Augmentation Based Synthetic Sampling and Ensemble Techniques for Imbalanced Data Classification

Wakjira Mulugeta Asefaw, Ronghua Shang, Michael Aggrey Okoth, Licheng Jiao

Augmentation Based Synthetic Sampling and Ensemble Techniques for Imbalanced Data Classification

Wakjira Mulugeta Asefaw, Ronghua Shang, Michael Aggrey Okoth, and Licheng Jiao

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi Province 710071, China.
wakjira@stu.xidian.edu.cn; rhshang@mail.xidian.edu.cn;
mokoth83@yahoo.com; lchjiao@mail.xidian.edu.cn

Abstract. The imbalance data problem appears in data mining fields and has recently attracted the attention of researchers. In order to solve this problem, scholars proposed various approaches such as undersampling majority class, oversampling minority class, synthetic Minority Oversampling (SMOTE) technique, Proximity Weighted Random Affine Shadowsampling (ProWRAS), etc. However, this work proposes a new method called Augmentation Based Synthetic Sampling (ABS) for imbalanced data classification that concatenates data to predict features with imbalance problems. The proposed study integrates sampling and concatenated features to generate synthetic data. This study shows the ability of the proposed method and the average of the AUC (area under the curve) to generate good data samples while experimenting compared to the previous study. In addition, this study merged the proposed method with the boosting to create a technique known as ABSBoost. Therefore, the experimental outcomes show that the proposed ABS method and ABSBoost are effective on the given datasets.

Keywords: Data augmentation· Imbalanced data· Concatenated data· Oversampling· Undersampling.

1 Introduction

In the field of data mining, imbalanced data are a common problem when the data is made up of minority and majority classes [16]. It is challenging to balance between the majority and minority classes. To overcome the imbalanced data, many researchers used different techniques such as Synthetic Minority Oversampling (SMOTE) technique [3], Proximity Weighted Random Affine Shadowsampling (ProWRAS) [1], Evidential Combination of Classifiers (ECC) [10], Deep Density Hybrid Sampling (DDHS) [8], Distributed SMOTE (D SMOTE) and Modified Biogeography-Based Optimization (M BBO) [6], Spatial Distribution-based UnderSampling (SDUS) [14], etc.

The existing classification methods mainly focus on the majority class accuracy and classification of sample categories into the majority class, but there is no fair division of minority class [15]. Synthetic Minority Oversampling (SMOTE) is a technique that works and relies on a random oversampling algorithm and adds new sample values between minority samples of data of the neighbours to produce new minority samples and insert them into the dataset [3,4,15]. Data augmentation helps to enlarge the training

of datasets and increase the performance accuracy of the model [2]. The Formula that generates synthetic data samples:

$$y_{new} = y_i + (y_i{}' - y_i) * z \tag{1}$$

where y_i is the minority class of data samples, $y_i^{'}$ is the chosen neighbor for y_i, and z is a random number distributed uniformly from 0 until 1. According to SMOTE formula in equation (1), y_{new} relies on the random number z, which identifies the location of synthetic data in linear interpolation between y_i and $y_i^{'}$. Given that z is near 0, the synthetic data samples will be near y_i. In contrast, the synthetic sample will be near $y_i^{'}$ as z approaches 1.

Today, many organizations have large amounts of data, but the data is not balanced. However, we propose an augmentation based synthetic sampling method to overcome this challenge. This proposed method works by concatenating features with other features. This paper achieves state-of-the-art performance for imbalanced data on several public large-scale datasets. Experiments also indicate that the proposed method can be easily integrated into various backbones with significant performance improvements. The main structure of this work is organized as follows. Section 2 introduces the principles and execution process of the proposed method in detail. In Section 3, an explanation of the experimental setting and the experimental result is displayed and analyzed. In Section 4, draft the Conclusions.

2 Augmentation Based Synthetic Sampling Method

This part introduces the proposed method, including Data Augmentation (DA), notation, and proposed method.

2.1 Data Augmentation (DA)

Data Augmentation (DA) is a process of enlarging the feature size [5,12]. The extra features are generated synthetically by applying simple transformations to existing data. Its purpose may differ based on the particular use case and challenge. However, according to the existing applications, DA has two main benefits [7]: On one hand, it improves the generalization ability of a model by adding a piece of useful information to the training data, and on the other hand, it enhances the robustness of a model against the input perturbations. Data augmentation (DA) can enlarge the training features of the input dataset.

2.2 Notations

In this part, the notations used in the following part are introduced. Each data object $y^{(a)}$ is represented as a feature vector of length p i.e., $y^{(a)}$ =[$y_1^{(a)}$, $y_2^{(a)}$,..., $y_p^{(a)}$]. y_{-b} is introduced to represent all the features of data y except feature b. The data collection is represented by $M = [y_1^{(1)}, y_2^{(2)}, ..., y_p^{(q)}]$, indicating that the number of data objects is q and $y^{(a)} \in \mathbb{R}^p$. To simplify the explanation, we focus on binary classification, but

the proposed work could be expanded to multi-class problems. The focus of this study is imbalanced data; therefore, data could be separated into two classes, namely, the minority class and the majority class. We use M^c to denote the minority class samples and use M^d to represent majority class samples. Thus, the entire data set can be divided into two divisions, namely, $M = M^c \cup \mathrm{M}^d$. Using the sampling feature, we generate temporary synthetic data, so we further introduce the value set $W = [w_1, w_2, ..., w_p]$ for all the possible feature values of the minority class, so that $\forall y^{(a)} \in M^c$, $y_1^{(a)} \in w_1$, $y_2^{(a)} \in w_2$, ..., $y_p^{(a)} \in w_p$.

2.3 Proposed Method

The proposed method aims to concatenate data to generate synthetic data and balance data as original features. In this study, the proposed method involves three parts. The first part concatenates input data with sample feature and training feature models. The second part selects the sample feature to generate temporary sampling data randomly. The third part is concatenated minority data with temporary data to generate final synthetic data. The previous study used to generate synthetic data by training features without concatenated input data with trained features. The main objective of this proposed algorithm is to concatenate features with other features and keep the originality of the given data. [9]. Therefore, concatenated data is helpful to enhance and enlarge the minority class.

Concatenate input data with sample feature and training feature models. In order to concatenate the input data with trained features and find the relationship between the features, we proposed an augmentation-based synthetic sampling method. This work concatenates various features so that features characterize each data sample.

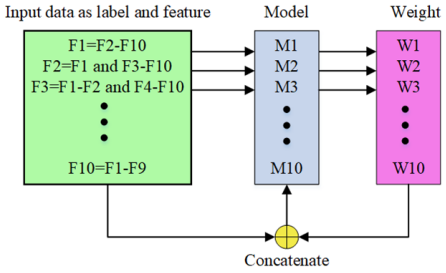

Fig. 1: The list of feature weights.

In the given data sample $y^{(a)}$, we use one of the features, say ${y_b}^{(a)}$, as a label, but the other remaining is as a feature, $y_{-b}^{(a)}$, train using model b after training the feature we concatenate both the first trained feature y_{-b} as a feature, y_b as a label with an output of trained feature.

As shown in Figure 1, first, we calculate $F1$ as a label and the remaining $F2 - F10$ as a feature using model $M1$ to obtain the trained feature as weight. Next, concatenate the weight $W1$ with the feature $F2 - F10$, then train using model $M1$ to gain the final trained feature as weight. The main target of this step is to create a relationship with various features and augment the feature to increase the performance of the algorithms.

Randomly select the sample feature to generate temporary sampling data. To generate the initial synthetic data, we used a sampling technique with a replacement (1/7) formula that calculates the number of sample possibility values from the given feature domains $W = [w_1, w_2, ..., w_p]$ for all features in the minority class from the training samples data. Given the value set W_1 = $[w_{11}, w_{12}, ..., w_{1q}]$, the first step is to sample a value from the set $w1$ with replacement, and the sampled one is the value of the first feature, namely ${y_1}^{(a)}$, for the temporary synthetic data sample $y^{(a)}$. The second step is to implement the sampling process to get the next sample feature value for the temporary synthetic data value until the bth sample feature value.

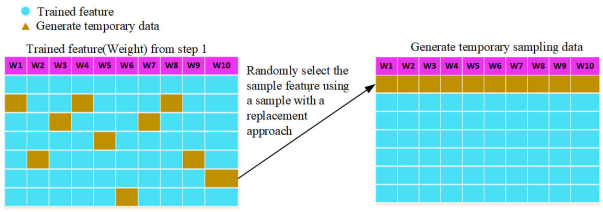

Fig. 2: Model of the sampling feature.

Continuing from step 1, using the final trained feature as weight, use a sample with a replacement approach to generate temporary sample data, as shown in Figure 2.

Concatenate minority data with temporary data to generate final synthetic data. In order to predict synthetic data, we concatenate the minority data with temporary data using model b. This last step goals to make the synthetic data reproduce the feature relationships with real observations. Given an initial synthetic data sample $y^{(a)}$ obtained from the previous step, one can predict y_b, the feature of the final synthetic data sample, with the trained model b and the input y_{-b}. $F1 - F10$ are represented as minority data, as shown in figure 3. Continuing from step 2, concatenate generated temporary sampling data $W1 - W10$ with minority data $F1 - F10$, as shown in Figure 3, to predict final synthetic data $S1 - S10$. As proposed algorithm 1 indicates, the input data include the minority data M^c, over-sampling rate R, the feature sample value set W, and the number of iterations for repeating the generation process L. In the first step, the temporary sampling data set T, and the synthetic data set Z are initialized as empty matrices of size $p * q^{sy}$ as shown in Algorithm 1. In the next step, we train p feature models for

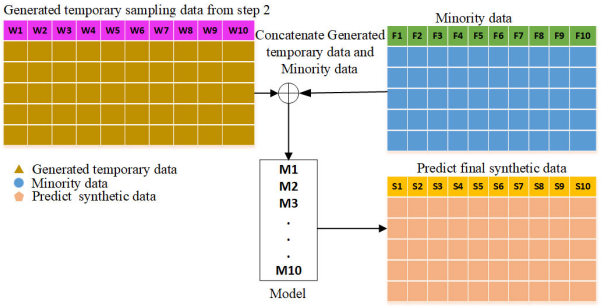

Fig. 3: Model of the synthetic prediction feature.

Algorithm 1 Augmentation Based Synthetic Sampling for Imbalanced Data Classification.

Input: M^c: minority data, R: Oversampling rate. $W = [w_1, w_2, \cdots, w_p]$: possible feature values of minority dataset, L: total iteration for repeating generation process.
Output: Z Generate synthetic data.
$q^c \leftarrow$ The number of minority data samples
$p \leftarrow$ The number of features
$q^{sy} \leftarrow q^c * R$ (The size of a synthetic data sample)
$T \leftarrow \emptyset$ (temporary data with p columns and q^{sy} rows)
$Z \leftarrow \emptyset$ (synthetic data with p columns and q^{sy} rows)
for $b \leftarrow 1$ *to* p **do**
  Train model b with y_b as label, concatenate y_{-b} as a feature with y_b as label
end
for $a \leftarrow 1$ *to* q^{sy} **do**
  for $b \leftarrow 1$ *to* p **do**
    $T_{ab} \leftarrow$ randomly sample a value from w_b with replacement
  end
end
for $l \leftarrow 1$ *to* L **do**
  for $b \leftarrow 1$ *to* p **do**
    $Z_b \leftarrow$ predict feature b by model b and concatenate T with minority data.
    $T = Z$ (update the temporary dataset for predicting by predicted dataset Z)
  end
end
return Z.

the p features, in which model b $(1 \leq b \leq p)$ is trained with y_b as the label concatenates y_{-b} as the features, y_b as the label with train feature. The second step, generate temporary data randomly using a sample with a replacement technique on the features using the proposed method. The final step, generate synthetic data samples, which Z_b denotes the bth feature of the synthetic data sample and is obtained from the prediction of the model$b$ by concatenating temporary data T with minority data as the input.

3 Experiment settings and result analysis

3.1 Datasets

In order to evaluate the performance of the proposed method and other existing methods, thirteen datasets were used in the experiments conducted. All of the data used in

the experiments are publicly available datasets from Kaggle and UCI machine learning repository:https://www.kaggle.com/ and https://archive.ics.uci.edu/ml, which are used the same dataset in [9].

3.2 Evaluation Metric

Classification is the task that calculates the probability's performance and accuracy. However, the restriction performance measure of imbalanced dataset accuracy had built immediately, and receiver operating characteristic (ROC) curves soon appeared as a well-known possibility, with the true positive rate (TPR) is on the y-axis and the false positive rate (FPR) is the x-axis. Therefore, as an alternative method of evaluating the accuracy performance of the classifier, we use the area under the curve (AUC) as the evaluation metric.

3.3 Experimental Results

We apply the proposed method to the twelve datasets and use AUC as the performance metric. We have a baseline method called original in the experiments, which trains the model using original data without a sampling method.

Table 1: Average of AUC results for proposed and different existing methods.

Dataset	Original	Over	Under	CBO	SBC	Smote	ADASYN	ProWRAS	MBS	**ABS**
Pima	0.8329	0.8320	0.8308	0.8125	0.8257	0.8323	0.8311	0.5979	0.8322	**0.8734**
Haberman	0.6699	0.6680	0.6685	0.5512	0.6514	0.6671	0.6657	0.7396	0.6791	**0.7549**
Satimage	0.7638	0.7627	0.7626	0.7446	0.7438	0.7639	0.7543	0.5797	0.7663	**0.8265**
Ecoli	0.9066	0.9042	0.9022	0.8880	0.8886	0.9074	0.9039	0.6256	0.9178	**0.9678**
Ionosphere	0.8270	0.8145	0.7847	0.8005	0.7799	0.8371	0.8309	0.8063	0.8275	**0.9759**
Vehicle	0.9829	0.9828	0.9728	0.9850	0.9663	0.9900	0.9931	0.8297	0.9838	**0.9989**
Credit	0.6993	0.7010	0.7004	0.6968	0.7072	0.7001	0.7008	0.6267	0.7173	**0.7343**
Diabetes	0.6302	0.6303	0.6304	0.5455	0.6017	0.6293	0.6280	0.5019	0.6317	**0.6990**
Hmeq	0.7909	0.7914	0.7916	0.6754	0.7818	0.7911	0.7907	0.7393	0.7913	**0.8563**
Promotion	0.6481	0.6483	0.6480	0.6444	0.6423	0.6485	0.6488	0.6245	0.6486	**0.6885**
Bank	0.8577	0.8593	0.8590	0.7592	0.8425	0.8591	0.8606	0.5187	0.8608	**0.9052**
Spambase	0.9683	0.9681	0.9653	0.9341	0.8539	0.9657	0.9677	0.8780	0.9685	**0.9993**
Average	0.7981	0.7969	0.7930	0.7531	0.7738	0.7993	0.7980	0.6723	0.8021	**0.8567**

We used "ABS_method" to represent our proposed method. We used different existing methods to compare with our works: "Over" and "Under" to denote random oversampling and random undersampling, respectively; "CBO" and "SBC" to represent the cluster-based oversampling method and undersampling method, respectively. This paper has also compared: "ProWRAS" [1] with the proposed method on 12 datasets. In addition, we also compare: "Smote"; "MBS" which represent the linear feature model and "ADASYN" [9]. We used the R programming language, an open-source programming language and a free software environment in this experiment. The experimental results, including the average AUCs for all methods, are presented in Table 1. The experimental results show that the proposed method consistently outperforms the given datasets as the number of iterations increases. Table 1 shows that we used 12 different datasets with different existing methods, and the ABS method outperforms all other methods based on AUC results.

4 Integration of Augmentation Based Synthetic Sampling Method and Ensemble Techniques

The bagging method is a bootstrap ensemble method that can improve model stability. Boosting is a machine learning technique to improve the performance of a classifier. The most well known boosting algorithm is Adaboost, or Adaptive Boosting [13]. Using the ensemble learning approach to deal with imbalance problems is popular, so we further combined the proposed ABS with the boosting technique to devise a method called ABSBoost, which is an integration of AdaBoost.M2 [11] and ABS. ABS was performed to increase minority samples at each iteration of model training so that each weak classifier could be trained on a relatively balanced subset.

Table 2: Average AUC results for different ensemble-based methods.

Dataset	**ABSBoost**	RUSBoost	UnderBagging
Pima	**0.8473**	0.8168	0.8273
Haberman	**0.7107**	0.6600	0.6835
Satimage	**0.9719**	0.9512	0.9385
Ecoli	**0.9465**	0.9249	0.9312
Ionosphere	**0.9895**	0.9558	0.9410
Vehicle	**0.9998**	0.9879	0.9757
Credit	**0.7842**	0.7168	0.7473
Diabetes	**0.6310**	0.5627	0.6035
Hmeq	**0.9872**	0.9265	0.9113
Promotion	**0.6246**	0.5916	0.5166
Bank	**0.8863**	0.8746	0.8557
Spambase	**0.9967**	0.9763	0.9690
Average	**0.8646**	0.8288	0.8251

We compare ABSBoost with state of the art ensemble-based methods, including RUSBoost and UnderBagging. The experimental results are presented in Table 2, indicating that ABSBoost is effective and better than the existing method.

5 Conclusion

In data mining, imbalanced data is a prevalent problem in the world. In order to overcome the challenge that some standard imbalanced data techniques cannot accurately balance majority and minority classes, Augmentation Based Synthetic Sampling (ABS) for imbalanced data classification is proposed. ABS method concatenates the features and increases the number of samples from existing samples to generate synthetic data. This study conducted experiments on 12 datasets and compared the proposed method with existing methods. In addition, this study combines the proposed method with the boosting technique to devise a method called ABSBoost and compare the performance of the combination with two states of the art ensemble-based methods. The experimental outcomes show that the proposed method and ABSBoost are effective and better than other existing methods on the given datasets. In future work, this paper will focus on other datasets by keeping the original data and balancing data such as primary and minority classes to further improve the algorithm's accuracy. At the same time, this paper will apply reinforcement learning with different methods to improve the execution efficiency of the algorithm.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China under Grants Nos. 62176200, 61773304, and 61871306, the Natural Science Basic Research Program of Shaanxi under Grant No.2022JC-45, 2022JQ-616 and the Open Research Projects of Zhejiang Lab under Grant 2021KG0AB03, the 111 Project, the National Key R&D Program of China, the Guangdong Provincial Key Laboratory under Grant No. 2020B121201001 and the GuangDong Basic and Applied Basic Research Foundation under Grant No. 2021A1515110686.

References

1. Bej, S., Schulz, K., Srivastava, P., Wolfien, M., Wolkenhauer, O.: A multi schematic classifier independent oversampling approach for imbalanced datasets. IEEE Access **9**, 123358–123374 (2021)
2. Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Synthetic data augmentation using gan for improved liver lesion classification. In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). pp. 289–293. IEEE (2018)
3. Gameng, H.A., Gerardo, B.B., Medina, R.P.: Modified adaptive synthetic smote to improve classification performance in imbalanced datasets. In: 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS). pp. 1–5. IEEE (2019)
4. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on knowledge and data engineering **21**(9), 1263–1284 (2009)
5. Jiang, X., Ge, Z.: Data augmentation classifier for imbalanced fault classification. IEEE Transactions on Automation Science and Engineering **18**(3), 1206–1217 (2020)
6. Khurana, A., Verma, O.P.: Optimal feature selection for imbalanced text classification. IEEE Transactions on Artificial Intelligence (2022)
7. Laermann, J., Samek, W., Strodthoff, N.: Achieving generalizable robustness of deep neural networks by stability training. In: German conference on pattern recognition. pp. 360–373. Springer (2019)
8. Liu, C.L., Chang, Y.H.: Learning from imbalanced data with deep density hybrid sampling. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2022)
9. Liu, C.L., Hsieh, P.Y.: Model-based synthetic sampling for imbalanced data. IEEE Transactions on Knowledge and Data Engineering **32**(8), 1543–1556 (2019)
10. Niu, J., Liu, Z., Lu, Y., Wen, Z.: Evidential combination of classifiers for imbalanced data. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2022)
11. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence rated predictions. Machine learning **37**(3), 297–336 (1999)
12. Taylor, L., Nitschke, G.: Improving deep learning with generic data augmentation. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1542–1547. IEEE (2018)
13. Wah, Y.B., Rahman, H.A.A., He, H., Bulgiba, A.: Handling imbalanced dataset using svm and knn approach. In: AIP Conference Proceedings. vol. 1750, p. 020023. AIP Publishing LLC (2016)
14. Yan, Y., Zhu, Y., Liu, R., Zhang, Y., Zhang, Y., Zhang, L.: Spatial distribution-based imbalanced undersampling. IEEE Transactions on Knowledge and Data Engineering (2022)
15. Yuan, Z., Zhao, P.: An improved ensemble learning for imbalanced data classification. In: 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC). pp. 408–411. IEEE (2019)
16. Yusof, R., Kasmiran, K.A., Mustapha, A., Mustapha, N., MOHD ZIN, N.A.: Techniques for handling imbalanced datasets when producing classifier models. Journal of Theoretical & Applied Information Technology **95**(7) (2017)