud sample, this is probably due to the imbalanced distribution of positive and negative samples.

Table 4 shows the money amount evaluation metrics for false samples.

Table 4: Money Amount Evaluation without weights added into model training.

	Xgboost	LightGBM	Catboost
Money Amount of False Negative	3,430	4,391	4,709
Money Percentage of False Negative	19.05%	24.39%	26.15%
Money Amount of Positive Negative	202,085	242,387	436,248
Money Percentage of Positive Negative	2.70%	3.23%	5.82%
Money Loss From False Samples	13706	16730	26757

We can see that the results in this table for three models are consistent with previous results: Xgboost has the best result and Catboost does not perform well. Under our 0.05 interest yield assumption, the money loss caused by negative samples of Xgboost model is only about half of money loss from Catboost model.

The following tables shows the model result with CVA weights added.

Table 5: Evaluation Metrics with weights added into model training.

	Xgboost	LightGBM	Catboost
AUC	0.93	0.91	0.89
KS	0.86	0.87	0.87

We can see that the AUC and KS for three models are all decreased a little bit, while the Xgboost model still has the highest AUC but KS become the lowest. The decrease of AUC and KS is because after we add sample weights into model training process, model will be more focused on samples with more money amount that account for a small part of the total samples. And then the new models' results will lose its accuracy when predicting samples with less money amount but account for a large part of the total samples. Thus, in total, the new models' AUC and KS will decrease.

Table 6: Confusion Matrix with weights added into model training.

	Xgboost	LightGBM	Catboost
True Positive	138	134	134
True Negative	84209	84136	79566
False Positive	1,074	1,147	5,717
False Negative	22	26	26

The confusion matrix shows a similar change, the false samples, both positive and negative, have increases for three models.

Table 7: Money Amount Evaluation with weights added into model training.

	Xgboost	LightGBM	Catboost
Money Amount of False Negative	5,211	4,410	4,449
Money Percentage of False Negative	28.94%	24.49%	24.71%
Money Amount of False Positive	73,692	85,673	263,535
Money Percentage of False Positive	0.98%	1.14%	3.51%
Money Loss From False Sample*s*	9156	8914	17848

The money amount evaluation shows an interesting result here.

We can see the money amount of the false negative increases in the new models, but the proportion is not as much as the increase in sample numbers for LightGBM and Catboost. For false positive samples, the total money amount even decreases while the sample numbers of false positives increases in all three models. And here we can see the LightGBM model performs the best as it has the least money loss from false samples.

The change of average money amount of false samples and money loss are listed in following table.

Table 8: Change of some metric for new models.

	Xgboost	LightGBM	Catboost
Change in Average Money Amount of False Negative	8	-89	-3
Change in Average Money Amount of False Positive	-121.14	-165.06	-40.44
Change in Money Loss From False Samples	-4,550	-7,816	-8,909

We can see the average money amount of both false positive and false negative samples are decreased, which results in a decreased total money loss from false samples. This change is because that after we add CVA weights, we will focus on samples with higher CVA weights. Thus our models will be more accurate in predicting samples that could cause more gains or losses, thus resulting a more financially preferred results.

5. Robust Check

To check if the conclusion is robust, the author further applies the above method to another dataset, Kaggle's Lending Club Loan Dataset that contains the complete loan data for all loans issued through the 2007-2015, including the current loan status and payment information.

The data is also randomly split into train and test sets with ratio 7:3. After split, the data distribution of the sample is:

Table 9: Sample set description.

	Train Set	Test Set	Total
number of normal transaction	1,513,655	648,710	1,880,108
Num of fraud transaction	197644	84613	282257
Money amount of normal transaction($)	19,858,273,400	8,502,040,125	28,360,313,525
Money amount of fraud transaction($)	3,134,323,900	1,335,623,375	4,469,947,275

Here the same methodology has been applied as what has done on Fraud Detection dataset.

A. Conduct feature processing, and remove useless data(duplicated records, columns whose records are all null, data acquired after loan issue) from dataset.

B. Randomly select 70% of the data as test set

C. Use GBDT feature importance method on test set, and select the features that with importance larger than zero.

D. As the data are imbalanced, we use SMOTE method to those oversampling fraud samples to make a balanced training dataset.

E. Train test data with LightGBM.

F. Evaluate the model by using data from test set and check how they perform under confusion matrix-based evaluation metrics and volume-based metrics.

G. Add CAV weights into model training process to make new model, and evaluate the model again.

Here, the interest yield is set in Eq.12 as 0.05 as assumption.

And the following tables are results:

Table 10: Comparison with without CVA weigths.

	LightGBM	LightGBM with CVA weigths
AUC	0.7451	0.7439
KS	0.3447	0.3436

And if we take 0.5 as threshold, confusion matrices for the three models are:

Table 11: Confusion Matrix based metrics.

	LightGBM	LightGBM with CVA weigths
True Positive	10,300	10,385
True Negative	563,923	563,839

False Positive	174	258
False Negative	74,313	74,228

As what happened on pervious dataset, the KS and AUC of new models slightly decrease, and we can also find similar results from confusion matrix that negative samples of new model slightly increase.

And if we use money amount evaluation metrics:

Table 12: Money Amount Evaluation.

	LightGBM	LightGBM with CVA weigths
Money Amount of False Negative($)	1,74,575,200	1,173,098,775
Money Percentage of False Negative	87.94%	87.83%
Money Amount of False Positive($)	2,974,400	4,454,875
Money Percentage of False Positive	0.035%	0.052%
Money Loss From False Samples($)	1,233,452,680	1,231,976,458

We can see although the average money amount of false positive samples has increased, the total money loss from false samples has decreased because of the decrease in average money amount of false in negative samples. The result is similar to what happened on on Fraud Detection dataset, so the robustness of model has also been checked on Lending Club dataset.

6. Conclusion

In this paper, the author shows that for a loan default/fraud risk prediction problem, employing CVA components (Equation 13) to adjust model objective function will generate a financially optimized model.

The author uses three popular GBDT based models: Xgboost, LightGBM, and Catboost, and applies both the original and modified versions of each model on Kaggle's European credit card fraud detection dataset. Then by comparing the results from both versions, it is found that the modified models with CVA weights can reduce the money loss from prediction error, which implies that the modified models with CVA weights are more financially preferred than the original models.

For robustness check, the author also uses the two versions of LightGBM model on Kaggle's Lending Club Loan Dataset and receives the similar results.

The limitation of this work is that both the two datasets used in the paper are imbalanced, so the performance of the method on the balanced datasets needs to be explored in the future. Also, the author assumed a fixed interest yield and the independence of default event from loan principal and interests, so the effectiveness of the model in a flexible interests condition and correlated default event with loan principal and interests condition also need to be further checked.

References

[1] Quinlan J R. Induction of decision trees[J]. Machine learning, 1986, 1(1): 81-106.

[2] Breiman L, Friedman J, Stone C J, et al. Classification and regression trees[M]. CRC press, 1984

[3] Friedman J H. Greedy function approximation: a gradient boosting machine[J]. Annals of statistics, 2001: 1189-1232.
[4] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.
[5] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[C]//Advances in neural information processing systems. 2017: 3146-3154.
[6] Prokhorenkova L, Gusev G, Vorobev A, et al. CatBoost: unbiased boosting with categorical features[C]//Advances in neural information processing systems. 2018: 6638-6648.
[7] Akkizidis I., Kalyvas L. Credit Valuation Adjustments. In: Final Basel III Modelling. [8] Palgrave Macmillan, Cham,2018 Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002, 16: 321-357.
[8] https://www.kaggle.com/mlg-ulb/creditcardfraud
[9] Bradley A P. The use of the area under the ROC curve in the evaluation of machine learning algorithms[J]. Pattern recognition, 1997, 30(7): 1145-1159.
[10] Shannon C E. A mathematical theory of communication[J]. Bell system technical journal, 1948, 27(3): 379-423.
[11] https://www.kaggle.com/wordsforthewise/lending-club