DOI:10.25092/baunfbed. 1447440 *J. BAUN Inst. Sci. Technol., 26(2), 518-526, (2024)*

A novel undersampling method based on data classification method

Nur UYLAŞ SATI*

Muğla Sıtkı Koçman University, Bodrum Vocational High School, Department of Motor Vehicles and Transportation Technologies, 48100, Muğla, Türkiye

Geliş Tarihi (Received Date): 05.03.2024
Kabul Tarihi (Accepted Date): 06.06.2024

Abstract

Data mining is one of the most important research area in literature. Due to the increasing volume of data, which is directly proportional to technological advancements, the number of researches in this field is growing rapidly. The goal of data mining is to extract various insights and obtain information from raw data by leveraging machine learning techniques. The structural characteristics and also class distributions of the datasets used in machine learning techniques significantly affect the performances of the algorithms. In this study, our aim is balancing the imbalanced binary dataset, used in the machine learning techniques, with an undersampling approach including a classification method via polyhedral conic functions.

Keywords: *Data mining, machine learning, undersampling, polyhedral conic functions.*

Veri sınıflandırma yöntemine dayalı yeni bir alt örnekleme yöntemi

Öz

Veri madenciliği literatürdeki en önemli araştırma alanlarından biridir. Teknolojik gelişmelerle doğru orantılı olarak artan veri hacmi nedeniyle bu alanda yapılan araştırmaların sayısı da hızla artmaktadır. Veri madenciliğinin amacı, makine öğrenimi tekniklerinden yararlanarak çeşitli tahminlerde bulunmak ve ham verilerden bilgi elde etmektir. Makine öğrenmesi tekniklerinde kullanılan veri kümelerinin yapısal özellikleri ve sınıf dağılımları algoritmaların performanslarını önemli ölçüde etkilemektedir. Bu çalışmada amacımız, dengesiz ikili veri kümelerini, çokyüzlü konik fonksiyonların kullanıldığı bir sınıflandırma yöntemini içeren yeni bir alt örnekleme yaklaşımıyla dengelemektir.

* Nur UYLAŞ SATI, nursati@mu.edu.tr, https://orcid.org/0000-0003-1553-9466

Anahtar kelimeler: *Veri madenciliği, makine öğrenme, alt örnekleme, çokyüzlü konik fonksiyonlar.*

1. Introduction

Machine learning techniques used in data mining aims to find natural patterns in data that generate insight and help you make better decisions and predictions. They are used in variously real-world problems as in healthcare, agriculture, finance, retail, education, and more. In real-world data mining classification scenarios, imbalanced datasets are common, characterized by varying distributions of examples across different classes [1]. Researchers have increasingly focused on learning from such imbalanced data in recent years, with many attempting to address binary-class imbalanced problems [2]. A problem with a binary dataset is called imbalanced problem when the majority class (negative class) is significantly larger than the minority class (positive class). The straightforward method to solve the imbalanced machine learning problem is the resampling method by adding records to the minority class or deleting ones from the majority class [3]. In this paper, we have experimented a common approach, deleting the majority ones, called as undersampling.

The undersampling method typically employs random or clustering techniques to decrease the size of the majority class within the dataset, as certain data points within the majority class may not contribute significantly to the classification model [4].

The most commonly used undersampling method is Random Undersampling method that equalizes class distribution by randomly removing majority class instances. However, a significant drawback is the potential discarding of valuable samples from the majority class [5]. This approach has been integrated with various ensemble methods. UnderBagging, introduced by the authors in [6], combines a Random Undersampling technique with a bagging-based ensemble. In a similar vein, Seiffert et al. [7] introduced Rusboost, which integrates the Random Undersampling technique with boosting. Here, boosting adjusts the distribution of weights used to train the classifier towards the minority class and eliminates instances from the majority class. When recent years are investigated, in [8], a novel boosting-based algorithm named Ouboost is proposed for learning from imbalanced datasets. It combines the Peak under-sampling algorithm with the over-sampling technique (SMOTE) within the boosting procedure. As a result, they created temporary new datasets with lower imbalance levels than the original dataset. A novel Schur decomposition class-overlap undersampling method (SDCU) is proposed in [9] to find the global similarity of datasets. They showed that the performance of SDCU has obvious advantages compared with other state-of-the-art methods on three different types of classifiers: SVM, CART, and 3NN.

As can be seen from the literature the researches on undersampling methods are based on decreasing the size of the majority class by deleting the most redundant ones. In this approach, it is important to note that the majority elements to be discarded should not be overlooked as valuable and significant. In this paper a novel undersampling algorithm via classification is proposed. The main aim in the algorithm is to find the redundant majority class elements that will be deleted. Firstly, previously defined classification via polyhedral conic functions algorithm is performed to seperate the imbalanced binary classes. Then, the the separator function that separates the minority class from the

majority class is used to detect the majority class points that wrongly classified. These points can be thought as redundant data. After making some mathematical operations on them, the chosen ones are deleted from the original dataset for undersampling. To show the effectiveness of the proposed method, both imbalanced and balanced datasets are implemented on state of art classification algortihms.

The remainder of this paper is organized as follows: In section 2, previously defined polyhedral conic functions (pcfs) algorithm which is the basis of the suggested undersampling method is given. In section 3, the pseudocode of the suggested undersampling method is propesed and explained in detail. In section 4, the experimental results are presented and the results are discussed. And finally in section 5 the paper is finalized and future studies are suggested.

2. Polyhedral conic functions (PCFs)

Polyhedral conic functions (PCFs), initially propsed by Gasimov and Öztürk in 2006, were designed to separate two distinct datasets in R^n. Their definition is provided as follows in [10]:

Definition 1: A function $g: R^n \rightarrow R$ is called polyhedral conic if its graph is a cone and all its sublevel sets $S_\alpha = \{x \in R^n: g(x) \leq \alpha\}$, for $\alpha \in R$, are polyhedrons.

Here, polyhedral conic function (pcf) $g_{(w,\xi,\gamma,a)}: R^n \rightarrow R$ is described as:

$$g_{(w,\xi,\gamma,a)}: R^n \rightarrow R = w'(x-a) + \xi \left\| x-a \right\|_1 - \gamma \tag{1}$$

where x is an n-dimensional vector, and

$x, w, a \in IR^n, \xi, \gamma \in IR, w'x = w_1x_1 + ... + w_nx_n,\ \left\|x\right\|_1 = \left|x_1\right| + ... + \left|x_n\right|$.

2.1. PCFs method

Gasimov and Öztürk applied polyhedral conic functions within a linear programming context to distinguish between two separate sets, introducing the initial PCFs algorithm in [10], then to decrease running time and to prevent over-fitting, this algorithm is improved in [11-13]. These algorithms are used in various classification researches [14 -17]. In this study, we used the one that allows misclassifications since our expectation from this algorithm is to detect the wrongly classified majority class elements. The used algorithm's pseudocode is given as follows [12]:

Consider two sets A and B, each comprising c and t, n-dimensional vectors, respectively: $A = \left\{a^i \in R^n, i \in I\right\}$, $B = \left\{b^j \in R^n, j \in J\right\}$ where $I = \{1,...,c\}, J = \{1,...,t\}$.

Algorithm 1: Binary classification via PCFs

Step 0. Perform a clustering algorithm on the set A. Let "s" denote the number of clusters, and set k=1. $I_k = I$.

Step 1. Let a_k denote the center of the k-th cluster. Solve the subproblem as follows:

$$(P_k) \quad \min \frac{1}{m}\sum_{i=1}^{m} y_i + C\frac{1}{p}\sum_{i=1}^{p} z_j \tag{2}$$

$$w(a^i - a_k) + \xi \left\| a^i - a_k \right\|_1 - \gamma + 1 \le y_i,\ i \in I_k \tag{3}$$

$$-w(b^j - a_k) - \xi \left\| b^j - a_k \right\|_1 + \gamma - 1 \le z_j,\ j \in J \tag{4}$$

$y_i,\ z_j \ge 0,\ C \ge 1,\ w \in R^n,\ \xi \in R,\ \gamma \ge 1$.

Let $w_k,\ \xi_k,\ \gamma_k,\ y_k$ be a solution of (P_k). Let $g_k(x) = g_{(w_k,\xi_k,\gamma_k,a_k)}(x)$.

Step 2. If $k < s$, let $k=k+1$, $I_k = \left\{ i \in I_{k-1} : g_{k-1}\left(a^i\right) > 0 \right\}$ and go to *Step 1*.

Step 3. Define the function *g(x)* (separating the sets A and B) as $g(x) = \min_k g_k(x)$ and **STOP**.

While combining this algorithm with the above proposed method, we call the A set as the majority class (number of data is larger than the other class elements' number), and the other set as B set. Here the defined number of clusters and the C penalty parameter used in the objective function is so important since they have a significant impact on the accuracy rate. In numerical experiments cluster number is defined as "2" and C penalty parameter is defined as "10". If the number of majority class elements is much larger than the minority class elements, since in this case it allows more mislabeling of the majority class elements, C penalty parameter can be decreased

3. A novel undersampling method for imbalanced binary data

A novel undersampling method for imbalanced binary data is suggested in this section. In real-world, most of binary datasets are imbalanced that the number of elements in one is greater than the other. Balancing a dataset helps to create more reliable and fair models by ensuring that all classes are adequately represented during the model training process. It helps to get better accuracy results and also decreases the running times of the implementations. As mentioned in the Introduction section, undersampling is one of the techniques to balance the dataset, it uses deleting the redundant majority ones approach. Based on this approach the suggested algorithm is presented below in a detailed pseudocode (Algorithm2).

Algorithm 2: Undersampling of an imbalanced binary dataset via PCFs algorithm

Step 0: Consider two sets A and B, each comprising *c* and *t*, where *c*>*t*, *n*-dimensional vectors, respectively:
, where $I = \{1,...,c\}, J = \{1,...,t\}$.
Define the minority and majority classes of the imbalanced dataset as follows:

Majority class: A=$\{a_1,a_2,...,a_c\}$ and Minority class: B=$\{b_1,b_2,...,b_t\}$.
Set the number of elements to be deleted as "*nd*"= *c-t*.

Step 1. Apply **Algorithm 1** (Binary classification via PCFs) that allows mislabelling and seperate minority from majority class. Define the separating function as *g()* function.

Step 2. Detect the mislabeled "a" majority points ($g(a_i)<=0$).
Let Mislabeled=$\{a_i, g(a_i)<=0\}$ and set "*k"*, as the number of elements in Mislabeled set.

Step 3. If $k<nd$,
Delete all the elements in Mislabeled from A set. Then call the new one as NewMajority.
NewMajority=$\{c_i, i=1,...,c-k\}$.
Let $nd=nd-k$ and C1=$\{g(c_i),c_i \in NewMajority\}$.
Order C1 set from max to min.
Delete first '*nd*' number of elements' subjected c_i values in C1 from NewMajority.

Step 4. If $k=nd$,
Delete all elements in Mislabeled from A set. Then call it as NewMajority.

Step 5. If $k>nd$,
C2=$\{g(a_i),a_i \in$ Mislabeled$\}$.
Order C2 set from min to max.
Delete first '*nd*' elements' subjected a_i values in C2 from A set. And call it as NewMajority.

Step 6. Define the balanced set (BS) by combining NewMajority and Minority (B) sets as
$BS = NewMajority \bigcup Minority$,
and **STOP**.

In below algorithm's initilization (*Step 0*), the dataset is parted as minority and majority classes and the number of elements to be deleted (*nd*) are defined.

In Step 1, the PCFs algorithm was applied to define the crucial part of the algorithm, which is the separator function (*g()*). Prior to this application, the main parameters of the algorithm1 (number of clusters and penalty parameter (*C*)) is determined.

In Step 2, a set called "Mislabelled" was defined for misclassified majority elements using the identified separator function. Then the number of the elements in this set is called as "*k*".

In Step 3 to 5, according to the difference between the number of elements in Mislabeled "*k*" and number of elements to be deleted "*nd*", three different operations are performed.

If "*k*" is less than "*nd*", it means that just deleting the mislabeled ones is not enough to get a balanced set so we need to find "*nd-k*" number of elements that will be added to the delenda list. For this aim in Step 3 firstly NewMajority set is defined by deleting the mislabeled ones from Majority class (A)..Then first "*nd-k*" elements that get the maximum value in the function *g()* is deleted from NewMajority. For the purpose of this operation on the algorithm, each value taken by the elements of NewMajority in the *g()*

function was calculated, and a set C1 was defined by sorting these values in descending order. The first "*nd-k*" indexed c_i elements of C1 subjected to the *g()* function are determined, and removed from the NewMajority set.

In Step 4, if "*k*" is equal to the "*nd*" then the whole Mislabeled set elements are deleted from Majority class (A) and new set is defined as NewMajority.

In Step 5, if "*k*" is less than "*nd*" then each value taken by the elements of Mislabeled in the *g()* function is calculated, and a set C2 is defined by sorting these values in ascending order. The "a_i" elements subjected to the *g()* function for the first "*nd*" elements are removed from the Majority set and the newly formed set is defined as NewMajority.

And finally in Step 6, the balanced dataset is defined as the combination of the NewMajority set and Minority set (B).

4. Experimental results

In the experimental results section, we applied the proposed method to imbalanced datasets. To assess the effectiveness of the approach in classification applications, both imbalanced and balanced datasets are implemented on state-of-art classification methods.

To define the effectiveness; accuracy, cross validation and running time performance metrics are used. Accuracy is a measure of the proximity of each result (x_i) obtained from the analytical method to the correct value (x_t) and the correct accepted value. Accuracy is expressed in terms of absolute error (E) or relative error (E_r) [18]:

$$E = x_i - x_t$$

$$E_r = \frac{x_i - x_t}{x_t} \times 100 .$$

k-fold cross-validation is one of the most commonly used methods in the literature. The basic idea of this method is to split the elements into *k* groups randomly [19,20]. In the experiments, "*k*" parameter of *k*-fold cross-validation is defined as 10. The dataset B is randomly divided into 10 heterogeneous equal sized subsets (folds). The algorithm undergoes training and testing 10 times.

Also we consider the real running time consumption, where the computer is Casper Nirvana Intel(R) Core(TM) i5-8250U and the used software programs are Matrix Laboratory (MATLAB) and Waikato Environment for Knowledge Analysis (WEKA) for imbalancing the dataset and classification operations respectively.

The datasets were sourced from the UC Irvine (UCI) Machine Learning Repository [21]. Below in Table 1, the features (imbalance level, number of instances, number of attributes, number of minority elements and number of majortiy elements) of the used imbalanced datasets are presented.

Table 1. Imbalanced dataset details.

	Hearth	Iris	Pima	Vehicle	Haberman	Ecoli 0-1	Breast Cancer
Imbalance Level	1.25	2	1.86	3.25	2.77	1.85	1.68
Number of instances	270	150	768	846	225	220	539
Number of attributes	13	4	8	18	3	7	30
Number of majority class elements	150	100	500	647	144	143	357
Number of minority class elements	120	50	268	199	81	77	212

These datasets are implemented on Naive Bayes, Classification via Regression (ClssfViaReg.), Logistics, and J48 (a decision tree algorithm) state-of art classification algorithms. The cross-validation, accuracy and running time performances of these algorithms on both imbalanced and balanced datasets are presented in Table 2 and 3 respectively.

Table 2. Cross validation(%), accuracy(%) and running time(sec.) results of the classification algorithms on imbalanced datasets.

Algorithms	Performance Metrics	Ecoli 01	BreastCancer	Hearth	Iris	Pima	Vehicle	Haberman
Naive Bayes	Accuracy	97.72	93.84	85.18	100	76.30	65.72	75.81
	CrossVld.	97.27	92.97	83.70	100	76.30	66.07	74.50
	RunningTime	0.01	0	0.01	0	0.01	0.02	0
ClssfViaReg	Accuracy	99.09	98.41	85.15	100	77.34	99.05	74.50
	CrossVld.	98.63	94.02	80	100	76.69	95.5	71.24
	RunningTime	0.1	0.11	0.27	0.09	0.33	0.26	0.17
Logistics	Accuracy	98.64	100	85.55	100	78.25	97.99	74.83
	CrossVld.	98.18	93.84	83.70	100	77.25	97.16	74.83
	RunningTime	0.1	0.11	0.27	0.02	0.05	0.08	0.01
J48	Accuracy	98.63	99.12	91.48	100	84.11	98.69	77.12
	CrossVld.	99.09	93.32	76.66	100	73.82	93.26	71.56
	RunningTime	0.01	0.07	0.04	0.01	0.05	0.08	0.02

Table 3. Cross validation(%), accuracy(%) and running time(sec.) results of the classification algorithms on balanced datasets via Algorithm 2.

Algorithms	Performance Metrics	Ecoli 01	BreastCancer	Hearth	Iris	Pima	Vehicle	Haberman
Naive Bayes	Accuracy	96.75	**98.58**	**87.5**	**100**	74.62	**98.74**	72.22
	CrossVld.	95.45	**98.11**	**85**	**100**	71.26	**98.74**	70.37
	RunningTime	**0**	**0**	**0.01**	**0**	**0.01**	**0.01**	**0**
ClssfViaReg	Accuracy	98.70	**99.29**	**86.66**	**100**	73.88	**99.74**	**86.41**
	CrossVld.	98.05	**98.31**	**82.5**	**100**	72.76	**98.99**	**79.01**
	RunningTime	**0.1**	**0.07**	**0.16**	**0.08**	**0.32**	**0.11**	**0.04**
Logistics	Accuracy	98.05	**100**	**88.33**	**100**	73.88	**100**	75.30

	CrossVld.	98.05	**95.99**	**85.41**	**100**	73.50	**99.49**	74.69
	RunningTime	**0.02**	**0.01**	**0.03**	**0.01**	**0.05**	**0.06**	**0**
J48	Accuracy	98.70	**99.82**	**93.75**	**100**	83.58	**99.74**	**86.41**
	CrossVld.	98.05	**94.32**	**77.08**	**100**	71.26	**99.24**	**82.71**
	RunningTime	**0.01**	**0.06**	**0.03**	**0.01**	**0.04**	**0.02**	**0.01**

The main objective of the suggested undersampling method is to balance imbalanced datasets by removing required number of unnecessary major elements. To demonstrate that the removed elements are indeed useless, classification algorithms are applied to each balanced and imbalanced datasets, and the performance results of these algorithms are compared. In Table 3, the better or equal results are given in bold. The results indicate that balanced datasets via the suggested undersampling method (Algorithm 2) get better accuracy and cross validation performance results in most of the cases. And also it is seen that in the other cases the differences are negligibly small. When examining the computational efficiency on the running time it is seen that due to the decreasing of the instances on the dataset, all balanced dataset implementations need less running time than the imbalanced dataset implementations. Based on these results, it can be concluded that the proposed algorithm is beneficial and useful for addressing imbalanced datasets in the context of machine learning algorithms.

5. Conclusion

In conclusion, this study introduces a novel undersampling method for addressing imbalanced datasets in machine learning applications. By leveraging a classification approach based on polyhedral conic functions, the proposed method effectively balances labeled binary datasets. Through extensive experimentation on both imbalanced and balanced datasets across various classification algorithms, it is evident that the suggested undersampling technique yields promising results. The balanced datasets do not only achieved an improved performance in terms of accuracy but also demonstrate enhanced computational efficiency. These findings underscore the utility and efficacy of the proposed algorithm in mitigating the challenges posed by imbalanced datasets in machine learning tasks. Moving forward, further exploration and validation of this method in diverse real-world applications could provide valuable insights into its broader applicability and effectiveness. Also for future studies, this method can be developped for multi-class datasets instead of binary.

References

[1] Ayoub, S., Gulzar, Y., Rustamov, J., Jabbari, A., Reegu, F.A. and Turaev, S., Adversarial Approaches to Tackle Imbalanced Data in Machine Learning, **Sustainability**, 15, 7097, (2023).

[2] Raghuwanshi, B.S. and Shukla, S., Class-Specific Extreme Learning Machine for Handling Binary Class Imbalance Problem, **Neural Networks**, 105, 206–217, (2018).

[3] Mohammed R., Rawashdeh J. and Abdullah M., Machine learning with oversampling and undersampling techniques: Overview study and experimental results, **2020 11th International Conference on Information and Communication Systems (ICICS)**, 243-248, Irbid, Jordan, (2020).

[4] Hoyos-Osorio J., Alvarez-Meza A., G. Daza-Santacoloma, Orozco-Gutierrez A. and Castellanos-Dominguez G., Relevant information undersampling to support imbalanced data classification, **Neurocomputing**, 436, 136-146, (2021).

[5] Sun Z., Song Q., Zhu X., Sun H., Xu B. and Zhou Y., A novel ensemble method for classifying imbalanced data, **Pattern Recognition**, 48, 5, 1623-1637, (2015).

[6] Barandela R., Valdovinos R.M. and Sánchez J.S., New applications of ensembles of classifiers, **Pattern Analysis & Applications**, 6, 3, 245-256, (2003).

[7] Seiffert C., Khoshgoftaar T.M., Van Hulse J. and Napolitano A., Rusboost: a hybrid approach to alleviating class imbalance, **IEEE Transactions on Systems, Man Cybernetics-Part A: Systems and Humans,** 40, 1, 185-197, (2010).

[8] Mostafaei, S.H. and Tanha, J., OUBoost: boosting based over and under sampling technique for handling imbalanced data. **International Journal of Machine Learning and Cybernetics,** 14, 3393–3411 (2023).

[9] Dai Q., Liu J. and Shi Y., Class-overlap undersampling based on Schur decomposition for Class-imbalance problems, **Expert Systems with Applications: An International Journal**, 221, C, 119735, (2023).

[10] Gasimov R.N., and Öztürk G., Separation via polyhedral conic functions. **Optimization Methods and Software,** 21, 4, 527-540, (2006).

[11] Uylas N., Methods based on mathematical optimization for data classification. PhD, Ege University, İzmir, Turkey, (2013).

[12] Uylas Sati N., A binary classification algorithm based on polyhedral conic functions, **Düzce University Journal of Science and Technology**, 3, 152-161, (2015).

[13] Öztürk G., and Çitfçi M., Clustering based polyhedral conic functions algorithm in classification, **Journal of Industrial and Management Optimization**, 11, 3, 921-932, (2015).

[14] Uylas Sati N. and Ordin B., Application of the polyhedral conic functions method in the text classification and comparative analysis, **Scientific Programming**, vol. 2018, Article ID 5349284, 11 pages, (2018).

[15] Acar M. and Kasimbeyli R., A polyhedral conic functions based classification method for noisy data, **Journal of Industrial and Management Optimization**, 17, 6, 3493-3508, (2021).

[16] Çevikalp H. and Saglamlar H., Polyhedral conic classifiers for computer vision applications and open set recognition, **IEEE Transactions on pattern analysis and machine intelligence**, 43, 2, 608-622, (2021).

[17] Cevikalp H., Uzun B., Köpüklü O. and Ozturk G., Deep compact polyhedral conic classifier for open and closed set recognition, **Pattern Recognition**, 119, 108080, ISSN 0031-3203, (2021).

[18] Skoog D.A., West D.M., Holler F.J. and Crouch S.R., **Fundamentals of Analytical Chemistry**, Nelson Education, (2013).

[19] Szeghalmy, S. and Fazekas, A., A Comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning, **Sensors,** 23, 4, 2333, (2023).

[20] Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A. and Lawrence, N.D., **Dataset shift in machine learning**, MIT Press: Cambridge, MA, USA, (2022).

[21] Dua D, and Graff C., UCI Machine Learning Repository 2019. https://archive.ics.uci.edu/, (12.02.2024).