

Experimental Comparison of Sampling Techniques for Imbalanced Datasets Using Various Classification Models

Sanjibani Sudha Pattanayak and Minakhi Rout

Abstract Imbalanced dataset is a dataset, in which the number of samples in different classes is highly uneven, which makes it very challenging for classification, i.e., classification becomes very tough as the result may get biased by the dominating class values. But misclassification of minor class sample or interested samples is very much costlier. So to provide solution to this problem, various studies have been made out of which sampling techniques are successfully adopted to preprocess the imbalance datasets. In this paper, experimental comparison of two pioneering sampling techniques SMOTE and MWMOTE is simulated using the classification models SVM, RBF, and MLP.

Keywords Sampling techniques • SMOTE • MWMOTE • SVM • RBF • MLP

1 Introduction

The dataset, in which the ratio of number of samples in major and minor classes is very high, is called imbalanced dataset. This imbalanced nature of dataset is undesirable for a good classification because classifier shows very good result for major datasets but poor result for the minor datasets, i.e., the classifier could not be trained for minor class properly. But misclassification of minor class sample is much more costlier than major class sample, i.e., classification of minor class sample with high accuracy is very much essential. Unfortunately, most of the real-world applications like fraudulent transaction detections, detecting network intrusion, Web mining, medical diagnostics, and many other find imbalanced data. And it is always very much essential to give justice to the minor class. Various solutions have been proposed till date to eradicate the imbalanced nature. The solutions might be in data

S. S. Pattanayak (✉) · M. Rout
ITER, Siksha O Anusandhan University, Bhubaneswar 751030, Odisha, India
e-mail: sanjibani@gmail.com

M. Rout
e-mail: minakhirout@soauniversity.ac.in

level or in algorithm level. At data level, sampling is considered as a major technique to handle imbalance nature of dataset. Data are sampled to bring balance between uneven classes. Broadly, the sampling techniques can be classified into two categories: undersampling and oversampling. Using undersampling, the size of majority class can be reduced to match with the minority class. But in this approach, there is a chance of losing important samples. So, in many cases, oversampling is adopted instead of undersampling, whereby generating new synthetic samples, the size of minority class can be developed.

1.1 Literature Survey

Significant works have been done to handle the imbalance nature of datasets. Sampling is one of the major techniques which have been adopted highly by researchers. In SMOTE (Chawla 2008), the dataset is oversampled by creating synthetic examples of minority class. Synthetic samples are generated by considering the feature sample (Minor sample) and its k -nearest neighbor. K value is chosen depending upon how many synthetic samples you need to generate. The difference between the featured sample and nearest neighbor is taken and it is multiplied with a random value in the range $(0, 1)$. This result is added with the same featured sample to generate the synthetic sample. This is how it makes the region of minor samples more general [1]. Many variations of SMOTE have been mentioned here. Chawla et al. [2] proposed SMOTEboost that combines the features of SMOTE and boosting to focus more on difficult examples that belong to the minority class than to the majority class. It gives higher weight to synthetic samples and thus adjusts the imbalance nature [2]. Unlike SMOTE, which generates the synthetic samples from every minor sample, MSMOTE method by Hu et al. [3] considers the distribution of minority class samples and latent noise in the dataset. It eliminates those noisy samples [3]. Maciejewski and Stefanowski [4] proposed LNSMOTE which is more careful about choosing the local neighborhood to avoid the overgeneralization cases of SMOTE [4]. Han et al. [5] proposed the method of Borderline SMOTE, in which, instead of considering all the minor samples, only the examples on the borderline and the ones nearby are used to generate synthetic samples as they prone to more error, i.e., misclassification [5]. He et al. [6] proposed Adasyn algorithm that assigns weight to minor samples. As per the algorithm, the samples which are more difficult to learn will be assigned a higher weight and more synthetic samples will be generated from the samples having higher weight than others and this is how it tries to bring justification [6]. MWMOTE [7] first identifies the difficult to learn minor samples and assigns weight according to their distance from nearest majority class samples. It also eliminates the noisy samples. Then, it generates the synthetic samples by forming clusters of minor samples. Now, the featured sample and the nearest neighbor sample will be chosen from the same cluster. Hence, create more concrete synthetic samples [7]. Jayashree and Gavya [8] have used oversampling technique MWMOTE and undersampling technique SSO for better imbalanced learning [19].

1.2 Objective

The objective of this paper is to study the performance of classifiers on imbalanced dataset before oversampling and after oversampling. Also, the two well-known oversampling techniques, Synthetic Minority Oversampling Technique (SMOTE) and Majority Weighted Minority Oversampling Technique (MWMOTE), have been compared using three distinct classification models.

2 Classification Models and Sampling Techniques

2.1 Classification Models

In this paper, popular classification models like SVM and artificial neural networks like MLP and RBF have been used for simulation.

SVM: Support vector machine is a supervised learning machine used to classify two-class problems linearly. Multi-class problems can be modified or shaped to two-class problems so that SVM can work with that too. SVM operates by constructing a hyperplane between the two sets of data. It is very important to choose the right hyperplane for better accuracy because the data near the borderline are most difficult to learn. The hyperplane that has the largest distance to the nearest data point is considered as the best hyperplane. Generally, the kernel trick is used to solve the nonlinear problems. Different variations of the SVM have been developed and tested by the researchers for imbalanced datasets. Tang et al. [14] proposed modified SVM to handle imbalanced dataset and they found that among different variations of SVM, GSVM-RU shows the best efficiency [14]. Another variation of SVM, z-SVM of Imam et al. [15], maintains a good boundary between classes and the separator line [15].

RBFN: Radial basis function network is another powerful supervised learning machine which uses a radial basis function to train the neurons. It has a three-layer architecture in which the input layer consists of the feature vectors which will be classified. Hidden layer uses a radial basis function, whereas the output layer deals with weight vector to generate the actual output for the given input. Commonly, Gaussian function is used as the radial basis function. Perez-Godoy et al. [16] experimented on RBF with LMS and SVD for imbalanced dataset and found that SVD outperforms LMS [16]. Haddad et al. [17] studied the effect of imbalanced training set size using RBF and they found that with the increased size of imbalanced dataset, the classifier performance degrades [17].

MLP: Multilayer perceptron uses backpropagation technique to train the network in a supervised method. It has input layer, output layer, and in between number of hidden layers. It uses nonlinear activation functions to solve nonlinear problems. To get

a right match of the input and output, the weight vector that has been provided at the input layer is adjusted with the backpropagation of error to the neural network. Bruzzone and Serpico 1997 proposed a multilayer perceptron to classify the imbalanced remote-sensing data and found improved speed and more stable result [8]. Oh 2011 modified the error function of MLP to update the weight value. Here, the weight value of minority class intensifies and of majority class weakens, and hence avoids the biasness in the imbalanced datasets [18].

2.2 Sampling Techniques

Sampling techniques are most popular and widely used techniques to handle imbalanced dataset problem. Here, it tries to balance the number of samples by dropping few samples from majority class or adding new synthetic samples to minority class. Two popular oversampling methods SMOTE and MWMOTE have been discussed below. Instead of replication of the same data, if new synthetic samples will be generated, the region of minority class will become more broader, and hence learning can become more generic.

SMOTE: SMOTE which is abbreviated for Synthetic Minority Oversampling Technique generates synthetic samples by considering k -nearest neighbors of each of the minority samples. The value of k depends on how many number of synthetic samples will be generated. The difference between the feature sample under consideration and the neighboring sample will be computed which will be multiplied with a random number between 0 and 1. This multiplication result will be now added with feature sample in turn to create the synthetic sample.

MWMOTE: MWMOTE or Majority Weighted Minority Oversampling Technique is another oversampling technique which is more cautious in choosing the minority class samples to generate synthetic samples. Unlike SMOTE, it gives priority to those minority class samples which are more difficult to learn. Experiment shows that the samples which lie near border, the samples which are a part of a minority cluster that is sparsely populated, are more difficult to classify. So at first, all difficult to learn minority samples are identified. Next, their difficulty level is found out so that it can be decided which minority sample will contribute how many numbers of synthetic samples.

3 Performance Measure

Confusion Matrix Confusion matrix is a tabular representation of accuracy of a classifier. Rows represent the true class, whereas the columns represent the predicted class. TP stands for true positive, i.e., TP value says how many positive samples (i.e., minority class samples) are identified correctly. FP (False positive) indicates the number of positive samples misclassified. Similarly, TN value says the negative

samples (majority class samples) that are classified correctly and FN represents number of negative samples that are misclassified. So the main diagonal of the matrix shows how many samples are predicted correctly.

The traditional method to measure the accuracy, shown in Eq. 1, is not suitable for imbalanced dataset:

$$\text{OverallAccuracy} = \frac{(TP + TN)}{(TP + FN + FP + TN)}. \quad (1)$$

In imbalanced dataset, more than 98% data belong to majority class set. So even if the classifier failed to identify the minority class samples, it will show very high accuracy. But high accuracy with correct identification of minority class set is very much important. (For example, correct identification of one positive cancer sample is highly necessary among 1 lakh negative samples. The misclassification of the same is intolerably dangerous here.) Hence, various metrics listed from Eqs. 2–7 are used for performance measure of classifier over imbalanced datasets.

TPR says out of total minority class samples, how many are classified correctly:

$$\text{TPR(} \text{TruePositiveRateorRecallorSensitivity)} = \frac{TP}{(TP + FN)}. \quad (2)$$

TNR measures how many majority class samples are correctly classified. High TNR value is desirable for more majority class samples that are correctly identified:

$$\text{TNR(} \text{TrueNegativeRateorSpecificity)} = \frac{TN}{(TN + FP)}. \quad (3)$$

FPR or False Positive Rate indicates how many minority class samples are misclassified:

$$\text{FPR} = \frac{FP}{(TN + FP)}. \quad (4)$$

While TPR or recall says out of total actual minority class samples how many are classified correctly, precision says out of total predicted minority class samples, how many are actually belong to minority class:

$$\text{Precision} = \frac{TP}{(TP + FP)}. \quad (5)$$

Fmeasure is the harmonic mean of precision and recall, i.e., it gives the balance between precision and recall. Its value will be high for high precision and high recall:

$$\text{Fmeasure} = \frac{(2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}. \quad (6)$$

Table 1 Dataset description

Dataset name	No. of features	Total no. of samples	No. of samples in majority class	No. of samples in minority class	Imbalance ratio
Winequality	11	691	681	10	68.1
Poker	10	1477	1460	17	85.88
kddcup-rootkit-imap-vs	41	2225	2203	22	100.14

Gmean is the geometric mean, i.e., a kind of average of two parameters recall or sensitivity and specificity:

$$Gmean = \sqrt{Sensitivity * Specificity}. \quad (7)$$

4 Dataset Description

In this paper, the simulation has been done on the openly available two-class datasets.

The datasets' descriptions have been given in Table 1. Winequality dataset has total 691 samples out of which 681 samples belong to majority class, whereas only 10 samples belong to minority class. Hence, the imbalance ratio is here 68.1. It poses 11 numbers of feature. Next dataset, i.e., Poker which is having 10 features, is a bigger dataset than the first one. It has 1460 samples in majority class and only 17 samples in minority class. So here the imbalance ratio is 85.88. The third dataset kddcup is the biggest one among three having more features and more imbalanced proportion of majority and minority class samples. Here, out of total 2225 samples, 2203 samples belong to majority class and only 22 samples belong to minority class. Hence, the imbalance ratio is more than 100.

5 Simulation Study and Result

Different steps that are involved in the entire simulation process are shown below:

- Choosing the Datasets,
- Introducing Synthetic Samples,
- Modeling the Classifiers,
- Preparing the Training data and Testing data, and
- Measuring the performance.

Without oversampling, when the datasets are passed to these classifiers, it shows very poor performance, i.e., the classifiers could not be trained for the minority class. So for correct classification, datasets are oversampled to generate synthetic samples

of minority class. For each dataset, two sets of oversampled data are generated using SMOTE and using MWMOTE. As per previous studies, 200–400% sampling shows good result, so here the dataset has been sampled to 400%.

Then the oversampled data are passed to the classifiers. Each dataset has been distributed in 80:20 ratios for training and testing purpose. The outcome of each classification model has been shown in separate table. Here, the simulation has been done using the mathematical tool Matlab. The underperformance of classifiers without sampling is very clear in Table 2. After sampling, significant improvement in the performance is observed which has been shown in Tables 3, 4 and 5.

Among the three classification models, SVM is giving best result for the said datasets. It is also found that MWMOTE sampling is giving better result than SMOTE.

Table 2 Performance of classifiers without sampling

Dataset name	Classification model	Precision	Recall
Winequality	SVM	0	0
	RBF	0	0
	MLP	0	0
Poker	SVM	0	0
	RBF	0	0
	MLP	0.2	0.5
kddcup- rootkit- imap	SVM	1	1
	RBF	0	0
	MLP	1	1

Table 3 Performance using SVM with sampling

Dataset name	Sampling method	Recall	Precision	Fmeasure	Gmean
Winequality	SMOTE	1	0.75	0.8571	0.9928
	MWMOTE	1	1	1	1
Poker	SMOTE	1	0.5789	0.7333	0.9864
	MWMOTE	1	1	1	1
kddcup- rootkit-imap	SMOTE	1	1	1	1
	MWMOTE	1	1	1	1

Table 4 Performance using RBF with sampling

Dataset name	Sampling method	Recall	Precision	Fmeasure	Gmean
Winequality	SMOTE	0.5	0.4	0.4444	0.6996
	MWMOTE	0.8750	0.8750	0.8750	0.9320
Poker	SMOTE	0.7647	0.7222	0.7428	0.8669
	MWMOTE	0.9047	0.95	0.9268	0.9495
kddcup-rootkit-imap	SMOTE	0.6666	1	0.8	0.8165
	MWMOTE	0.7948	1	0.8857	0.8915

Table 5 Performance using MLP with sampling

Dataset name	Sampling method	Recall	Precision	Fmeasure	Gmean
Winequality	SMOTE	0.5	0.4	0.4444	0.6917
	MWMOTE	0.8	0.9231	0.8571	0.8910
Poker	SMOTE	0.7222	0.6190	0.6666	0.8380
	MWMOTE	1	1	1	1
kddcup-rootkit-imap	SMOTE	1	1	1	1
	MWMOTE	1	1	1	1

6 Conclusion

Handling imbalanced datasets is a real challenge in most of the real-life applications. Here, data-level solutions for imbalanced datasets have been described. From the simulation work, two oversampling techniques using three different classifiers SVM, RBF, and MLP are analyzed. It is found that the performance of MWMOTE is surpassing SMOTE in most of the cases. The comparative performance study of various classifiers has been shown which can be helpful for further research.

7 Future Work

Intensive research study has been made since one decade to handle imbalanced datasets. Here, as for future study, the following points can be considered:

- The overfitting problem with oversampling needs to be focused, i.e., the models need to be trained more about the generalized situations.

- Different undersampling techniques can be verified on the same data sets with the same classification models.
- Hybridization of these oversampling techniques with undersampling can be verified using the same classification models.
- Other sampling techniques like Smoteboost, Adasyn, and RAMO can be compared using the same classification model.
- These data-level solutions can be combined with algorithmic solution and checked if it is performing better.

References

1. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. In: *Foundations and Trends in Information Retrieval*, vol. 16, pp. 321–357 (2002)
2. Chawla, N V., Lazarevic, A., Hall, O.: SMOTE Boost improving prediction of the minority class in boosting. In: *The 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 1322–1328. Springer (2003)
3. Hu, S., Liang, Y., Ma, L., He, Y.: Improving classification performance when training data is imbalanced. *IEEE* (2005)
4. Maciejewski, T., Stefanowski, J.: Local neighborhood extension of SMOTE for mining imbalanced data. In: *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pp. 978-1-4244-99 (2011)
5. Han, H., Wang, W.Y., Mao, B.H.: Borderline-SMOTE: a new oversampling method in imbalanced data sets learning. In: *Proceedings International Conference Intelligent Computing*, pp. 878–887 (2005)
6. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: *Proceedings of International Joint Conference Neural Networks*, pp. 1322–1328 (2008)
7. Barua, S., Islam, M.M., Yao, X., Murase, K.: MWMOTE majority weighted minority over-sampling technique for imbalanced data set learning. *IEEE Trans. Knowl. Data Eng.* **26**(2) (2014)
8. Jayashree, S., Alice Gavya, A.: Classification of imbalanced problem by MWMOTE and SSO. *IJMTES* **2**(5) (2015)
9. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. *Intell. Data Anal.* **6**(5), 429–449 (2002)
10. He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **21**(9), 1263–1284 (2009)
11. Buckland, M., Gey, A.: The relationship between recall and precision. *J. Am. Soc. Inf. Sci.* **45**(1), 12–19 (1994)
12. Ramentol, E., Caballero, Y., Bello, R., Herrera, F.: SMOTE-RSB: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. *Knowledge and Information System*, vol. 33(2), pp. 245–265. Springer (2012)
13. Han, H., Wang, W.Y., Mao, B.H.: Borderline-SMOTE a new oversampling method in data sets learning. In: *Proceedings of International Conference on Intelligent Computing*, pp. 878–887 (2005)
14. Tang, Y., Zhang, Y.Q., Chawla, N.V., Krasser, S.: Modeling for highly imbalanced classification. *J. latex class files.* **1**(11) (2002)
15. Imam, T., Ting, K.M., Kamruzzaman, J.: z-SVM: An SVM for Improved Classification Of Imbalanced Data. *Advances in Artificial Intelligence*, vol. 4304, pp. 264–273 (2006)

16. Prez-Godoy, M.D., Rivera, A.J., Carmona, C.J., delJesus, M.J.: Training algorithms for radial basis function networks to tackle learning processes with imbalanced data-sets. *Appl. Soft Comput.* **25**, 26–39 (2014)
17. Haddad, L., Morris, C W., Boddy, L.: Training radial basis function neural networks: effects of training set size and imbalanced training sets. *J. Microbiol. Methods* **43**(1), 33–44 (2000)

Progress in Advanced Computing and Intelligent
Engineering

Proceedings of ICACIE 2016, Volume 2

Saeed, K.; Chaki, N.; Pati, B.; Bakshi, S.; Mohapatra,
D.P. (Eds.)

2018, XX, 723 p. 319 illus., 185 illus. in color., Softcover

ISBN: 978-981-10-6874-4