ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/299821566

On Class Imbalance Correction for Classification Algorithms in Credit Scoring

Conference Paper · February 2016

DOI: 10.1007/978-3-319-28697-6_6

READS

30

3 authors, including:

Bernd Bischl

Ludwig-Maximilians-University of Munich

44 PUBLICATIONS **277** CITATIONS

SEE PROFILE

All in-text references underlined in blue are linked to publications on ResearchGate, letting you access and read them immediately.

Available from: Bernd Bischl
Retrieved on: 08 July 2016

On Class Imbalance Correction for Classification Algorithms in Credit Scoring

Bernd Bischl, Tobias Kühn and Gero Szepannek

Abstract Credit scoring is often modeled as a binary classification task where defaults rarely occur and the classes generally are highly unbalanced. Although many new algorithms have been proposed in the recent past to mitigate this specific problem, the aspect of class imbalance is still underrepresented in research despite its great relevance for many business applications. Within the "Machine Learning in R" (mlr) framework methods for imbalance correction are readily available and can be integrated into a systematic classifier optimization process. Different strategies are discussed, extended and compared.

1 Introduction

Credit scoring denotes the assignment of ordered values (scores) to individuals that are supposed to be decreasing with risk. Here, risk is interpreted as the probability of a lender to default in the future. Business application scoring models are a major element in credit decisions and the IRBA Basel capital framework [19, 1].

In order to model credit risk, typically a binary random variable with two outcomes (default and non-default) and classification algorithms are used. The most widely-used technique is logistic regression, but in the recent past several new models have been proposed and studies have compared performances in different data situations [17] as well as on real world credit scoring data [2, 13].

One complicating factor in credit scoring is that classes typically follow a highly unbalanced distribution, i.e., the default class is much smaller. The effect of this on the performance of different classification algorithms has been investigated by Brown and Mues [7]. Vincotti and Hand [20] discuss both the introduction of mis-

---

The opinions expressed in this paper are those of the authors and do not reflect views of any organization or employer.

Bernd Bischl, LMU München, Germany e-mail: bernd_bischl@gmx.net

classification costs at either scorecard construction or classification stage and preprocessing of the training sample by over- or undersampling of the classes. The effect of over- and undersampling in relation to effective class sizes has been extensively investigated by Crone and Finlay [9].

In this article, we study different strategies for imbalance correction together with different classifiers in a comprehensive setting, introducing several new aspects:

- Joint framework for tuning of classifiers and imbalance correction.
- Newer techniques like SMOTE [8] and *overbagging* are investigated.
- Extension of SMOTE through the Gower distance for categorical data.
- Iterated F-racing instead of grid search [13] for tuning within mlr [5, 16].
- Large data base of real world data sets and validation on credit scoring data.
- Realistic evaluation of logistic regression using coarse classed data.

2 Methodology

Imbalance Correction: A standard approach for class imbalance correction consists in sampling [9]: In *undersampling* a random subset of the majority class is used for training, whereas *oversampling* randomly duplicates instances of the minority class. Some classifiers allow *weighting* of observations during training, which is a straightforward, alternative *intrinsic imbalance correction* to sampling, if one downweights majority and upweights minority class observations. Oversampling can be extended to *overbagging*, where the oversampling of the minority class is repeated several times. Majority class instances are bootstrapped in each iteration and and for the new training sets we fit a bagging predictor in order to reduce prediction variance. The popular *synthetic minority over-sampling (SMOTE)* [8] generates new observations of the minority class as random convex combinations of neighboring observations. As categorical features occur in many real-world problems, we use the Gower distance in this mixed space to identify neighbors and sample a new category for each categorical feature from the respective two entries of the neighbors during the convex combination step.

The **mlr R package** [6] offers an interface to more than 50 classification, regression and survival analysis models, and most standard resampling and evaluation procedures. Models can be chained and extended with, e.g., preprocessing operations and jointly optimized. The package allows for different optimization / configuration techniques, from simple random search, to iterated F-racing and sequential model based optimization. The latter two are arguably among the most popular and successful approaches for algorithm configuration nowadays.

Iterated F-racing [14, 12] builds upon the simpler racing technique, where algorithm candidate configurations are sequentially evaluated on a stream of instances. After each iteration, a statistical test is performed - usually the non-parametric Friedman test - to identify outperformed candidates, which are eliminated from the candidate set. In our case, candidates are joint hyperparameter settings for classifiers and imbalance correction and instances are subsampled versions of the training data

set. Iterated F-racing samples one set of candidate configurations from a joint distribution over the parameter space, performs a usual F-race to reduce the current candidates to number of elite configurations and adapts the distribution by centering it around the elites as well as reducing its spread. The latter results in exploration in the beginning and exploitation in the later stages of the optimization.

3 Experiments

A typical problem in credit scoring research is the availability of data so that most studies are based on only a few data sets [2]. In order to obtain general results we follow a two-fold approach: First, all methods are evaluated on a large set of public unbalanced data sets[1], among them the popular German credit data, which is not very representative due to the low degree of imbalance (30%) and the low number of observations. In a second step, we validate the results on two more realistic real world credit scoring problems: gmsc[2] and glc [15].

/bin/sh: 1: Manual: not found /bin/sh: 2: Although: not found We address this by generating additional binned data sets (suffix "nom") using decision trees with varying complexity parameters [18] and subsequent manual investigation of bins concerning numbers of defaults and default rates using binomial tests. The manual step implies a loss in scientific rigor, but allows to assess the results with respect to industrial practice.

As a general preprocessing step, constant features are removed from the data sets. Afterwards, five classification techniques, *logistic regression (logreg)*, *rpart decision tree (rpart)*, *random forest (RF)*, *gradient boosting (gbm)* and *support vector machines (ksvm)* with a Gaussian kernel, are applied to each data set. We combine all classifiers with all mentioned imbalance correction techniques. In this context, we study the following variants: classifiers without tuning or imbalance correction (bl = baseline), normal tuning of hyperparameters (tune), and joint tuning of the classifier and an imbalance correction method like class weighting (cw), undersampling (us), oversampling (os), SMOTE (sm) and 10 iterations of overbagging (ob).

Learner	Tuning Parameters with Range (lower, upper)
gbm	n.trees (100, 5000) / interaction.depth (1, 5) / shrinkage (1e-05, 0.1) / bag.fraction (0.7, 1)
ksvm	C (2^{-12}, 2^{12}) / sigma (2^{-12}, 2^{12})
logreg	-
RF	ntree (10, 500) / mtry (1, 10)
rpart	cp (0.0001, 0.1) / minsplit (1, 50)

Table 1 Overview of tuning parameters (arguments of corresponding R functions) for each learner.

[1] `http://www.cs.gsu.edu/~zding/research/benchmark-data.php`

[2] `http://www.kaggle.com/c/GiveMeSomeCredit`

The parameter controlling the upsampling ratio / minority class upweighting is tuned in the range of 1 and $1.5 \times IR$, where IR is the class imbalance ratio. For undersampling we use a range of $0.67 \times IR^{-1}$ and 1.

For all set-ups (except the baseline) tuning is performed via iterated F-racing and a budget of 400 evaluations. During the inner resampling for tuning (in each racing step) we use 80% of the observations for training and 20% for testing. The whole tuning / model selection process is embedded into an outer loop of stratified 5-fold cross-validation to ensure unbiased performance estimation. As it represents a standard for credit scoring applications the area under the ROC curve (AUC) is used both as a measure for tuning and performance evaluation [4, 11]. We parallelize our experiments via the BatchJobs and BatchExperiments R packages [3].

4 Results and Summary

Data	Learner	Base	Tuning	Data	Learner	Tuning	Imbal	Method
balance	gbm	0.29	0.89	poker	rpart	0.47	0.76	sm
poker	gbm	0.53	1.00	abalone19	rpart	0.56	0.81	ob
balance	ksvm	0.68	0.92	balance	rpart	0.50	0.73	ob
mammography	gbm	0.71	0.94	balance	logreg	0.29	0.50	us
gmsc	gbm	0.66	0.87	solar flare m0	ksvm	0.62	0.82	sm
satellite image	gbm	0.78	0.97	ozone level	rpart	0.67	0.84	ob
abalone7	rpart	0.50	0.67	poker	logreg	0.34	0.52	us
vehicle	gbm	0.69	0.86	abalone7	rpart	0.67	0.83	os
coil2000	rpart	0.50	0.66	oil spill	rpart	0.70	0.85	ob
glc	rpart	0.70	0.85	balance	RF	0.36	0.50	ob

Table 2 Effect of tuning vs. imbalance correction: top ten AUC improvements of the baseline by tuning (left, columns 1 to 4) as well as improvements of tuning by additional imbalance correction together with the best strategy (right, columns 5 to 8).

Learner	Tuning Mean	Imbal Mean	Imbal Max	Method
gbm	0.14	0.02	0.04	ob
ksvm	0.05	0.04	0.06	us
logreg	0.00	0.05	0.13	us
RF	0.00	0.04	0.14	ob
rpart	0.04	0.13	0.29	sm

Table 3 Effect of tuning vs. imbalance correction: Mean improvements per learner by tuning across data sets (left) and further improvements by imbalance correction across data sets, averaged over all sampling methods and best sampling method on average (right).

The tables show the results of the conducted experiments. Often strong improvements are achieved, mostly using upsampling strategies - which are unfortunately

Data	IR	N	Feat	Learner	Base	Tuning	Weights	Sampling	Method	Rate
vehicle	2.90	846	18	ksvm	0.868	0.926	0.489	**0.935**	os	2.90
satellite image	9.28	6435	36	ksvm	0.935	0.967	0.965	**0.968**	sm	9.29
abalone7	9.68	4177	10	ksvm	0.774	0.85	0.865	**0.87**	os	8.72
balance	11.76	625	20	ksvm	0.68	**0.917**	0.857	0.765	os	8.13
us crime	12.29	1994	100	ksvm	0.87	0.927	0.925	**0.928**	sm	4.17
yeast ml8	12.58	2417	103	ksvm	0.592	0.605	**0.619**	0.605	sm	9.80
scene	12.60	2407	294	RF	0.763	0.783	0.804	**0.815**	os	9.64
coil2000	15.76	9822	85	gbm	0.685	0.758	0.762	**0.762**	os	15.64
solar flare m0	19.43	1389	32	ksvm	0.628	0.619	0.814	**0.818**	sm	18.00
oil spill	21.85	937	48	RF	0.93	0.923	0.907	**0.948**	sm	10.59
yeast2vs8	23.10	482	8	RF	0.927	0.847	0.906	**0.929**	os	14.54
wine quality4	25.77	4898	11	RF	0.898	0.898	0.874	**0.907**	sm	4.15
yeast uci me2	28.10	1484	8	RF	0.93	0.923	0.915	**0.934**	os	16.44
ozone level	33.74	2536	72	ksvm	0.845	0.886	0.915	**0.916**	os	17.07
yeast6	41.40	1484	8	gbm	0.903	0.945	0.944	**0.954**	sm	33.08
mammography	42.01	11183	6	gbm	0.708	0.943	**0.953**	0.949	os	22.18
poker	58.40	1485	10	gbm	0.525	0.998	**1**	1	os	26.38
abalone19	129.53	4177	10	logreg	0.816	0.816	0.816	**0.842**	os	153.70
gcd	2.33	1000	19	RF	**0.798**	0.792	0.781	0.787	os	2.98
gcd nom	2.33	1000	20	logreg	**0.787**	0.787	0.787	0.784	os	2.78
glc	11.91	28882	26	gbm	0.788	0.922	**0.924**	0.922	os	9.49
glc nom	11.91	28882	19	logreg	**0.909**	0.909	0.909	0.909	os	8.94
gmsc	13.96	150000	10	gbm	0.656	0.865	**0.866**	0.864	os	8.94
gmsc nom	13.96	150000	10	logreg	0.86	0.86	0.86	**0.861**	os	13.72

Table 4 AUC of the best algorithm (bold) and learner per dataset. In comparison AUC of the same learner without tuning or imbalance correction (Base), with tuning of hyperparameters only (Tuning), - and class weights (Weights) - and the best sampling method (Sampling). Method gives the name of the sampling strategy, the best found sampling rate / class upweighting parameter for this method is shown in Rate.

the computationally most expensive ones. Also, these improvements are only observed in combination with proper hyperparameter tuning, especially for SVMs and boosting which reflects their strong dependence on parameterization. Decision trees are most strongly affected by imbalance correction, followed by random forests and logistic regression. Note that in some rare cases the results after imbalance correction worsen, which might be due to an overfitting on the validation sets. The results do not uniquely favor a single combination of methods and the picture is much less clear than in [10], where only decision trees and no tuning was considered. Nevertheless, boosting and upsampling (sm or os) seem to be a good choice in many cases.

For credit scoring, the established pre-binned logistic regression shows good results, but improvements by the proposed integrated tuning and imbalance correction framework are visible. This is especially noteworthy, as the pre-binning comes with a substantial time investment for the human expert, while the automated one does not.

References

1. Baesens, B., van Gestel, T. : Credit Risk Management – Basic Concepts Oxford University Press, NY (2009)
2. Baesens B., Van Gestel T., Viaene S., Stepanova M., Suykens J., Vanthienen J.: Benchmarking state of the art classification algorithms for credit scoring, Journal of the Operational Research Society **54 (6)**, 627–635 (2003)
3. Bischl, B., Lang, M., Mersmann, O., Rahnenführer, J., Weihs, C.: BatchJobs and BatchExperiments: Abstraction mechanisms for using R in batch environments (ACCEPTED), Journal of Statistical Software (2015)
4. Bischl, B., Schiffner, J., Weihs, C.: Benchmarking local classification methods, Computational Statistics, **28 (6)**, 2599–2619 (2013)
5. Bischl, B., Schiffner, J., Weihs, C.: Benchmarking classification algorithms on high-performance computing clusters. In: Spiliopoulou, M. Schmidt-Thieme, L. and Janning, R. (eds), Data Analysis, Machine Learning and Knowledge Discovery, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 23–31. Springer, Heidelberg (2014)
6. Bischl, B., Lang, M., Richter, J., Judt, L.: mlr: Machine Learning in R. R package version 2.0. `http://CRAN.R-project.org/package=mlr` (2014)
7. Brown I., Mues C.: An experimental comparison of classification algorithms for imbalanced credit scoring data sets, Expert Systems with Applications **39 (3)**, 3446–3453 (2012)
8. Chawla N.V., Bowyer K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority oversampling technique, Journal of Artificial Intelligence Research **16**, 321–357 (2002)
9. Crone S., Finlay S.: Instance Sampling in Credit Scoring: an empirical study of sample size and balancing, International Journal of Forecasting **28 (1)**, 224–238 (2012)
10. Galar, M., Fernandez, A., Barrenechea Tartas, E., Bustince Sola, H., Herrera, F.: A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches, IEEE Trans. on Systems, Man, and Cybernetics, Part C **42 (4)**, 463-484 (2012)
11. Koch, P., Bischl, B., Flasch, O., Bartz-Beielstein, T., Weihs, C., Konen, W.: Tuning and evolution of support vector kernels, Evolutionary Intelligence **5 (3)**, 153–170 (2012)
12. Lang, M., Kotthaus, H., Marwedel, P., Weihs, C. Rahnenführer, J., Bischl, B.: Automatic Model Selection for High-Dimensional Survival Analysis, Journal of Statistical Computation and Simulation (2014)
13. Lessmann S., Seow H.-V., Baesens, B., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: A ten-year update, `http://www.business-school.ed.ac.uk/waf/crc_archive/2013/42.pdf` (2013)
14. Lopez-Ibanez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M.: The irace Package: iterated racing for automatic algorithm configuration, Technical Report TR/IRIDIA/2011-004, IRIDIA, Bruxelles (2011)
15. Strackeljahn, J., Jonscher, R., Prieur, S., Vogel, D., Deslaers, T., Keysers, D., Mauser, A., Bezrukov, I., Hegerath, A.: GfKl Data mining competition 2005 – predicting liquidity crisis of companies. In: Spiliopoulou, M., Kruse, R., Borgelt, C., Nürnberger, A., Gaul, W. (eds.) From Data and Information Analysis to Knowledge Engineering, pp. 748–758. Springer (2005)
16. Szepannek, G., Gruhne, M., Bischl, B., Krey, S., Harczos, T., Klefenz, F., Dittmar, C.,Weihs, C.: Perceptually based phoneme recognition in popular music. In Locarek-Junge, H., Weihs,C. (eds.) Classification as a Tool for Research, pp. 751758. Springer, Heidelberg (2010)
17. Szepannek, G., Schiffner, J., Wilson, J.C., Weihs, C.: Local modelling in classification. In: Perner, P. (ed.) Advances in Data Mining: Medical Applications, E-Commerce, Marketing, and Theoretical Aspects, pp. 153-164. Springer LNAI 5077, Berlin (2008)
18. Therneau, T., Atkinson, E.: In introduction to recursive partitioning using RPART routines, TR 61, Mayo Foundation, `http://www.mayo.edu/hsr/techrpt/61.pdf` (1997)
19. Thomas, L.C., Edelman, D.B., Crook, J.N.: Credit Scoring and Its Applications. SIAM (2002)
20. Vincotti T., Hand D.: Scorecard construction with unbalanced class sizes, Journal of the Iranian Statistical Society **2**, 189–205 (2002)