Symulacja w Badaniach i Rozwoju
Vol. 8, No. 3-4/2017

Paweł SZESZKO
E-mail: p.szeszko@gmail.com

Magdalena TOPCZEWSKA
Politechnika Białostocka, Wydział Informatyki
ul. Wiejska 45A, 15-351 Białystok
E-mail: m.topczewska@pb.edu.pl

A new hybrid approach for data level balancing classes in classification problems

1 Introduction

The process of decision making applies to all areas of science and real life problems. The quality of data is the most important issue if the aim is to automate this process and build the classification decision boundaries on the basis of the information taken directly from a dataset. Firstly, the variables should be collected to describe the phenomenon as well as it is possible. Secondly, the data should be cleaned and checked for correctness. Despite adequate steps, some problems can be found due to the characteristics of the phenomenon. The lack of balance between classes' cardinalities can be the example. In such a case classes in a dataset are distinguished as: *minority* and *majority* class. The minority class is the set of objects with smaller cardinality, while the majority class is the set with larger cardinality. Usually the appropriate classification of the minority class objects is the area of interest, for instance the differentiation between the sick suffering a rare disease and the healthy or patients with other diseases. The results obtained in the classical classification process can be incorrect and biased only because the classification errors for two classes are treated equally.

Three main approaches have been established to overcome the problem of the imbalance between number of objects in classes. The first group - data level techniques - introduce the preprocessing step to the analysis by modification the cardinalities of classes to balance finally the number of objects in the classes. The second group - algorithm level techniques - do not change the data but operate on the algorithms. The implementations are adjusted to treat the minority class objects with special attention. The third group is formed by the combination of techniques previously mentioned.

This article concerns the methods belonging to the data level approaches and a new algorithm is compared with other techniques.

2 Algorithms

One of the most well-known algorithms is SMOTE (Synthetic Minority Over-sampling Technique) [5], that enables to create new elements belonging to the minority class. For each object of smaller class, among its k nearest neighbours belonging to the same class, C/100 elements are chosen randomly, where C% denotes the number of new

generated objects. To calculate the positions of new objects, the difference between values of attributes of considered object and its one neighbour is calculated by multiplying it by a vector of random numbers (each number between 0 and 1), and adding the result to the values of the attribute vector of considered object. Application such an algorithm produces more generalized decision regions and, therefore, improves the classification results [5]. Two variants of the methods are used in the paper: a variant that generates N=100% of synthetic minority class objects (SMOTE100) and a variant that balances the number of minority and majority class objects (SMOTEAuto).

The Borderline-SMOTE algorithm [6] focuses on the objects located close to the decision borders. In these places the largest number of misclassification errors can be observed. Thus, the aim is to strengthen the elements situated near the borders, hence the elements located further from the interface of classes have limited impact on decision making. In this paper two techniques are considered: Borderline-SMOTE (BordSMOTE) and Borderline-SMOTE2 (BordSMOTE2) - the difference involves the selection of danger elements (with at least a half of their neighbours from opposite class).

The Safe-Level-SMOTE algorithm [2] is another modification of SMOTE [5]. It pays particular attention to instances of classes which surround the examined minority class object. For each object the method defines a *safe level* indicator denoting the number of positive class objects among its k nearest neighbours. Due to the value of the indicator new artificial elements are created differently.

The CORE algorithm [3] is the expansion of the Safe-level SMOTE technique [2] and is complemented by the elements of the MUTE algorithm [4]. It is a hybrid technique that deletes majority class object with safe level indicator at a certain level, and creates new minority class objects using Safe-Level SMOTE algorithm. Two variants of this technique are used in the paper: a variant that generates (N=100% of minority class objects) synthetic objects (CORE100) and a variant that balances the number of minority and majority class objects (COREAuto).

The last method - SPIDER - is the selective preprocessing technique [7]. Each object is indicated as safe or noisy depending on the correctness of k nearest neighbours classification result. Three variants of a dataset modification was proposed. A weak amplification, denoted in this paper as SPIDER-W, increases the importance of minority class objects indicated as noisy by copying them.

A weak amplification and relabeling variant (SPIDER-WL) introduces additional step - some noisy majority class elements are relabelled as minority class objects. A strong amplification variant (SPIDER-STR) focuses on all minority class objects. Safe elements are copied and the number of new synthetic elements depends on the number of safe majority class objects among their three nearest neighbours. The modification of noisy elements depends on the five nearest neighbours classification results.

All mentioned algorithms are compared to the results obtained for the original datasets without any preprocessing step.

3 HImbA algorithm

The HImbA algorithm, presented in the Listing 1, is a hybrid technique that uses SMOTE and modified *k*-nearest neighbours method. It consists of four steps:

- finding nearest neighbours,
- generating synthetic objects,
- checking if new objects aren't placed in unwanted regions,
- changing class membership of objects from majority class considered as noise.

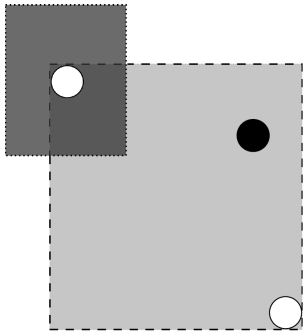

Fig. 1. Influence of a random number generation range limitation on a possible location of a new object

Modified method of *k* nearest neighbours operates on the entire training set. For each object with the usage of Euclidean metric the distances to other elements of a dataset are calculated and pairs (distance, neighbour) are stored in increasing order. Using this results, global average distance *R* as average of distances from each minority object to its five nearest neighbours from the same class is calculated. This value is then used to select nearest neighbours in the next phases of the algorithm.

The second step is new minority class objects generation and locating them in a temporary set *syntheticSet*. This phase is a modified version of SMOTE algorithm. Number of synthetic objects is adjusted to temporarily balance the number of objects in minority and majority classes. For instance, when *IR*=5, for each minority object four synthetic objects are generated. The next change is a change of method of choice neighbours taking part in creation of new objects. Algorithm choices randomly one of the nearest neighbours from the same class that are located at a distance less than or equal to *R*, calculated in previous step. If there is no minority class neighbour that meet this condition, algorithm creates a copy of current minority object.

Listing 1. Algorithm 1: HImbA(W)

```
input:  Original input set W
output: Output set S
P = Elements from set W belonging to minority class
N = Elements from set W belonging to majority class
syntheticSet = Ø;
syntheticOutput = Ø;
toRelabel = Ø;
For each object from input set calculate distances to other objects
R = average distance of each minority class object to its 5 nearest neighbours from P
for each p ∈ P do
        for i = 1...(|N| - |P|)/|P| do
                T = neighbours of p belonging to P placed at a distance ≤ R
                if |N| > 0 then
                        r = random(1, |T|)
                        n = T[r]
                        m = nearest neighbour of object p belonging to majority class
                        dn = distance between p i n
                        dm = distance between p i m
                        if dm < dn then
                                max = 0.5 * dm/dn
                        end
                        else
                                max = 1.0
                        end
                        s = createSynthetic(p, T[r], T;-max; max)
                end
                else
                        s = copy(p)
                end
                syntheticSet = syntheticSet ∪ s
        end
end
for each s ∈ syntheticSet do
        T = neighbours of s belonging to P placed at a distance ≤ R
        if |T| ≥ 2 then
                syntheticOutput = syntheticOutput ∪ s
        end
end
for each p ∈ P do
        T = neighbours of p belonging I placed at a distance ≤ R
        for each n ∈ T do
                if hasMinorityClass(n) break;
                NN = neighbours of n belonging to W placed at a distance ≤ R
                PNN = neighbours of n belonging to P placed at a distance ≤ R
                if |PNN| ≥ pr * |NN| then
                        toRelabel = toRelabel  ∪ n
                end
        end
end
for each n ∈ toRelabel do
```

Set class membership of object *n* to minority class
end
S = *W* ∪ *syntheticOutput*
return

To prevent algorithm from creating synthetic objects in undesirable regions we modified the range of values from random gap coefficient is chosen. SMOTE algorithm does not pay attention to synthetic objects neighbourhood. It may cause some of them are placed in majority class regions. This problem has been shown in the figure 1. White circles represent objects from minority class and black ones - from majority class. SMOTE can place new objects in light gray area, thus in regions that are unwanted. To minimize risk of occurrence of such a situation, algorithm looks for nearest majority class neighbour m of current minority class object that is located at distance lower than or equal to *R*. We calculate *max* value as half of quotient of distance between *m* and current minority object and distance between random minority class neighbour and current minority object. If there is no majority class object located at distance lower than or equal to *R*, *max* value is set to 1.0. Range of values from which gap coefficient is randomly chosen is <-*max*, *max*>. Area, in which now synthetic element may be placed has been marked as dark gray rectangle.
In third step we check every synthetic object if it is not noise. New object is placed in *sytheticOutput* set only if in its neighbourhood (at distance ≤ R) there are at least two objects from minority class.
In the last step we change class membership to minority class of some of majority class objects considered as noise. This action is performed when at distance ≤ R at least *pr%* of neighbours is from minority class.

4 Experiments

In the experiment the datasets from two repositories: KEEL [1] and the UCI (*University of California at Irvine Repository*) [8] were used. The characteristics of the selected 28 data sets are presented in Table 1. To evaluate 10-fold cross-validation was used, repeated additionally 10 times with different random generator seed to minimize the consequences of the randomness on the results. Four indicators calculated from the perspective of a minority class were taken into consideration and reported: precision, recall, F-measure and AUC. The *pr* parameter was set at 80% for the best obtained results.
The results are presented in table 2 and table 3. Due to the fact that calculated results considered as data cannot be described by normal distribution in table 2 levels of median values as main descriptive statistics for each preprocessing filter are reported. In the description of the results average values of standard deviations can be mentioned in brackets, but for information purposes only.

Table 1. Characteristics of the datasets

dataset	#no. of objects	#no. of attributes	#no. of minority class objects	IR
breast-cancer	286	9	85	2.36
bupa	345	6	145	1.37
ecoli-0 vs 1	220	7	77	1.86
ecoli1	336	7	77	3.36
ecoli2	336	7	52	5.46
ecoli3	336	7	35	8.60
ecoli4	336	7	20	15.80
german-credit	1000	20	300	2.33
glass*	214	9	51	3.20
glass0	214	9	70	2.06
glass1	214	9	76	1.82
glass4	214	9	13	15.46
haberman	306	3	81	2.78
hepatitis	155	19	32	3.84
iris0	150	4	50	2.00
new-thyroid1	215	5	35	5.14
new-thyroid2	215	5	35	5.14
phoneme	5404	5	1586	2.40
pima	768	8	268	1.87
thoracic-surgery	470	16	70	5.71
transfusion	748	4	178	3.20
vehicle0	846	18	199	3.25
vehicle1	846	18	217	2.90
vehicle2	846	18	218	2.88
vehicle3	846	18	212	2.99
vowel0	988	13	90	9.98
yeast1	1484	8	429	2.46
yeast3	1484	8	163	8.10

* glass-0-1-2-3_vs_4-5-6

The global median of precision for two classification methods and preprocessing filters equals 0.57 with the range between 0.08 and 1.00 (0.63±0.21). The difference in medians between using preprocessing approaches and results obtained for datasets with no preprocessing step is -0.13, while the difference in average values is at the level of -0.05 that means the 8% of average precision decrease by the usage of filters. The median of precision in the case of C4.5 classification method equals 0.61 with the range between 0.15 and 1.00 (0.64±0.21); in the case of SMO algorithm is 0.55 with the range between 0.08 and 1.00 (0.62±0.22).

Considering the recall indicator, the global median for both classification techniques and preprocessing approaches equals 0.84 with the range 0.02-1.00 (0.79±0.18). The difference in medians between using preprocessing filters and the case without filters is 0.23. The difference in average values of recall equals 0.20, thus the average increase in recall by the usage of preprocessing filters is at the level of 33.65%. For the C4.5 method the median is 0.76 with the range 0.16-0.98 (0.76±0.17); while for SMO: 0.86 with the range 0.02-1.00 (0.81±0.18).

Table 2. Results for preprocessing approaches and two classification methods - median values of precision (P), recall (R), F-measure (F), area under the curve (AUC) for 28 datasets

	Classification method							
	C4.5				SMO			
approach	P	R	F	AUC	P	R	F	AUC
Original	0.69	0.64	0.67	0.79	0.71	0.59	0.62	0.74
BordSMOTE	0.67	0.76	0.68	0.81	0.56	0.85	0.66	0.82
BordSMOTE2	0.67	0.78	0.68	0.83	0.56	0.85	0.66	0.83
CORE100	0.53	0.86	0.67	0.82	0.52	0.89	0.64	0.84
COREAuto	0.53	0.87	0.67	0.84	0.50	0.92	0.64	0.85
HImbA	0.60	0.84	0.67	0.88	0.58	0.86	0.65	0.82
SLSMOTE	0.69	0.76	0.72	0.83	0.55	0.86	0.65	0.84
SMOTE100	0.65	0.74	0.69	0.81	0.61	0.78	0.66	0.77
SMOTEAuto	0.65	0.80	0.72	0.87	0.55	0.86	0.65	0.83
SPIDER-STR	0.61	0.73	0.67	0.82	0.52	0.88	0.65	0.81
SPIDER-W	0.61	0.72	0.67	0.82	0.55	0.82	0.66	0.79
SPIDER-WR	0.60	0.76	0.67	0.82	0.54	0.85	0.66	0.79

For the F-Measure the global median for all used preprocessing filters and two classification methods is 0.66 with the range between 0.04-1.00 (0.69±0.19). The difference in medians comparing to the no filters approach equals 0.02, while the difference between average values of F-Measure is 0.07 that means the usage of filters caused the increase in F-Measure at the level of 10.48%. In the case of C4.5 the median of F-Measure is 0.67 with the range 0.16-0.99 (0.69±0.19); in the SMO case: median equals 0.66 with the range between 0.04 and 1.00 (0.69±0.19).

Table 3. Results for preprocessing approaches and two classification methods - median values for differences of precision (P), recall (R), F-measure (F), area under the curve (AUC) for 28 datasets

	Classification method							
	C4.5				SMO			
approach	P	R	F	AUC	P	R	F	AUC
BordSMOTE	-0.028	0.089	0.019	0.005	-0.038	0.135	0.038	0.028
BordSMOTE2	-0.033	0.115	0.012	0.005	-0.042	0.145	0.036	0.031
CORE100	-0.092	0.193	0.007	0.001	-0.103	0.233	0.025	0.034
COREAuto	-0.093	0.220	0.009	0.003	-0.121	0.311	0.021	0.042
HImbA	-0.053	0.106	0.011	0.012	-0.062	0.185	0.037	0.031
SLSMOTE	-0.041	0.102	0.025	0.002	-0.092	0.178	0.023	0.028
SMOTE100	-0.028	0.072	0.019	0.006	-0.039	0.118	0.042	0.029
SMOTEAuto	-0.044	0.099	0.017	0.006	-0.075	0.189	0.037	0.037
SPIDER-STR	-0.061	0.111	0.003	0.006	-0.082	0.229	0.026	0.029
SPIDER-W	-0.060	0.110	0.002	0.010	-0.056	0.132	0.023	0.022
SPIDER-WR	-0.069	0.135	0.001	0.005	-0.066	0.158	0.029	0.027

The last reported indicator is the AUC - the global median for two classifiers and all mentioned filters equals 0.82 with the range between 0.48 and 1.00 (0.80±0.14). The difference in medians between using preprocessing filters and classification

methods without filters 0.06, whilst the difference between average values equals 0.03 – the increase in AUC indicator value by the usage of preprocessing filters is at the level of 4.20%. The results are also presented in the figure 2 and the figure 3.

The HImbA algorithm has a good effect on AUC measure. It reached significantly better results than other methods for some of the tested datasets: ecoli3 (SMO: AUC=0.888, originally 0.78; C4.5: AUC=0.873, originally 0.8), ecoli4 (SMO: AUC=0.935, originally 0.918; C4.5: AUC=0.935, originally 0.918), glass4 (SMO: AUC=0.922, originally 0.573), vowel0 (SMO: AUC=0.966, originally 0.886; C4.5: AUC=0.986, originally 0.969), new-thyroid2 (C4.5: AUC= 0.961, originally 0.941), yeast1 (SMO: AUC=0.743, originally 0.728), yeast3 (SMO: AUC=0.934, originally 0.884).

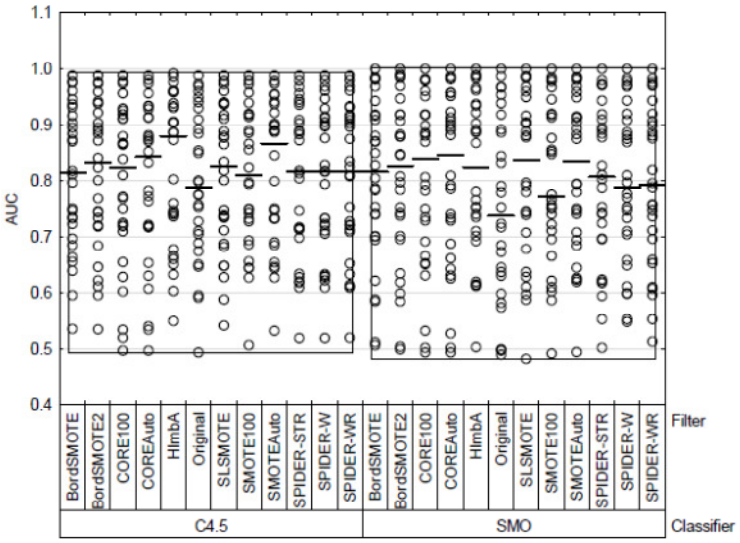

Fig. 2. The final results of AUC indicator for various filters

For two datasets it also reached the best F-measure value: bupa (SMO: F-measure=0.625, originally 0.53), yeast1 (C4.5: F-measure=0.586, originally 0.518).

For majority of datasets the best sensitivity values were obtained with CORE algorithm (+100% variant). Our algorithm reached very high value of sensitivity for ecoli4 dataset (0.91 for C4.5 and 0.9 for SMO) and vowel0 (0.984 for C4.5 and 0.999 for SMO), however it was associated with high decrease of precision. F-measure results for bupa and yeast1 datasets were not associated with the best value of sensitivity or precision.

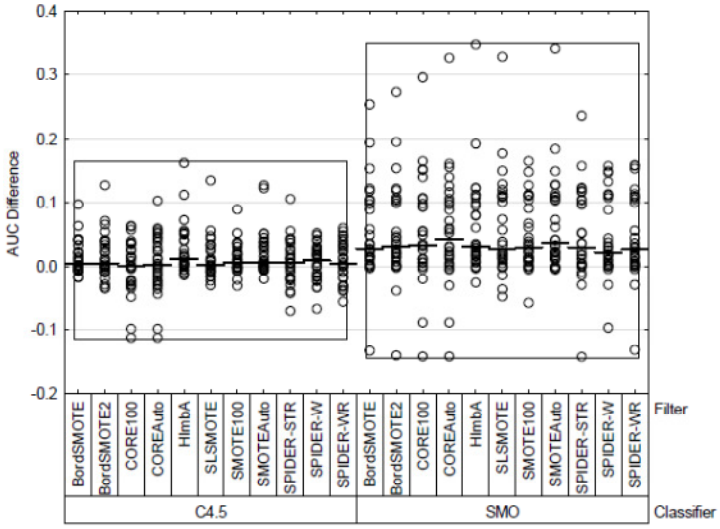

Fig. 3. The differences between AUC values fot the usage of preprocessing filters and the results for no filters classification approach

5 Conclusions

Imbalance in class cardinalities in datasets is a big problem nowadays. We have huge amounts of data that we need to process and make them valuable to users. Sometimes it is really hard to point what we should do to improve minority class object classification while not compromising to much overall results. It is really important when we want to support people in making hard decisions like treatment of people or spending money. One of the approaches can be imbalanced data preprocessing.

In this paper we introduced a new approach called HImbA. For some datasets it gives better results than other methods. There is a difference in algorithm efficiency when we use different classification algorithms, and our method works better for C4.5. The next step is to check how the algorithm works with additional preprocessing algorithms and with other classification techniques.

References

1. Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, and Salvador García. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing*, 17(2-3): 255-287, 2011
2. Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings, chapter Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem, pp. 475482. Springer, Berlin, Heidelberg, 2009

3. Chumphol Bunkhumpornpat and Krung Sinapiromsaran. Core: Core-based synthetic minority over-sampling and borderline majority under-sampling technique. *Int. J. Data Min. Bioinformatics*, 12(1): 4458, April 2015
4. Chumphol Bunkhumpornpat, Krungand Sinapiromsaran, and ChidchanokLursinsap. Mute: Majority under-sampling technique. In Information, Communications and Signal Processing (ICICS) 2011 *8th International Conference* on, pages 14, Dec 2011
5. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Ressearch*, 16:321357, 2002
6. Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August, 23-26, 2005, Proceedings, Part I, chapter Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning, pp. 878-887. Springer, Berlin Heidelberg, 2005
7. Jerzy Stefanowski, Szymon Wilk. DataWarehousing and Knowledge Discovery: 10th International Conference, DaWaK 2008 Turin, Italy, September 2-5, 2008 Proceedings, chapter Selective Pre-processing of Imbalanced Data for Improving Classification Performance, pages 283292. Springer, Berlin, Heidelberg, 2008
8. UC Irvine Machine Learning Repository, http://archive.ics.uci.edu/ml/, (accessed 20.05.2016)

Summary

The article concerns the problem of imbalanced data classification. A new algorithm is presented and tested. The HImbA technique is a hybrid method that uses well-known SMOTE algorithm and modified *k*-nearest neighbours method. 28 datasets have been preprocessed using the HImbA and 10 variants of existing techniques, classified using two algorithms (C4.5 and SMO) and the results have been compared. The new algorithm occurred to give the best results for some datasets.

Keywords: class imbalance, oversampling, classification

Nowe hybrydowe podejście równoważenienia liczności klas w problemie klasyfikacji

Streszczenie

Praca dotyczy braku zrównoważenia liczności klas w problemie klasyfikacji. Zaprezentowany oraz przetestowany został nowy algorytm. Technika HImbA jest metodą hybrydową, która łączy znany algorytm SMOTE oraz zmodyfikowaną wersję metody *k* najbliższych sąsiadów. Została ona zastosowana wraz z dziesięcioma wariantami istniejących technik w celu przetwarzania wstępnego 28 zbiorów danych, które zostały następnie poddane klasyfikacji (użyto dwóch algorytmów – C4.5 oraz SMO), a wyniki zostały porównane. Dla wybranych zbiorów przy użyciu nowego algorytmu uzyskano najlepsze rezultaty.

Słowa kluczowe: niezbalansowanie liczności klas, nadpróbkowanie , klasyfikacja

This work was performed in the framework of the grant S/WI/2/2013 (Bialystok University of Technology), founded by the Polish Ministry of Science and Higher Education.