Imbalanced data oversampling using one-class support vector machine classifier

Jakub Klikowski[0000−0002−3825−5514]

Department of Systems and Computer Networks
Wroclaw University of Science and Technology
Wroclaw, Poland
jakub.klikowski@pwr.edu.pl

Abstract. Classification of imbalanced data is an issue that still needs attention. Intelligent machine learning systems are increasingly finding various kinds of applications in the world of current technology. The major problem is insufficient data describing the minority class. Combined with the excessive amount of majority class samples, this reflects a skewed decision boundary toward the majority class. The effect of this is a relatively low predictive performance of classifiers. It is also worth noting that measuring quality itself can cause many difficulties. Inadequate selection of metrics such as accuracy can produce a seemingly good result for the highly biased model, which reflects the class distribution. One approach is data oversampling, which involves producing synthetic minority class samples. In this paper, a new technique that utilizes one-class support vector machines (OCSVM) for oversampling is proposed. Evaluated and compared with selected *state-of-the-art* methods shows promising performance and good ability to improve classification over the baseline method without oversampling.

Keywords: Imbalanced data · Oversampling · One-class SVM

1 Introduction

Classification of imbalanced data is a significant research problem [4]. Real-world data often has an uneven class distribution, which causes numerous difficulties when trying to classify it. One approach to dealing with such problems is data preprocessing [2]. These methods are split into two categories: undersampling, which reduces the majority class counts, and oversampling, which synthetically increases the number of majority class samples. One of the most recognized algorithms for oversampling is the SMOTE (Synthetic Minority Oversampling TEchnique) method [1]. The main idea of this approach is to generate new samples of the minority class sat between existing objects of this class. This method has many different variations, which create entirely new approaches based on this idea. One such variation is the Borderline SMOTE [3] approach, which primarily focuses on generating data near a potential decision boundary, where more majority class samples appear.

In this paper, a new method for oversampling imbalanced data will be proposed that uses a one-class SVM model for this purpose. A study will be conducted to verify if the proposed approach is able to improve the classification quality compared to selected *state-of-the-art* oversampling methods.

2 Proposed method

Oversampling means generating synthetic samples of a minority class to increase the importance of that class to others. These data should contain an appropriate distribution so that the relationship between objects from different classes is not disturbed. The main idea proposed in this paper focuses on selecting the best matching samples from the synthetically generated ones that follow the minority class distribution. The proposed One-Class Oversampling (OCO) method uses a one class SVM classifier for this selection process. This idea is flexible enough that other one-class models can be used.

Algorithm 1: One-Class Oversampling

Input: DS – Dataset
$OCSVM$ – One class SVM

Output: Oversampled DS

1 Divide DS into minority data (DS_m) and majority data (DS_M).
2 Train $OCSVM$ model on DS_m
3 $N \leftarrow$ Count number of samples in DS_M
4 Determine mean (μ) and standard deviation (σ) of DS_m
5 $X \leftarrow$ Generate N new synthetic samples,
where $X \sim \mathcal{N}(\mu, \sigma^2)$
6 $DS_n \leftarrow$ Predict X on $OCSVM$ model
7 $DS \leftarrow DS_M \cup DS_m \cup DS_n$
8 Return oversampled data set DS

Algorithm 1 presents the workflow of the proposed method in more detail. First, data is split into a minority class, and a majority class is performed. Then a one-class SVM classifier model is trained on the minority class data. Subsequently, the sample quantity in the majority class and the mean and standard deviation for the minority data are calculated. In the next step, new samples are generated according to the estimated normal distribution of the minority class data. Before oversampling, these samples are classified using the OCSVM model learned in step two. Any examples marked as outliers are discarded so that the minority class is reinforced with only the most matched objects. Then the newly composed data is returned by the algorithm. Project implementations are publicly available on the GitHub repository [1].

[1] https://github.com/w4k2/oco-pprai22

3 Experimental evaluation

The experimental study was conducted on a selected real-world data set with various features, samples, and imbalance ratios (Tab. 1). A 2-fold stratified cross-validation procedure with eight repeats was used for evaluation. The G-mean metric expressed classification quality. SMOTE, Borderline SMOTE (BSMOTE), Random Oversampling (ROS), and No Oversampling (NO) model algorithms were selected as reference methods. The support vector machine was used as the base classifier.

Table 1. G-mean metrics with datasets description and statistical analysis

Dataset	Features	Samples	Imb. Ratio	OCO (1)	BSMOTE (2)	SMOTE (3)	ROS (4)	NO (5)
car-vgood	7	1728	25.6	0.967 ± 0.042 *5*	0.972 ± 0.035 *5*	0.972 ± 0.037 *5*	**0.973 ± 0.032** *5*	0.783 ± 0.062
dermatology-6	35	358	16.9	0.959 ± 0.029 *2 5*	0.592 ± 0.459 *5*	0.961 ± 0.030 *1 2 5*	**0.961 ± 0.031** *1 2 5*	0.048 ± 0.128
ecoli-0-1-4-6_vs_5	7	280	13.0	**0.885 ± 0.070**	0.873 ± 0.097	0.876 ± 0.095	0.884 ± 0.096	0.875 ± 0.057
ecoli-0-1-4-7_vs_2-3-5-6	8	336	10.6	**0.851 ± 0.071** *5*	0.836 ± 0.050 *5*	0.842 ± 0.058 *5*	0.847 ± 0.052 *5*	0.745 ± 0.094
ecoli-0-1-4-7_vs_5-6	7	332	12.3	**0.889 ± 0.061** *5*	0.881 ± 0.045 *5*	0.886 ± 0.047 *4 5*	0.873 ± 0.038	0.856 ± 0.051
ecoli-0-1_vs_2-3-5	8	244	9.2	0.861 ± 0.053 *5*	0.841 ± 0.052 *5*	0.869 ± 0.044 *2 5*	**0.870 ± 0.058** *2 5*	0.812 ± 0.051
ecoli-0-1_vs_5	7	240	11.0	**0.897 ± 0.054** *5*	0.879 ± 0.061	0.888 ± 0.057	0.894 ± 0.053	0.874 ± 0.044
ecoli-0-2-3-4_vs_5	8	202	9.1	**0.893 ± 0.043** *2 5*	0.876 ± 0.053	0.892 ± 0.052 *2 5*	0.886 ± 0.051 *5*	0.867 ± 0.042
ecoli-0-2-6-7_vs_3-5	8	224	9.2	0.828 ± 0.063 *5*	0.825 ± 0.050 *5*	0.830 ± 0.058 *5*	**0.834 ± 0.055** *5*	0.789 ± 0.083
ecoli-0-3-4-6_vs_5	8	205	9.2	**0.898 ± 0.047** *5*	0.882 ± 0.063 *5*	0.887 ± 0.051 *5*	0.889 ± 0.053 *5*	0.843 ± 0.071
ecoli-0-3-4-7_vs_5-6	8	257	9.3	0.885 ± 0.038 *5*	0.884 ± 0.057 *5*	**0.889 ± 0.042** *5*	0.885 ± 0.038 *5*	0.838 ± 0.067
ecoli-0-3-4_vs_5	8	200	9.0	**0.894 ± 0.047** *5*	0.883 ± 0.066	0.888 ± 0.056 *5*	0.886 ± 0.055 *5*	0.867 ± 0.077
ecoli-0-4-6_vs_5	7	203	9.2	**0.883 ± 0.073** *2 5*	0.866 ± 0.077	0.867 ± 0.070	0.878 ± 0.066 *5*	0.853 ± 0.070
ecoli-0-6-7_vs_3-5	8	222	9.1	0.836 ± 0.065 *5*	0.825 ± 0.043 *5*	**0.839 ± 0.047** *5*	0.828 ± 0.049 *5*	0.788 ± 0.052
ecoli-0-6-7_vs_5	7	220	10.0	**0.883 ± 0.041** *4 5*	0.872 ± 0.037 *5*	0.872 ± 0.042 *5*	0.860 ± 0.036	0.838 ± 0.059
ecoli-0_vs_1	8	220	1.9	0.980 ± 0.013	0.980 ± 0.011	**0.982 ± 0.011**	0.981 ± 0.012	0.981 ± 0.013
ecoli1	8	336	3.4	0.878 ± 0.033 *5*	**0.886 ± 0.023** *3 5*	0.877 ± 0.024 *5*	0.882 ± 0.026 *5*	0.829 ± 0.058
ecoli2	8	336	5.5	**0.942 ± 0.027** *2 5*	0.924 ± 0.035	0.941 ± 0.026 *2 5*	0.937 ± 0.031 *5*	0.904 ± 0.053
ecoli3	8	336	8.6	0.890 ± 0.040 *5*	0.894 ± 0.032 *5*	0.891 ± 0.031 *5*	**0.894 ± 0.033** *5*	0.743 ± 0.066
ecoli4	8	336	15.8	**0.924 ± 0.031** *2 3 4 5*	0.897 ± 0.039	0.897 ± 0.038	0.897 ± 0.034	0.880 ± 0.079
flare-F	12	1066	23.8	**0.793 ± 0.055** *2 3 5*	0.744 ± 0.057 *5*	0.746 ± 0.057 *5*	0.775 ± 0.044 *2 3 5*	0.000 ± 0.000
led7digit-0-2-4-5-6-7-8-9_vs_1	8	443	11.0	**0.900 ± 0.015** *2 4*	0.863 ± 0.038	0.884 ± 0.032 *2 4*	0.871 ± 0.038	0.886 ± 0.037
newthyroid2	6	215	5.1	**0.925 ± 0.052** *2 3 4 5*	0.883 ± 0.047 *4 5*	0.861 ± 0.040 *4 5*	0.842 ± 0.050 *5*	0.397 ± 0.121
segment0	20	2308	6.0	0.935 ± 0.012 *5*	0.932 ± 0.014 *5*	**0.980 ± 0.006** *1 2 4 5*	0.977 ± 0.006 *1 2 5*	0.689 ± 0.025
vehicle0	19	846	3.3	0.755 ± 0.011 *5*	0.771 ± 0.015 *1 5*	**0.772 ± 0.012** *1 5*	0.772 ± 0.013 *1 5*	0.171 ± 0.213
vehicle1	19	846	2.9	**0.674 ± 0.012** *3 4 5*	0.670 ± 0.014 *3 4 5*	0.658 ± 0.017 *5*	0.660 ± 0.016 *5*	0.000 ± 0.000
vehicle2	19	846	2.9	**0.735 ± 0.029** *4 5*	0.720 ± 0.029 *5*	0.718 ± 0.024 *5*	0.712 ± 0.023 *5*	0.283 ± 0.028
vehicle3	19	846	3.0	**0.675 ± 0.018** *2 3 4 5*	0.663 ± 0.019 *4 5*	0.660 ± 0.021 *5*	0.659 ± 0.021 *5*	0.000 ± 0.000
yeast-0-2-5-6_vs_3-7-8-9	9	1004	9.1	0.793 ± 0.043 *5*	0.792 ± 0.026 *5*	**0.798 ± 0.028** *2 5*	0.796 ± 0.034 *5*	0.505 ± 0.082
yeast-0-2-5-7-9_vs_3-6-8	9	1004	9.1	**0.909 ± 0.017** *2 5*	0.901 ± 0.025 *5*	0.901 ± 0.028 *5*	0.900 ± 0.026 *5*	0.876 ± 0.028
yeast-0-5-6-7-9_vs_4	9	528	9.4	0.783 ± 0.030 *5*	0.780 ± 0.035 *5*	0.781 ± 0.036 *5*	**0.786 ± 0.032** *5*	0.134 ± 0.138
yeast-1-2-8-9_vs_7	9	947	30.6	**0.679 ± 0.090** *2 5*	0.604 ± 0.096 *5*	0.663 ± 0.067 *2 5*	0.653 ± 0.077 *5*	0.000 ± 0.000
yeast-1-4-5-8_vs_7	9	693	22.1	0.613 ± 0.057 *5*	0.575 ± 0.114 *5*	0.604 ± 0.055 *5*	**0.619 ± 0.046** *5*	0.000 ± 0.000
yeast-1_vs_7	8	459	14.3	0.698 ± 0.046 *5*	0.686 ± 0.083 *5*	**0.716 ± 0.057** *5*	0.695 ± 0.067 *5*	0.016 ± 0.063
yeast-2_vs_4	9	514	9.1	**0.896 ± 0.020** *3 4 5*	0.895 ± 0.033 *3 4 5*	0.879 ± 0.034 *5*	0.876 ± 0.030 *5*	0.750 ± 0.061
yeast3	9	1484	8.1	**0.917 ± 0.016** *2 3 5*	0.909 ± 0.012 *5*	0.911 ± 0.015 *5*	0.912 ± 0.015 *5*	0.808 ± 0.031
yeast4	9	1484	28.1	**0.820 ± 0.034** *2 3 5*	0.785 ± 0.036 *5*	0.792 ± 0.030 *5*	0.805 ± 0.017 *2 3 5*	0.000 ± 0.000
yeast5	9	1484	32.7	0.952 ± 0.022 *5*	0.954 ± 0.030 *5*	0.954 ± 0.031 *5*	**0.963 ± 0.020** *1 5*	0.291 ± 0.193
yeast6	9	1484	41.4	0.884 ± 0.036 *5*	0.871 ± 0.061 *5*	0.880 ± 0.034 *5*	**0.886 ± 0.034** *5*	0.000 ± 0.000

The results presented in Tab. 1 show the mean values with standard deviations. It is noticeable that the proposed approach obtains better results than other methods. The bold color indicates the best result among all. An important observation is that every time the proposed algorithm improves over the model without oversampling and very often obtains the best results.

Statistical analysis was also performed. Each method was compared pairwise with the other using a Student's t-test with alpha 0.05. Located below, the mean values and standard deviations indicate a statistically significant advantage of the technique in that column over the others. The information about which number defines the method is located in the first column of the table. It can be seen that the proposed approach very often obtains an advantage over the other methods. Another important observation is that OCO in the tested datasets always gets statistically significant better results than the model without oversampling.

4 Conclusions and future works

In this paper, a method for oversampling imbalanced data is proposed. Experimental evaluation has shown that the OCO method has good abilities to improve the model's predictive performance. This improvement is done at a comparable level to other state-of-the-art methods and sometimes exceeds their capabilities. Well, results are obtained for data sets with widely varying characteristics, without the influence of imbalance ratio, feature number, or samples amount.

The OCO algorithm has great potential for development. The properties of the method and different ways of generating new synthetic samples can be studied in the future. It is also worth conducting a much larger study using multi-class datasets and evaluating with more metrics.

Acknowledgements

This work was supported by the Polish National Science Centre under the grant No. 2017/27/B/ST6/01325 and by the research fund of the Department of Systems and Computer Networks, Faculty of Information and Communication Technology, Wroclaw University of Science and Technology.

References

1. Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. SMOTE: synthetic minority over-sampling technique. *Journal Of Artificial Intelligence Research.* **16** pp. 321-357 (2002)
2. Fernández, A., García, S., Galar, M., Prati, R., Krawczyk, B. & Herrera, F. Learning from imbalanced data sets. (Springer,2018)
3. Han, H., Wang, W. & Mao, B. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. *International Conference On Intelligent Computing.* pp. 878-887 (2005)
4. Krawczyk, B. Learning from imbalanced data: open challenges and future directions. *Progress In Artificial Intelligence.* **5**, 221-232 (2016)