SoftwareX 10 (2019) 100341

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

IRIC: An R library for binary imbalanced classification

Bing Zhu [a,*], Zihan Gao [a], Junkai Zhao [a], Seppe K.L.M. vanden Broucke [b]

[a] Business School, Sichuan University, Chengdu 610064, PR China
[b] Department of Decision Sciences and Information Management, KU Leuven, Leuven 3000, Belgium

ARTICLE INFO

Article history:
Received 3 May 2019
Received in revised form 21 September 2019
Accepted 2 October 2019

Keywords:
Imbalanced classification
R language
Integrated library
Parallel implementation

ABSTRACT

Imbalanced classification is a challenging issue in data mining and machine learning, for which a large number of solutions have been proposed. In this paper, we introduce an R library called IRIC, which integrates a wide set of solutions for imbalanced binary classification. IRIC not only provides a new implementation of some state-of-art techniques for imbalanced classification, but also improves the efficiency of model building using parallel techniques. The library and its source code are made freely available.

Code metadata

Current code version	v1.1
Permanent link to repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2019_167
Legal Code License	GNU general public license v3.0
Code versioning system used	Git
Software code language used	R ($\geq$ 3.3.0)
Compilation requirements, operating environments & dependencies	R, Rtools
Link to developer documentation/manual	https://github.com/shuzhiquan/IRIC/blob/master/ReferenceManual.pdf
Support email for questions	zhubing@scu.edu.cn

1. Introduction

Classification on class-imbalanced data is a hot research topic in data mining and machine learning [1]. Class imbalance relates to the context where the number of instances of one class is significantly outnumbered by those of other classes, with the minority class typically being of most interest to the modeler, which is very common in many real-world applications. Class imbalance poses a challenge, as standard classifiers are often biased towards the majority classes on class-imbalanced data sets [2], which often results in a high overall accuracy, but a low accuracy for the minority class. However, the minority class is typically the class of interest. Consider for example a binary-class data set containing 98 negative instances and 2 positive instances. If all the instances are classified as being negative, a high overall accuracy of 98% is obtained, although the accuracy for the minority, positive instances is zero.

Plenty of techniques have been proposed to alleviate the impact of the class imbalance in the past few years. Many of them have been implemented in different programming languages such as Java, R and Python. R is one of the preferred tools for data analysis due of its flexibility and rich visualization capabilities, with some of the class imbalance solutions being available in several existing R packages. This said, to the best of our knowledge, there is no R package containing a comprehensive and unified set of solutions to tackle the setting of imbalanced classification, which is inconvenient for academic research as well as industrial applications, as one typically needs to resort to dealing with multiple packages. In addition, some popular algorithms have not yet been made available in R. As such, in this paper, we introduce an R library named IRIC (**I**ntegrated **R** library for **I**mbalanced **C**lassification) to be used in the setting of binary imbalanced classification. The main contributions of IRIC are threefold: First, this is the first R library integrating a large collection of the solutions to the binary imbalanced classification task. Second, it provides an implementation of novel, recent algorithms which are not available in any current R packages. Third, the library provides a parallel implementation of bagging-based ensemble solutions to

* Corresponding author.
E-mail address: zhubing@scu.edu.cn (B. Zhu).

https://doi.org/10.1016/j.softx.2019.100341

Table 1
Overview of available R packages for imbalanced classification.

Package	Version	Strategy	No.	Techniques
imbalance [3]	1.0.0	Data level	12	**ADASYN**, ANSMOTE BLSMOTE, DBSMOTE **MWMOTE**, PDFOS RACOG, RSLSMOTE RWO, SLMOTE **SMOTE**, wRACOG
unbalanced [4]	2.0	Data level	9	**ubOver**, **ubUnder** **SMOTE**, Tomek ubOSS, ubCNN ubENN, ubNCL
smotefamily [5]	1.3	Data level	6	**ADASYN**, ANS, SLS Borderline-SMOTE, RSLS DBSMOTE, **SMOTE**
ebmc [6]	1.0.0	Ensemble-based	4	**RUSBoost**, **SMOTEBagging** **SMOTEBoost**, **UnderBagging**
DMwR [7]	0.4.1	Data level	1	**SMOTE**
ROSE [8]	0.0–3	Data level	1	**Random Oversampling**

reduce the runtime of model construction. We expect IRIC to offer a convenient framework for researchers and practitioners alike in the setting of binary imbalanced classification.

The remainder of this paper is organized as follows: Section 2 presents the problem and background. Section 3 describes the software framework in more detail. Section 4 provides more implementation details together with empirical results. Section 5 shows an illustrative example and finally Section 6 presents the conclusions and future work.

2. Problem and background

Imbalanced classification is quite common in many different fields such as marketing [9], finance [10] and medical diagnosis [11]. To reduce the negative influence of class imbalance, a large number of methods have been proposed. In general, these techniques can be categorized into three strategies: those that work on the algorithm level, those that work on the data level, and finally approaches that utilize ensemble-based learning [12]. The algorithm level approaches attempt to adjust existing algorithms or develop new ones to bias them more towards identifying the minority class in imbalanced classification. Typical example approaches are cost-sensitive learning techniques such as cost-sensitive decision trees [13]. The data level approaches try to balance the class distribution by resampling from the original data in the data preprocessing stage and are agnostic of the classifier technique used, making them a popular choice in practice. Synthetic Minority Oversampling Technique (SMOTE) is probably the most well-known data level approach [14]. The ensemble-based approaches usually embed the algorithm level or data level approaches into ensemble-based learning algorithms. For instance, AdaC2 [2] integrates a cost-sensitive method into the ensemble learning framework AdaBoost. SMOTEBagging [15] combines SMOTE sampling and bagging based ensemble models.

Many proposed approaches from the three strategies outlined above have been implemented in different languages. For Java, two well-known Java software tools KEEL and Weka provide functions to deal with imbalanced classification. KEEL has a module named Imbalanced Classification, which contains many algorithms from the three different families of techniques for imbalanced classification [16]. Meanwhile, Weka [17] is limited to cost-sensitive and simple sampling based solutions. For Python, the library imbalanced-learn [18] provides a wide set of methods for imbalanced classification and works on top of the popular scikit-learn data mining library. Specifically, it contains four groups of methods: undersampling, oversampling, combinations of oversampling and undersampling and ensemble-based learning.

As an increasingly popular platform, several R packages are also made available in the CRAN package repository for imbalanced classification. Recently, the package imbalance [3] was published, which offers oversampling based techniques, including five novel oversampling algorithms. Another package, unbalanced [4], contains nine well-known sampling techniques for imbalanced classification, such as random oversampling, SMOTE, and One Side Selection (OSS). Next, the smotefamily [5] package provides a collection of various oversampling techniques, implementing SMOTE and its variants. The ebmc [6] package contains four popular ensemble-based solutions, namely SMOTEBoost, RUSBoost, UnderBagging and SMOTEBagging. The DMwR [7] package implements a complete list of methods to carry out different data mining tasks. It is not specifically designed to address the imbalanced classification task, though it does include the SMOTE algorithm. Finally, the package ROSE [8] contains a sampling method called Random Over-Sampling Examples (ROSE).

Table 1 provides a comparative summary of R packages for imbalanced classification. Techniques that were included in our library IRIC have been marked in bold. As can be observed, each package implements a subset of all possible approaches over the three strategies outlined above, and there is no R package which contains a wide set of solutions from the three different strategies. This lack of integration makes it time-consuming for practitioners and researchers to find their way throughout all the available approaches and set up comparative studies. Moreover, we also find that some classical algorithms have not been implemented in R, such as RBBagging [19], EasyEnsemble and BalanceCascade [20].

3. Software framework

IRIC is written in the R language. Details of the IRIC library are presented in Code metadata. The current version (v1.1) includes 19 popular solutions towards binary imbalanced classification from the three strategies mentioned before.

The architecture of IRIC is illustrated in Fig. 1. IRIC is composed of three modules, which corresponds to the three categories of solutions, i.e. those at the algorithm level, the data level and ensemble-based strategies. The algorithm level module contains the cost-sensitive technique, which supports the cost-sensitive decision tree proposed by [13] at present. The data level module consists of three kinds of sampling methods named

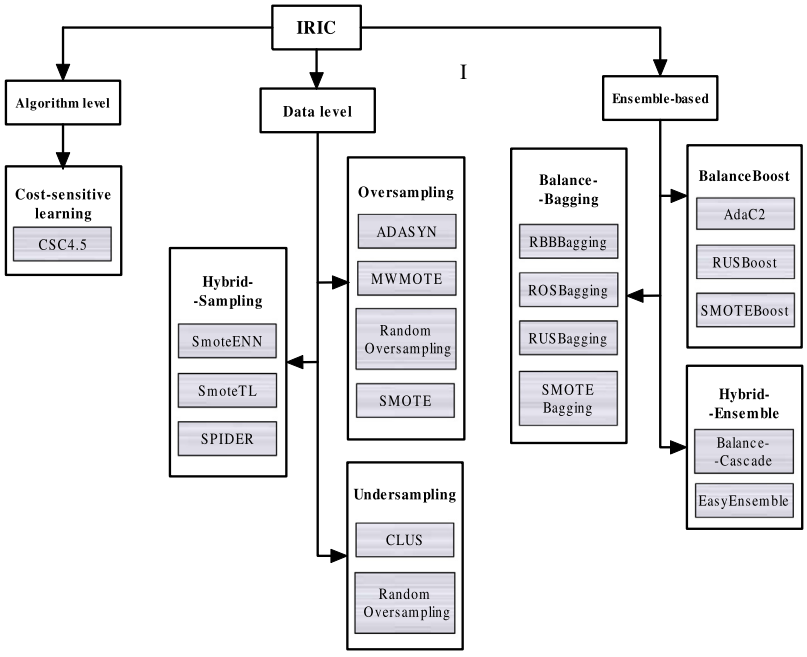

Fig. 1. IRIC library architecture.

Table 2
Summary of the data sets used in the experiment.

Data set	Source	Region	#Obs.	#Att.	IR (%)
Chile	Operator	South American	5 300	41	6.0
KDD	KDDcup2009	Europe	50 000	231	7.3
Korean	Operator	East Asia	26 224	11	4.2

undersampling, oversampling and hybrid sampling. The oversampling submodule includes SMOTE, MWMOTE [21], ADASYN [22] and Random Oversampling. The undersampling submodule contains two methods: Random Undersampling and CLUS [23]. SmoteENN, SmoteTL [24] and SPIDER [25] are included in the hybrid sampling submodule. The ensemble-based module has three submodules: BalanceBagging, BalanceBoost and HybridEnsemble. The BalanceBagging submodule has four bagging-based ensemble solutions: RUSBagging, ROSBagging, SMOTEBagging [15] and RBBagging [19]. The BalanceBoost submodule consists of three boosting-based solutions: RUSBoost [26], SMOTEBoost [27] and AdaC2 [2]. EasyEnsemble and BalanceCascade [20] make up the HybridEnsemble submodule.

The following eight methods are implemented in R for the first time, which are often used as a benchmark in existing literature:

- **CSC4.5**. CSC4.5 was proposed by [13] and associates a high cost with misclassifying minority instances to change the class distribution of training set to induce cost-sensitive trees.
- **SmoteENN**. SmoteENN is a sampling method proposed by [24]. It first samples from the original data set with SMOTE and then cleans up the sampled data by removing instances which are misclassified based on the Wilson's Edited Nearest Neighbors Rule (ENN) [28].
- **SmoteTL**. Similar to SmoteENN, SmoteTL also cleans the data set sampled with SMOTE. It cleans up the sampled data with Tomek links after SMOTE sampling.
- **SPIDER**. SPIDER (Selective Preprocessing of Imbalanced Data) was proposed by [25]. It is a sampling method which filters difficult instances from the majority class after local over-sampling of the minority class.
- **RBBagging**. Roughly Balanced Bagging (RBBagging) is an ensemble-based learning proposed by [19]. The key point of the RBBagging is that it uses negative binomial sampling to generate the roughly balanced data subsets to train the base classifiers.
- **AdaC2**. AdaC2 was proposed by [2] to improve the performance of classification with imbalanced data. It introduces the cost items into the AdaBoost learning framework by embedding the cost items into the weights update formula of AdaBoost to bias the learning towards the minority class instances. We choose to implement AdaC2 instead of other cost-sensitive boosting techniques, such as AdaC1 and AdaC3 because of its better performance [2].
- **BalanceCascade**. Similar to EasyEnsemble, BalanceCascade [20] also uses bagging as the main ensemble method and uses Adaboost for training each bag. The critical difference is that BalanceCascade guides the sampling process for subsequent classifiers.
- **EasyEnsemble**. EasyEnsemble is a hybrid ensemble-based solution proposed by [20]. It utilizes a double ensemble scheme which carries out random undersampling in bagging iterations and uses AdaBoost to train the base classifier. It performs bagging in a unsupervised manner to train the base classifier.

4. Implementation and empirical results

The IRIC library and its documentation are made available on GitHub. To justify the implementation of techniques in IRIC, two sets of experiments are conducted in this section: First, we verify the efficiency of parallel implementation for the bagging-based techniques. Second, we compare the implementation of overlapping techniques between IRIC and other packages.

In the model building process, bagging-based ensemble learning solutions typically generate subsets for base classifiers in a sequential manner, whereas we provide a parallel implementation of bagging-based solution to generate the subsets for training. To evaluate its efficiency, we set up an experiment to compare the efficiency of model building in a sequential manner with that in the parallel approach. As was shown in Table 1, the ebmc package

Table 3
The runtime for model building based on SMOTEBagging (in seconds) and the AUC.

Data set	Runtime			AUC		
	Nonparallel/IRIC	Parallel/IRIC	Nonparallel/ebmc	Nonparallel/IRIC	Parallel/IRIC	Nonparallel/ebmc
Chile	467.34	4.87	433.44	0.714	0.722	0.705
KDD	4459.45	25.85	4747.25	0.692	0.668	0.681
Korean	660.36	16.64	693.03	0.740	0.804	0.795

Table 4
The experiment result of the overlapping techniques in different packages.

Technique	Package	Runtime			AUC		
		Chile	KDD	Korean	Chile	KDD	Korean
ADASYN	imbalance	2.94	65.21	6.75	0.676	0.653	0.663
	smotefamily	2.19	51.65	5.56	0.710	0.653	0.672
	IRIC	1.73	17.64	2.55	0.677	0.660	0.683
MWMOTE	imbalance	5.03	634.79	34.36	0.690	0.617	0.692
	IRIC	2.12	43.45	3.67	0.654	0.666	0.720
Random Oversampling	unbalanced	0.96	2.63	1.41	0.662	0.658	0.738
	ROSE	0.92	2.92	1.64	0.677	0.670	0.700
	IRIC	0.84	1.26	0.87	0.683	0.671	0.709
Random Undersampling	unbalanced	1.01	2.69	1.20	0.650	0.646	0.711
	ROSE	0.78	1.50	0.91	0.701	0.663	0.695
	IRIC	0.58	0.86	0.66	0.722	0.666	0.719
RUSBoost	ebmc	10.89	129.99	19.47	0.647	0.718	0.834
	IRIC	21.79	164.62	23.06	0.716	0.680	0.817
RUSBagging	ebmc	8.65	79.58	8.75	0.717	0.701	0.826
	IRIC	8.08	76.06	7.86	0.709	0.701	0.774
SMOTE	imbalance	3.12	53.82	6.04	0.668	0.650	0.686
	unbalanced	3.72	45.03	6.64	0.791	0.657	0.706
	smotefamily	3.03	56.36	6.25	0.715	0.648	0.683
	DMwR	1.83	59.68	4.53	0.683	0.641	0.674
	IRIC	1.67	6.96	1.81	0.683	0.656	0.696
SMOTEBoost	ebmc	108.4	3551.12	251.31	0.745	0.700	0.819
	IRIC	275.84	4453.6	552.47	0.765	0.758	0.644

also includes the implementation of the bagging-based solutions in the sequential manner. Therefore, the bagging-based solutions from ebmc were also used as a benchmark of sequential implementation. Three real-world binary imbalanced data sets from the telecommunication industry were used in this experiment [29]. Table 2 presents more details on the data sets, including the imbalance ratio (IR), the number of observations (#Obs.) and the number of attributions (#Attr.). The number of bags was set to be 40 and CART [30] was used as the base learner. The experiment was conducted on R (v3.6.1) using computer with a 2.2 GHz Intel i5 dual-core CPU and 4 GB RAM memory. The experiment results of SMOTEBagging are presented in Table 3. Specifically, the runtime refers to the average time of model building over five runs.

As Table 3 shows, the parallel implementation accelerates the model building process significantly for SMOTEBagging. For example, the runtime on the "Chile" data set is reduced from more than 400 s either based on ebmc or IRIC to 4.87 s with the parallel implementation. These improvements hold in other bagging-based algorithms implemented in IRIC.

As for the overlapping techniques comparison, experiments were run with the same three data sets and computation environment mentioned above. The experiment results are presented in Table 4 in Appendix. In the experiments, Naive Bayes was used as the classifier for the sampling methods. The results in Table 4 show that the implementation of the overlapping techniques in IRIC has comparative performance with those in other packages for both classification performance and modeling time.

5. Illustrative example

In this section, we will offer an example for the usage of IRIC based on RBBagging. Data set "Korean" (see Table 2) was used as example for model training and prediction as follows:

```
1  # Example of RBBagging
2  # Load the package caret for data partitioning
3  library(caret)
4  # Load an example data set
5  load("Korean.RDa")
6  #Import the provided script file for RBBagging
7  source("BalanceBagging.R")
8  # Data split
9  sub<-createDataPartition(Korean$Churn,p=0.75,
10 list=FALSE)
11 trainset<-Korean[sub,]
12 testset<-Korean[-sub,]
13 str(trainset)
14 # Call RBBaging for model training
15 train_RB<-bbaging(trainset[,1:30],
16 trainset[ ,"Churn"], type="RBBagging")
17 # Prediction
18 pre_RB<-predict(train_RB, testset[,1:30],
19 type = "class")
```

In the code presented above, lines 3 and 5 load the necessary packages and data. Line 7 loads the function file provided by IRIC. Lines 9–11 apply a data splitting procedure and lines 13–16 show the model building and prediction process based on RBBagging with the training set and test set, respectively. Users can refer to online documentation and user manuals at GitHub repository for more detailed usage examples.

6. Conclusions

It is of great importance to deal with the class imbalance issue in many practical classification tasks. In this paper, we presented an R library called IRIC to improve the efficiency of imbalanced classification by unifying several approaches, many of which have been implemented for the first time in R. The library provides 19 popular techniques for imbalanced classification from three different groups of solutions, of which eight are new implementations in R. IRIC also improves the efficiency of bagging-based approaches by using parallelism, which was shown to reduce the runtime significantly. In future work, we will keep maintaining the library and incorporate more prevalent techniques for imbalanced classification into it.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 71401115) and fund from Sichuan University, PR China (Grant No. skqy 201742).

Appendix. Comparison of the overlapping techniques

See Table 4.

References

[1] Krawczyk B. Learning from imbalanced data: open challenges and future directions. Prog Artif Intell 2016;5(4):221–32. http://dx.doi.org/10.1007/s13748-016-0094-0.

[2] Sun Y, Kamel MS, Wong AK, Wang Y. Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit 2007;40(12):3358–78. http://dx.doi.org/10.1016/j.patcog.2007.04.009.

[3] Cordón Ignacio, García Salvador, Fernández Alberto, Herrera Francisco. Imbalance: preprocessing algorithms for imbalanced datasets. 2018, R package version 1.0.0. https://CRAN.R-project.org/package=imbalance.

[4] Dal Pozzolo Andrea, Caelen Olivier, Bontempi Gianluca. Unbalanced: racing for unbalanced methods selection. 2015, R package version 2.0. https://CRAN.R-project.org/package=unbalanced.

[5] Siriseriwan Wacharasak. Smotefamily: a collection of oversampling techniques for class imbalance problem based on SMOTE. 2019, R package version 1.3.1. https://CRAN.R-project.org/package=smotefamily.

[6] Hao Hsiang, Chen. Ebmc: ensemble-based methods for class imbalance problem. 2017, R package version 1.0.0. https://CRAN.R-project.org/package=ebmc.

[7] Torgo L. Data mining with R, learning with case studies. Chapman and Hall/CRC; 2010, http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR.

[8] Lunardon Nicola, Menardi Giovanna, Torelli Nicola. ROSE: a package for binary imbalanced learning. R J 2014;6(1):82–92.

[9] Amin A, Anwar S, Adnan A, Nawaz M, Howard N, Qadir J, et al. Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study. IEEE Access 2016;4:7940–57. http://dx.doi.org/10.1109/ACCESS.2016.2619719.

[10] Sanz JA, Bernardo D, Herrera F, Bustince H, Hagras H. A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data. IEEE Trans Fuzzy Syst 2015;23(4):973–90. http://dx.doi.org/10.1109/TFUZZ.2014.2336263.

[11] Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD. Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. Neural Netw 2008;21(2):427–36. http://dx.doi.org/10.1016/j.neunet.2007.12.031.

[12] Fernández A, García S, Galar M, Prati RC, Krawczyk B, Herrera F. Learning from imbalanced data sets. Springer International Publishing; 2018, http://dx.doi.org/10.1007/978-3-319-98074-4_2.

[13] Kai MT. An instance-weighting method to induce cost-sensitive trees. IEEE Trans Knowl Data Eng 2002;14(3):659–65. http://dx.doi.org/10.1109/TKDE.2002.1000348.

[14] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artificial Intelligence Res 2002;16(1):321–57.

[15] Wang S, Yao X. Diversity analysis on imbalanced data sets by using ensemble models. In: 2009 IEEE symposium on computational intelligence and data mining. IEEE; 2009, p. 324–31.

[16] Triguero I, González S, Moyano JM, García S, Alcalá-Fdez J, Luengo J, et al. KEEL 3.0: an open source software for multi-stage analysis in data mining. Int J Comput Intell Syst 2017;10(1):1238–49.

[17] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 2009;11(1):10–8.

[18] Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. J Mach Learn Res 2017;18(17):1–5, http://jmlr.org/papers/v18/16-365.html.

[19] Hido S, Kashima H, Takahashi Y. Roughly balanced bagging for imbalanced data. Stat Anal Data Min 2010;2(5–6):412–26.

[20] Liu X-Y, Wu J, Zhou Z-H. Exploratory undersampling for class-imbalance learning. IEEE Trans Syst Man Cybern B 2009;39(2):539–50. http://dx.doi.org/10.1109/TSMCB.2008.2007853.

[21] Barua S, Islam MM, Yao X, Murase K. MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans Knowl Data Eng 2013;26(2):405–25. http://dx.doi.org/10.1109/TKDE.2012.232.

[22] He H, Yang B, Garcia EA, Li S. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: IEEE international joint conference on neural networks; 2008. p. 1322–28. http://dx.doi.org/10.1109/IJCNN.2008.4633969.

[23] Lin W-C, Tsai C-F, Hu Y-H, Jhang J-S. Clustering-based undersampling in class-imbalanced data. Inform Sci 2017;409:17–26.

[24] Batista GE, Prati RC, Monard MC. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor Newsl 2004;6(1):20–9.

[25] Stefanowski J, Wilk S. Selective pre-processing of imbalanced data for improving classification performance. In: International conference on data warehousing and knowledge discovery. Springer; 2008, p. 283–92. http://dx.doi.org/10.1007/978-3-540-85836-2_27.

[26] Seiffert C, Khoshgoftaar TM, Hulse JV, Napolitano A. RUSBoost: a hybrid approach to alleviating class imbalance. IEEE Trans Syst Man Cybern 2010;40(1):185–97. http://dx.doi.org/10.1109/TSMCA.2009.2029559.

[27] Chawla NV, Lazarevic A, Hall LO, Bowyer KW. SMOTEBoost: Improving prediction of the minority class in boosting. In: European conference on principles of data mining and knowledge discovery. Springer; 2003, p. 107–19.

[28] Wilson DL. Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybern 1972;SMC-2(3):408–21. http://dx.doi.org/10.1109/TSMC.1972.4309137.

[29] Zhu B, Baesens B, vanden Broucke SK. An empirical comparison of techniques for the class imbalance problem in churn prediction. Inf sci 2017;408:84–99.

[30] Breiman L, Friedman J, Olshen R, Stone C. Classification and regression trees. Wadsworth and Brooks; 1984.