Bidirectional sampling method for imbalanced data

Junjie Shi[a,1], Deyu Song[b,2], Shengyao Zheng[a,3], Yueming Hu[a,4], Shuangshuang Chen[a,5], Fengque Pei[a,*]

[a] College of Mechanical and Electrical Engineering, Hohai University, #200 Jinling north road, Changzhou, Jiangsu China 213000; [b] College of Internet of Things, Hohai University, #200 Jinling north road, Changzhou, Jiangsu China 213000

ABSTRACT

Traditional over-sampling and under-sampling algorithms suffer from overfitting and high noise when unbalanced data classes are in the sample set. To improve the performance of the data classifier, this study proposes a SMOTECU algorithm combining SMOTE and ClusterCentroids under-sampling. It absorbs the advantages of both algorithms and avoids generating or rejecting excessive samples in the dataset, effectively reducing the harmful effects of overfitting and noise. We experiment with 16 unbalanced standard datasets combining three classifiers: RF, RBFNN, and RBFSVM. By comparing three evaluation metrics: F1-score, AUC, and running time, the results demonstrate that the performance of the SMOTECU-based random forest model is better, and compared with SMOTE and ClusterCentroids, SMOTECU can effectively avoid overfitting and save running time.

Keywords: Unbalanced data set; over-sampling; under-sampling; machine learning

1. INTRODUCTION

Classification learning is one of the important research directions of machine learning. However, in actual production lines and detection, there will be problems of data set imbalance. The classifier constructed according to the imbalanced data set will make the prediction result more biased towards the majority class, while the minority class samples are often important research objects. Therefore, reducing and eliminating this kind of imbalance problem has important significance.

In current related research, the main way to solve the data imbalance problem at the data level is to oversample the minority class samples and undersample the majority class samples. The most widely used oversampling algorithm is SMOTE algorithm proposed by Chawla et al.[1], which can effectively oversample the minority class samples and make samples balanced, but this algorithm has some blindness in neighbor selection. Hui H et al.[2] proposed an improved algorithm of SMOTE, the Borderline-SMOTE algorithm. This algorithm only uses minority class samples on the boundary to synthesize new samples, effectively improving the possible problem of high repetition in SMOTE, but there is a situation in that boundary samples are difficult to identify. Bao et al.[3] proposed two new SMOTE algorithms, CP-SMOTE and IO-SMOTE. CP-SMOTE algorithm generates new samples by clustering to obtain center points and linearly combines minority class samples with center points. IO-SMOTE algorithm divides samples into internal samples and external samples so that more internal samples can be used in the process of generating new samples. These two algorithms make the samples away from the classification boundary and obtain better classification performance.

*Email:tyf51129@aliyun.com;[1]1649841595@qq.com; [2]2061410112@hhu.edu.cn; [3]1245292420@qq.com

[4]2106329803@qq.com; [5]1085824318@qq.com

Seventh International Conference on Mechatronics and Intelligent Robotics (ICMIR 2023), edited by Srikanta Patnaik, Tao Shen, Proc. of SPIE Vol. 12779, 127792H · © 2023 SPIE · 0277-786X · Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2688900

For undersampling methods, He Yunbin et al.[4] proposed a weighted boundary point ensemble undersampling algorithm based on clustering, which effectively improved the execution efficiency of the algorithm and the accuracy of classification results. However, in some data ratio ranges, there will be a large loss of original data distribution information. Zhou Qian et al.[5] proposed a distance-weighted undersampling algorithm based on adaptive k-means clustering. They used the k-means clustering method to cluster majority class samples, eliminate outliers, sort data, and sample majority class samples in order, effectively improve the impact of unbalanced data on classification accuracy. Still, this algorithm has great limitations for multi-classification problems. Wang Lei et al.[6] proposed a cluster undersampling weighted random forest algorithm CUS-WRF. They used undersampling associated with clustering on the data side and weighted random forest algorithm on the algorithm side to get better classification results, But in the future, certain research is still needed in terms of time complexity and boundary sample recognition. Cui Caixia et al.[7] proposed an adaptive undersampling method based on density peak clustering. They considered overlapping areas, noise, inter-class and intra-class imbalance sample sparsity degree and proposed solutions to Improve the accuracy of classification results for minority class samples, but this method is not suitable for multi-class imbalance problems.

This research proposes a SMOTECU (Synthetic Minority Over-sampling Technique Cluster Undersampling) algorithm combining undersampling and oversampling. Firstly ClusterCentroids undersampling is performed on majority classes with average sample number as target reducing majority class number and retaining feature information. Then SMOTE oversampling is performed on minority classes reducing required synthetic minority classes thus reducing model's computational complexity and noise interference finally obtaining balanced data sets.

2. ALGORITHM INTRODUCTION

2.1 Clustercentroids algorithm introduction

The ClusterCentroids algorithm is an under-sampling method that synthesizes the majority class samples by dividing them into *K* clusters using the k-means++ algorithm and replacing them with the center points of these *K* clusters, thereby shrinking the number of majority class samples to *K*. ClusterCentroids algorithm can reduce the number of samples very efficiently. Still, when the data imbalance rate is high, the number of cluster centers is too small, and there is a high chance of losing critical information, resulting in overfitting.

2.2 SMOTE algorithm introduction

SMOTE is a common sampling method that solves the problem of sample imbalance by creating synthetic samples from the minority class. It does this by interpolating between the nearest neighbors of the minority class samples to increase the number of samples and balance the sample quantity. When the data imbalance ratio is large, SMOTE may over-sample too much data, resulting in high computation cost, low information gain, and noise amplification. This is because the new data generated by SMOTE may have a high degree of overlap with the existing data

2.3 SMOTECU algorithm

To overcome the limitations of these two algorithms, this study proposes a SMOTECU algorithm that combines them: first, ClusterCentroids is used to under-sample the majority class by replacing the original data with the cluster cores after clustering, which reduces their number while preserving the feature information of the sample set; then, SMOTE is used to oversample the minority class by synthesizing new samples between neighboring ones, which increases their number. Finally, sample quantity balance is achieved, as shown in Figure 1.

Algorithm steps:

1) Divide the minority class sample set $S_{minority}$ and the majority class sample set S_{most} with the average value $m = round(\frac{N_{max} + N_{min}}{2})$ of the maximum N_{max} and minimum N_{min} samples of all classes as the boundary;
2) Set the average value m as the target sample number of the minority class sample and majority class sample;
3) Use the k-means++ algorithm to cluster the majority class sample set S_{most} into m clusters: $C=(C_1, C_2,\ldots,C_m)$;
4) Keep the cluster cores $c=(c_1, c_2,\ldots, c_m)$, remove other data, and get the adjusted majority class sample set $NewS_{most}$;
5) For the minority class samples $S_{minority}$, find the K nearest neighbors of each sample point according to the Euclidean distance (to prevent the sample point and the nearest neighbor line from passing through the majority class sample space, K should not be too large, this study takes K=10);

6) Randomly draw t minority class samples, set the sampling rate n_1 according to the distance between the sample number $NumS_{minority}$ and the target number m, and randomly draw n_1 times in the K nearest neighbors of each sample. Set the sampling rate of the remaining $NumS_{minority}$ - t minority class samples to n_2, and randomly draw n_2 times in the K nearest neighbors of each sample;

$$n_1 = ceil(\frac{m - NumS_{\min ority}}{NumS_{\min ority}}) \tag{1}$$

$$n_2 = floor(\frac{m - NumS_{\min ority}}{NumS_{\min ority}}) \tag{2}$$

$$w = \frac{m - NumS_{\min ority}}{NumS_{\min ority}} - floor\,(\frac{m - NumS_{\min ority}}{NumS_{\min ority}}) \tag{3}$$

$$t = round(NumS_{\min ority} \times w) \tag{4}$$

7) generate a new sample x_{new} randomly on the line between the minority class sample point x and the nearest neighbor x_n drawn each time;
8) Add the generated sample points to the original sample set to obtain the adjusted minority class sample set $NewS_{minority}$;
9) Balanced sample set $NewS=\{NewS_{most}, NewS_{minority}\}$.

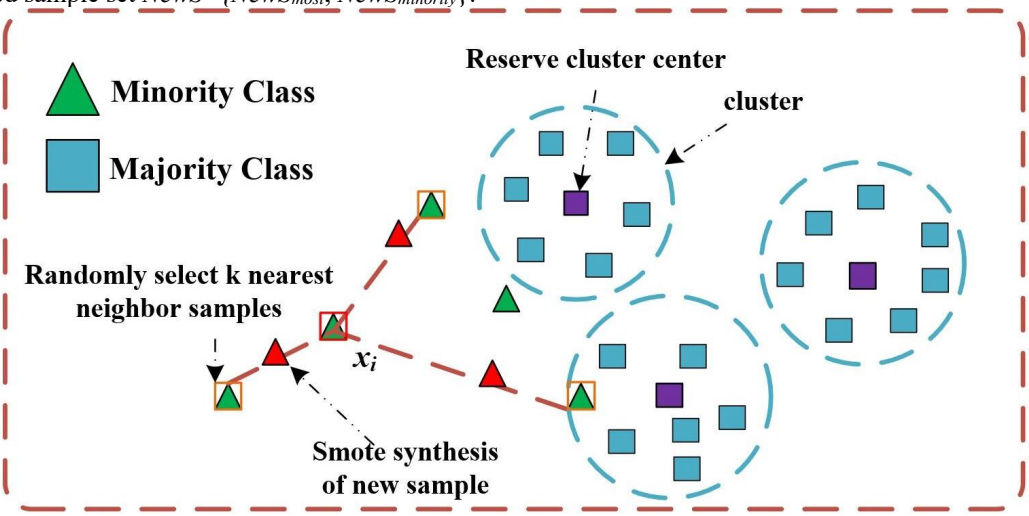

Figure 1. SMOTECU Algorithm

The algorithm combines the ideas of ClusterCentroids under-sampling and SMOTE oversampling, using the advantages of these two algorithms to efficiently reduce or increase the number of samples to adjust the sample size. At the same time, it uses the average value of majority class and minority class as the target number, avoiding generating or eliminating too many samples. Compared with ClusterCentroids under-sampling, SMOTECU sets more clustering centers, which can retain more features and reduce the risk of overfitting. Compared with SMOTE over-sampling, SMOTECU reduces the number of samples that need to be synthesized by the minority class, shortens the calculation time of the model, and avoids too dense sample points of minority class, thereby reducing the risk of generating meaningless data and noise data.

3. EXPERIMENTAL RESULTS AND ANALYSIS

3.1 Dataset introduction

To verify the effectiveness of SMOTECU algorithm, this study collected 16 standard datasets with imbalanced samples. Table 1 lists the information and the unbalance rate ubrate of these datasets.

Table 1. Imbalanced dataset information

Data set	Features	Samples	ubrate	Data set	Features	Samples	ubrate
car_eval_34	21	1728	11.90%	us_crime	100	1994	12.29%
abalone	10	4177	9.68%	spectrometer	93	531	10.80%
arrhythmia	278	452	17.08%	thyroid_sick	52	3772	15.33%
sick_euthyroid	42	3163	9.80%	mammography	6	11183	42.01%
satimage	36	6435	9.28%	oil	49	937	21.85%
scene	294	2407	12.60%	optical_digits	64	5620	9.14%
solar_flare_m0	32	1389	19.43%	ozone_level	72	2536	33.74%
wine_quality	11	4898	25.77%	pen_digits	16	10992	9.42%

3.2 Performance comparison of different algorithms

For imbalanced datasets, the classification results tend to be biased towards the majority class. Therefore, relying solely on accuracy to evaluate classification performance is one-sided and cannot accurately measure the generalization ability of the classification model. In this study, the standard metrics for classification problems, AUC and F1-score were used to evaluate the classification performance of the classifier. We calculate the AUC[8] and F1-score values by the confusion matrix of the classification results, and the closer the values are to 1, the better the classification performance.

Table 2. Confusion Matrix

Confusion Matrix		Predict	
		1(Positive)	**0(Negative)**
Actual	**1(Positive)**	TP(True Positive)	FN(Fales Negative)
	0(Negative)	FP(Fales Positive)	TN(True Negative)

$$TPR = \frac{TP}{TP + FN} \tag{5}$$

$$FPR = \frac{FP}{TN + FP} \tag{6}$$

$$F1 - score = \frac{2TP}{2TP + FN + FP} \tag{7}$$

$$AUC = \frac{1 + TPR - FPR}{2} \tag{8}$$

Then, this research uses Random Forest, RBF neural network (RBFNN), and support vector machine based on RBF (RBFSVM) to compare the classification effects of 16 datasets processed by SMOTE, ClusterCentroid,s and SMOTECU algorithms. Divide the training and testing into a 7:3 ratio and repeat ten times . The average values of the F1-score and AUC are shown in Table 3.

Table 3. Comparison of F1-score and AUC values for partial datasets

Sequence of Datasets	Algorithm	SMOTE		ClusterCentroids		SMOTECU	
		F1	AUC	F1	AUC	F1	AUC
car_eval_34	RF	0.9952	0.9944	1.0000	1.0000	1.0000	1.0000
	RBFNN	0.9833	0.9816	1.0000	1.0000	0.9949	0.9939
	RBFSVM	0.9757	0.9749	0.9877	0.9875	0.9842	0.9826
sick_euthyroid	RF	0.9811	0.9811	0.9560	0.9545	0.9773	0.9762
	RBFNN	0.9352	0.9291	0.8229	0.8242	0.8713	0.8473
	RBFSVM	0.8199	0.7898	0.8400	0.8182	0.8287	0.7809
satimage	RF	0.9769	0.9756	0.8947	0.8939	0.9627	0.9574
	RBFNN	0.9188	0.9075	0.7365	0.7516	0.8802	0.8507
	RBFSVM	0.8663	0.8414	0.8125	0.7766	0.8557	0.8076
scene	RF	0.9309	0.9275	0.8545	0.8501	0.9401	0.9397
	RBFNN	0.9028	0.8832	0.7321	0.7254	0.9171	0.9101
	RBFSVM	0.8049	0.7801	0.7769	0.7453	0.8299	0.8088
wine_quality	RF	0.9677	0.9668	0.9434	0.9457	0.9465	0.9454
	RBFNN	0.8873	0.8869	0.6512	0.7166	0.8745	0.8676
	RBFSVM	0.7414	0.7549	0.4722	0.6545	0.7387	0.7470
mammography	RF	0.9770	0.9768	0.8903	0.8911	0.9733	0.9724
	RBFNN	0.9223	0.9239	0.8690	0.8790	0.9170	0.9164
	RBFSVM	0.9034	0.9050	0.7630	0.7372	0.8746	0.8800
oil	RF	0.9908	0.9908	0.9091	0.9161	0.9735	0.9717
	RBFNN	0.9674	0.9630	0.6667	0.7222	0.9610	0.9492
	RBFSVM	0.9137	0.9110	0.7333	0.6667	0.8797	0.8668
ozone_level	RF	0.9807	0.9804	0.9756	0.9762	0.9746	0.9742
	RBFNN	0.8993	0.8797	0.8889	0.8831	0.8981	0.8835
	RBFSVM	0.8768	0.8704	0.8444	0.8409	0.8662	0.8570

According to the test results of the datasets in Table 3, Random Forest has the best classification performance, followed by RBF Neural Network, and RBFSVM performs the worst. Regarding runtime, RBF Neural Network takes much longer than Random Forest and RBFSVM.

For the test results, the classification performance of the SMOTECU-based models on the car_eval_34 and the scene datasets is better than that of SMOTE and ClusterCentroids. In the other test results, the classification results based on SMOTECU are slightly worse than SMOTE. In the random forest classifier, the SMOTECU algorithm can reduce the computational complexity and maintain high classification performance compared with SMOTE oversampling, and can effectively avoid the overfitting phenomenon (100% accuracy and less than 2 seconds running time) compared with ClusterCentroids. Moreover, the RBF Neural Network classification model based on SMOTECU can significantly reduce the runtime while maintaining high accuracy.

3.3 Analysis of dataset feature space

To further investigate the applicability of the SMOTECU algorithm, we use two dimensionality reduction algorithms, t-SNE[9] and UMAP[10], to reduce the high-dimensional feature space of these 16 datasets to two dimensions. Then the classification performance was compared to further analyze the results. The dimensionality reduction diagram of some datasets is shown in Figure 2.

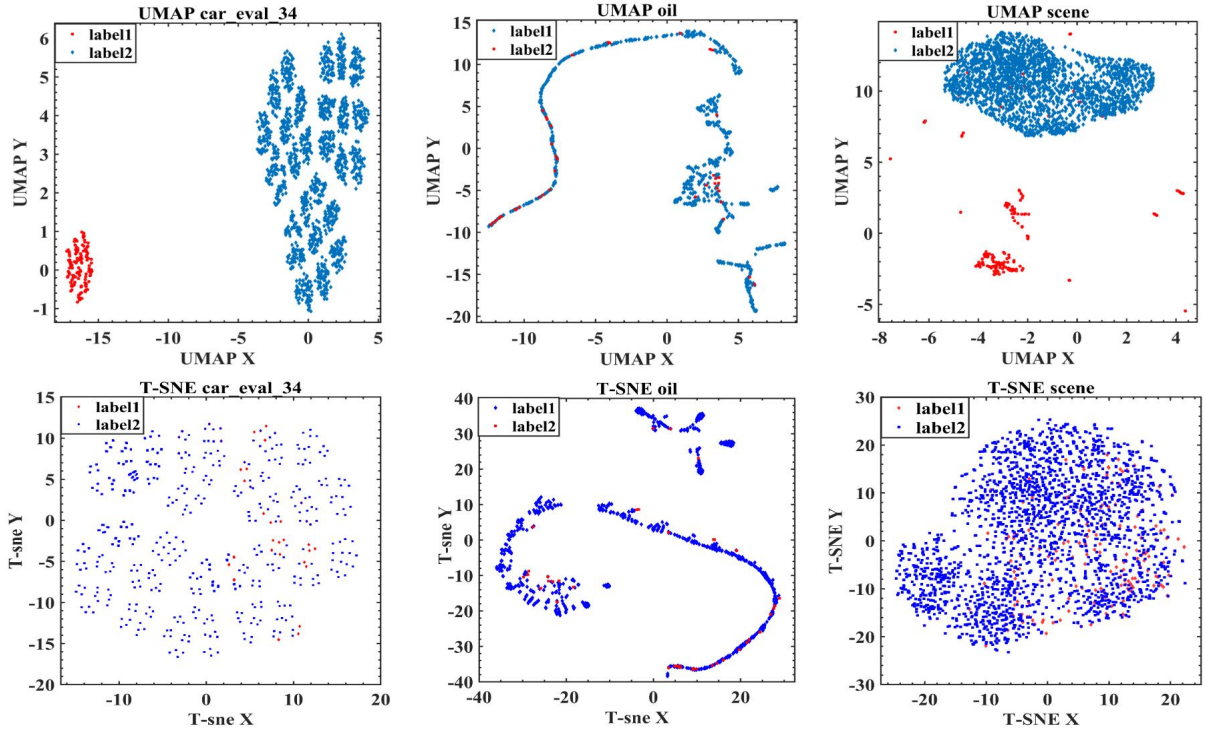

Figure 2.T-SNE and UMAP dimensionality reduction for some of the datasets

Through comparing the feature space dimensionality reduction of the tested data set, it was found that SMOTECU is good at dealing with data sets where sample points are more dispersed on the t-SNE dimensionality reduction map, and the minority and majority class are clustered separately with high distinguishability on the UMAP dimensionality reduction map. However, the classification performance based on SMOTECU is significantly worse than that of the SMOTE oversampled data set, where the sample points are linearly distributed on the t-SNE dimensionality reduction map, and the sample points of different labels are more chaotic and not clearly distinguished on the UMAP dimensionality reduction map.

4. CONCLUSION

This study addresses the problem of imbalanced data classification and proposes the SMOTRECU algorithm from the perspective of data. The algorithm combines over-sampling and under-sampling methods to achieve data balancing. Firstly, the majority class samples are clustered by the k-means++ clustering method and replaced with cluster centroids, reducing the number of samples while retaining the main features of the data. Then, SMOTE over-sampling is applied to minority samples, reducing the number of generated sample points and mitigating the negative impact of traditional SMOTE over-sampling on excessive sample generation and noise amplification. By adjusting the number of minority and majority samples simultaneously, the algorithm makes the dataset structure more reasonable and effectively reduces the risk of overfitting. The algorithm has been tested on standard datasets, demonstrating good classification performance on highly discriminative data and random forest classification models and saving runtime for RBF neural networks. However, the advantages of the SMOTRECU algorithm are insignificant for imbalanced datasets with low discriminability, and further research is needed in this regard.

REFERENCES

[1] Chawla, N. V., Bowyer, K. W., Hall, L. O., et al., SMOTE: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, 16, 321-357 (2002).

[2] Han, H., Wang, W. Y., Mao, B. H., Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1 (pp. 878-887). Springer Berlin Heidelberg.(2005).

[3] Bao, Y., Yang, S. B., Two Novel SMOTE Methods for Solving Imbalanced Classification Problems, IEEE Access, 11, 5816-5823 (2023).

[4] He, Y., Leng, X., Wan, J., Unbalanced data weighted boundary point integration undersampling method, Journal of Xidian University, 48(4), 176 (2021).

[5] Zhou, Q., Yao, Z., B. Sun, B., Under-sampling Algorithm withWeighted Distance Based on Adaptive K-Means Clustering, Data Analysis and Knowledge Discovery, 6(5), 127-136 (2022).

[6] Wang, L., Liu, Y., Liu, Z., et al., Clustering under-sampling weighted random forest algorithm for processing unbalanced data, Application Research of Computers, 38(5), 1398-1402 (2021).

[7] Cui, C., Cao, F., Liang, J., Adaptive Undersampling Based on Density Peak Clustering, Pattern Recognition and Artificial Intelligence, 33(9), 811-819 (2020).

[8] Shen, Z., Hua, X., Jinhai, C., Resampling Algorithms for Imbalanced Data Author, Journal of Small and Microcomputer Systems, 1-8 (2023).

[9] Wei, V., Ivkin, N., Braverman, V., *et al.*, Sketch and Scale Geo-distributed tSNE and UMAP, IEEE International Conference on Big Data. 996-1003 (2020).

[10] Sainburg, T., McInnes, L., Gentner, T. Q., Parametric UMAP Embeddings for Representation and Semisupervised Learning, Neural Computation, 33(11), 2881-2907 (2021).