CREDIT RISK MODELING: COMBINING CLASSIFICATION AND REGRESSION ALGORITHMS TO PREDICT EXPECTED LOSS

Tim Kreienkamp,

Barcelona Graduate School of Economics (GSE),
Graduate Student at Maastricht University, School of Business and Economic

Andrey Kateshov,

PhD Candidate at Quantitative Economics Department,
School of Business and Economics, Maastricht University

Abstract

Credit risk assessment is of paramount importance in the financial industry. Machine learning techniques have been used successfully over the last two decades to predict the probability of loan default (PD). This way, credit decisions can be automated and risk can be reduced significantly. In the more recent parts, intensified regulatory requirements led to the need to include another parameter – loss given default (LGD), the share of the loan which cannot be recovered in case of loan default – in risk models. We aim to build a unified credit risk model by estimating both parameters jointly to estimate expected loss. A large, high-dimensional, real world dataset is used to benchmark several combinations of classification, regression and feature selection algorithms. The results indicate that non-linear techniques work especially well to model expected loss.

JEL: C52, G32

Keywords: Basel II; Credit risk; LGD; kaggle; gradient boosting; feature selection

Introduction

Credit scoring, the numerical assessment of credit default risk, was first developed in the 1940's and has constantly evolved ever since. Credit scoring is an important first step towards reducing credit risk and is proven to be highly effective. While classical techniques included scorecards and statistical techniques like logistic regression, with the advent of data mining, credit risk analysts started to utilize modern machine learning algorithms for credit scoring (Yap et al., 2011). The main aim of credit scoring is to estimate expected loss (*EL*), which is defined as

$$EL=PD*LGD*EAD, \qquad (1)$$

where the parameters are probability of default (*PD*), loss given default (*LGD*) and exposure at default (*EAD*) (Bluhm et al., 2003). *PD* indicates how likely a borrower is to not be able to (fully) pay back their loan. *LGD* is the share of the loan which the issuer will not be able to recover. *EAD* is the amount of money at risk. It is also common to express *EL* as a percentage figure of *EAD* (Basel Commitee on Banking Supervision, 2005):

$$EL=PD*LGD. \qquad (2)$$

Past research has focused mainly on estimating the first parameter, *PD*(e.g. Brown & Mues, 2012; Heiat, 2012; Mooney, 2011; West, 2000; Yap et al., 2011). This parameter is arguably of high importance. Unfortunately, the focus on *PD* has led to comparably little interest in the second parameter, *LGD*, which enables the issuer to develop a more nuanced picture of credit risk. Recently introduced regulatory requirements have fueled interest in predicting the second parameter of importance, *LGD*. The regulations summarized under the term "Basel II", introduced in 2006, impose minimum capital requirements on financial institutions to stabilize the financial system. The calculation of these requirements and the resulting risk models are primarily built around the three parameters mentioned above. *LGD* enters the capital requirements in a linear way (Basel Commitee on Banking Supervision, 2005), which makes its accurate calculation strategically important. The earlier research in this domain has illustrated that this is not an easy task. Developed models often suffer from low predictive accuracy. The primary techniques used here were linear regression, fractional-response regression and regression trees(Bastos, 2010; Caselli, Gatti, & Querci, 2008). One benchmarking study on *LGD* was conducted by Loterman et al. (Loterman et al., 2012). Here, a large variety of regression algorithms were benchmarked on 5 real-world credit datasets. Loterman et al. provide evidence that non-linear techniques, like Support Vector Machines (*SVM*) and Artificial Neural Nets (*ANN*), outperform "traditional" linear techniques, which suggests that there exist non-linear relationships between the features and *LGD* parameter (Loterman et al., 2012). This is supported by Tobback et al. (2014) who found that non-linear support vector regression gives the best results when forecasting *LGD*.

In estimating *PD* the machine-learning task is expressed as a binary classification problem. Since the last decade researchers have developed a broad variety of approaches to classification problems. For validating the performance of the classification mainly one benchmark data set – the "German Credit" data set[1] – has been used. The most researched machine learning techniques for credit scoring are *ANN, SVM* and ensemble methods. It is known that the "German Credit" data, an SVM-based model, performs very well and better that ANN approach (Heiat, 2012). However, this improvement over neural nets is only marginal. For estimating *LGD*, the task is different. As it is a continuous percentage share, regression models have to be developed to model it. Here, the natural skewness of the dataset further complicates the task.

One of the distinctive features of this paper compared to the previous research is the use of a new dataset. This article largely follows the methodology of Loterman et al. (Loterman et al., 2012) and aims to compare several *LGD* estimation techniques. The dataset we explore is inherently different from the previous studies, exhibiting much more features. Unlike the previous works we introduce various feature selection methods to reduce the dimensionality of the data. The dataset used in this article comes from a machine learning competition sponsored by the Imperial College London through the "kaggle" platform[2]. We will therefore refer to this data as the "kaggle" dataset. The outcomes of this article are largely based on the experiences of the first author, who participated in the challenge ranking within the top 10% of the contenders.

Credit Scoring Datasets

The "kaggle" dataset presents challenges in the following three dimensions:

- number of features;
- balance of the data;
- outcomes estimated.

As we have mentioned above, most researchers have focused on modeling the PD parameter, utilizing one particular data set for benchmarking: the "German Credit" data set. Some researchers also used the "Australian Credit" data set[3]. In Table 1 we provide the comparison of the data sets with respect to their size.

Table 1

Datasets size

	German Credit	Australian Credit	Kaggle (Our Dataset)
Features (Columns)	29	14	759
Observations (Rows)	1000	690	105471
Missing Data	No	Yes	Yes
Size (MB)	~ 0.25	~ 0.03	~1500

More recent publications that also consider LGD parameter used about 5-44 features and up to 79479 observations (Loterman et al., 2012)Loterman et al., 2012; Tobback et al., 2014). This article pushes this frontier even further with 759 features and more that 100 000 observations.

The growth of available data requires new approaches. Methods developed in a low-dimensional space not necessarily generalize well to high-dimensional sparse data sets (Domingos, 2012). The vertical size

[1] https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29.

[2] https://www.kaggle.com/c/loan-default-prediction.

[3] https://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval).

of the data (the number of observations) requires faster optimization procedures. Recent advancements in large scale convex optimization are briefly discussed in this article in application to LGD prediction task.

Datasets analyzed for predicting probability of default were fairly well balanced (about 30% of defaults). In a real-life setting, it is more likely to encounter significantly more imbalanced data sets, since the rate of default on consumer credit in the U.S., for example, was only 1.5% in 2012 (2013). The kaggle dataset exhibits roughly 10% of defaults, this being a better representation of the real-life data. It also presents additional challenges. Particularly, it is not entirely clear whether the asymptotic consistency of the cross-validation procedure is preserved in this setting.

Additionally, most past research has focused exclusively on one of the two parameters (PD and LGD) described before. Loss given default research has commonly utilized training data consisting only of defaulters for their regression models, while PD-research has only investigated classification methods. In this article we consider both tasks together, developing a hybrid approach using a single dataset.

Finally, the features of the kaggle dataset were completely anonymous. The organizers of the challenge intentionally did not provide any information that would describe the features in any way, naming them simply as F1 ... Fm. The following plots show the distribution of the response variable, highlighting the skewness of the data.

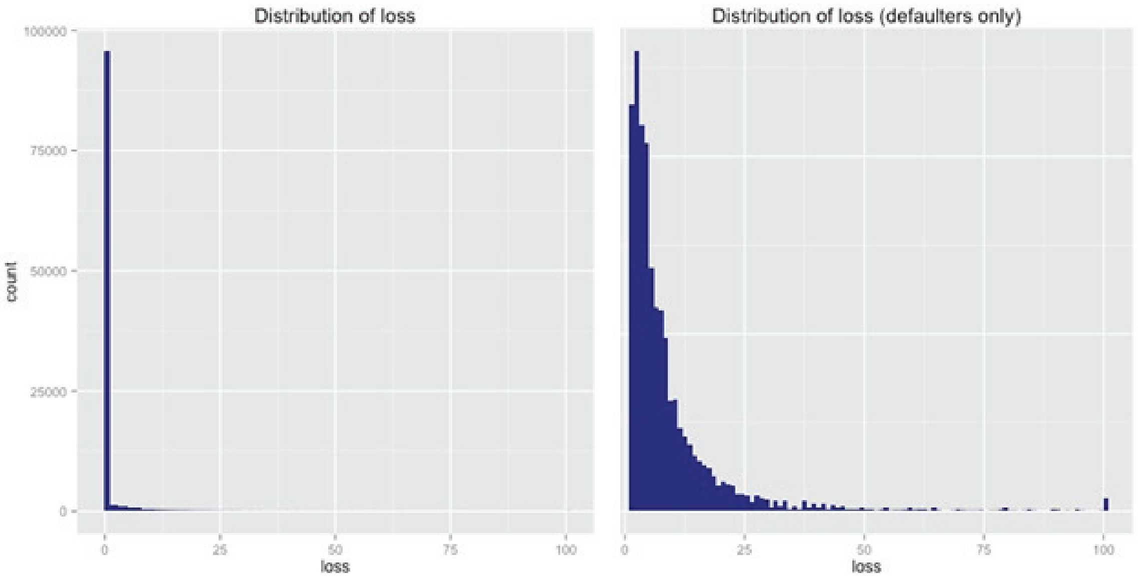

Figure 1. Target Variable Distribution

Experimental Set-Up

The kaggle data were preprocessed in two steps. First, missing values were imputed with the column means because most predictive algorithms in the package used cannot handle missing values, and there is evidence that missing value imputation improves prediction accuracy (Batista and Monard, 2003). Then, the data were scaled. Scaling the data is due to several practical considerations. Some values in the original data were so high that the software package could not handle them and raised an infinity error. Furthermore, especially for support vector machines it is strongly recommended to scale the data before analysis (Hsu et al., 2010).

Two metrics are considered for the purpose of technique benchmarking. F1 score is used to assess the accuracy of the defaulters classification task:

$$F1 = \frac{2 * recall * precision}{recall + precision} \qquad (3)$$

with

$$recall = \frac{TP}{TP + FN}, precision = \frac{TP}{TP + FP}, \qquad (4)$$

where *TP* is the sum of *true positives*, i.e. the number of observations correctly classified as "1" (loan default), *FN* is the sum of *false negatives,* i.e. the number of observations falsely classified as "0" (no loan default), and *FP* is the sum of *false positives,* i.e. the number of observations falsely classified as "1". F1 is a very popular metric to measure the performance of binary classifiers (Zhang and Zhang, 2004). The higher *F*1 score corresponds to the better performance of the classifier. Mean absolute error (*MAE*) is used to measure the performance of *LGD* estimation. *MAE* was also the used to evaluate the contenders of the "kaggle" challenge. The comparison of estimation techniques is facilitated through the use of five-fold cross-validation procedure, which in most cases produces a reasonable tradeoff between variance and bias (Kohavi, 1995).

Feature Selection

Feature subset selection is the process of selecting features to be used in machine learning models. This research direction emerged with the (horizontal, number of parameters) growth of data available, reaching a preliminary peak in 2003, with a special issue of the Journal of Machine Learning Research devoted to it. The primary motivations for feature selection are:

- improvement of predictive accuracy;
- data storage;
- computational cost.

Feature selection methods can be loosely grouped in 3 categories: filters, wrappers and embedded methods. Filters apply simple, mostly univariate, correlation based criteria to detect relationships between individual features and the response. They thus act independently of the chosen learning algorithm. Univariate approaches are usually fast, but not always accurate. With wrapper methods, the predictive performance of an algorithm is compared for different subsets of features. Depending on the search method chosen, this could deliver very good results but comes at a potentially high, or even prohibitively high, computational cost – consider, for instance, an exhaustive search over our dataset with 759 features.

Embedded methods are learning algorithms which already incorporate implicit feature selection. Examples include l1-regularized ("sparse") models and decision trees. The resulting models can either be used for predictive purposes on their own or the selected features can be fed to another algorithm. Embedded methods are able to capture interdependencies among the features better than filers and wrappers. For that reason, as well as for computational considerations, we chose to compare the latter method, namely l1-regularized linear models, to the first. We hypothesize initially that only a small fraction of the available features are actually relevant for predictive purposes. So we choose to have the univariate feature selection procedure for the top 50 features, and set the regularization parameter of the l1-based feature selection parameter achieving roughly the same number of features.

Classification and Regression Techniques

The table below summarizes different prediction techniques that are compared in this article. We split them in two categories: one-step and two-step. One step techniques perform only the binary classification task thus labeling defaulters. Instead of predicting the precise LGD value for the defaulters a certain general rule is used. Two-step techniques use regression tools to predict the LGD value based on the features given.

Further, we introduce a naïve strategy of assigning "0" expected loss value to all observations. That is predicting that nobody will default at all. Apart from being compared to each other the "naïve" strategy is used as a benchmark, and those models that can outperform it are considered to be viable solutions for the predictive task.

Table 2

Techniques descriptionModel	Classification	Regression	Type
One-step			
LOG-only	Logistic Regression	1 assigned to all defaulters	linear

Two-step			
LOG/OLS	Logistic Regression	Linear Regression	linear
LOG/RiR	Logistic Regression	Ridge Regression	linear
Linear-SVM	Linear Support Vector Classification	Support Vector Regression (Linear Kernel)	linear
BT	Stochastic Gradient Boosted Tree Classification	Stochastic GBT Regression	non-linear

Results and Discussion

The table below summarizes the results for various methods achieved without the use of feature selection. The results for the models without feature selection are shown below.

Table 3

Results – No Feature Selection

Model	**F1**	**MAE**
Naïve	N/A	0.8
Log-Only	0.8400	0.7412
Log/OLS	0.8400	0.9859
Log/RiR	0.8400	0.8158
LinearSVM	0.5729	1.1771
BT	0.9333	0.4899

The results for the techniques where feature selection was applied are summarized below.

Table 4

Results – 50 best features (based on ANOVA-F)

Model	**F1**	**MAE**
Naïve	N/A	0.8
Log-Only	0.7133	0.7738
Log/OLS	0.7133	1.0351
Log/RiR	0.7133	1.0339
LinearSVM	0.4878	1.4686
BT	0.8443	0.5830

Table 5

Results – l1 feature selection

Model	**F1**	**MAE**
Naïve	N/A	0.8
Log-Only	0.8341	0.7429
Log/OLS	0.8350	0.8272
Log/RiR	0.8343	0.8315
LinearSVM	0.8165	0.6808
BT	0.9227	0.5088

From the tables above we can conclude that *BT* is the best technique. Linear*SVM* shows good performance in case of l1 feature selection. *BT* clearly outperforms the Linear*SVM* not only general performance (*MAE* score), but also in the classification stage (*F*1).

Other techniques are beaten by the naïve strategy only in presence of the l1 feature selection (regularization). We believe that is largely due to the overfitting problem. Overfitting may occur either due to sparseness of data caused by many features or due to certain properties of the underlying process that the above techniques fail to capture. One of such properties is of non-linear relationship between the features and *LGD*. Non-linearity is likely to occur in the regression stage: we observe a relatively high *F*1 score of linear models for classification (good classification performance). It is especially surprising that Ridge regression, being robust to overfitting, fails to produce a desirable result after successful classification.

ANOVA-F based feature selection also did not improve the performance of the techniques failing to capture the right structure of the features. This is clearly a too primitive technique for our dataset with many features, where correlation between the features has to be taken into account.

Our results are quite surprising. The performance gap between linear models, hardly managing to keep up with the naïve strategy, and non-linear model (*BT*) is much bigger than that discovered by Loterman et al. (2012). In their study the difference between *LOG/OLS* and best performing non-linear model (*ANN*) was 6% in the worst case, while in our experiments it was almost 50% difference (l1 feature selection case). Our results therefore strongly support the hypotheses that non-linear relationships in real-world loss given default data sets exist.

Conclusions and Future Research

We have confirmed the results of Loterman et al. (2012), namely that non-linear models perform better for loss given default prediction task. Our result was obtained on a dataset with much more features than the one of Loterman et al. (2012). The performance gap between traditional linear models and non-linear approaches, such as decision trees, was notably bigger than that found in Loterman et al. study. Among all tested models we found the model based on boosted decision trees and l1 feature selection to perform best in our scenario.

Our analysis provides valuable insight into the results of the machine learning challenge sponsored by the Imperial College London. The dataset used in this paper was supplied as part of this challenge. We address the theoretical argumentation for the choice of the models used also by the winners of the challenge.

Further research may address the problem of ranking model performance across various different dataset as suggested in (Demšar, 2006). Additionally, bootstrapping techniques can be used instead of cross-validation to rank the models.

References

1. *2013. National Credit Default Rates Decreased in March 2013 According to the S&P/Experian Consumer Credit Default Indices* [Online]. UBM plc. (Accessed 05/05 2014).
2. *Basel commitee on banking supervision 2005.* An Explanatory Note on the Basel II IRB Risk Weight Functions. Bank for International Settlements.
3. Batista, G.E.A.P.A., and Monard, M.C. (2003), “An Analysis of Four Missing Data Treatment Methods for Supervised Learning”, *Artificial Intelligence*, no. 17, pp. 519–533.
4. Bluhm, C., Overbeck, L., and Wagner, C. (2003), *“An Introduction to Credit Risk Modelling”,* CRC Press LLC.
5. Demšar, J. (2006), “Statistical comparisons of classifiers over multiple data sets”, *The Journal of Machine Learning Research*, no. 7, pp. 1–30.
6. Domingos, P. (2012), “A Few Useful Things to Know About Machine Learning”, *Communications of the ACM*, no. 55, pp. 78–87.
7. Heiat, A. (2012), “Comparing Perfomance of Data Mining Models for Consumer Credit Scoring”, *Journal of International Finance & Economics*, no. 12, pp. 78–82.
8. Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2010), “A Practical Guide to Support Vector Classification”.
9. Kohavi, R. (1995), *“A study of cross-validation and bootstrap for accuracy estimation and model selection”*, Ijcai. P.1137–1145.
10. Loterman, G., Brown, I., Martens, D., Mues, C., and Baesens, B. (2012), “Benchmarking regression algorithms for loss given default modeling”, *International Journal of Forecasting*, no. 28, pp. 161–170.

11. Tobback, E., Martens, D., Gestel, T.V., and Baesens, B. (2014), “Forecasting Loss Given Default models: impact of account characteristics and the macroeconomic state”, *Journal of the Operational Research Society,* no. 65, pp. 376–392.

12. Yap, B.W., Ong, S.H., and Husain, N.H.M. (2011), “Using data mining to improve assessment of credit worthiness via credit scoring models”, *Expert Systems with Applications*, no. 38, pp. 13274–13283.

13. Zhang, H., and Zhang, X. (2004), “Data Mining Static Code Attributes to Learn Defect Predictors”, IEEE Transactions on Software Engineering. P. 33.