

A Novel Approach for Handling Outliers in Imbalanced Data

Gillala Rekha^{1*}, V.Krishna Reddy²

^{1,2}Department of CSE, KLEF, KL University, India

*Corresponding author E-mail: rekha.jogam@gmail.com

Abstract

Most of the traditional classification algorithms assume their training data to be well-balanced in terms of class distribution. Real-world datasets, however, are imbalanced in nature thus degrade the performance of the traditional classifiers. To solve this problem, many strategies are adopted to balance the class distribution at the data level. The data level methods balance the imbalance distribution between majority and minority classes using either oversampling or under sampling techniques. The main concern of this paper is to remove the outliers that may generate while using oversampling techniques. In this study, we proposed a novel approach for solving the class imbalance problem at data level by using modified SMOTE to remove the outliers that may exist after synthetic data generation using SMOTE oversampling technique. We extensively compare our approach with SMOTE, SMOTE+ENN, SMOTE+Tomek-Link using 9 datasets from keel repository using classification algorithms. The result reveals that our approach improves the prediction performance for most of the classification algorithms and achieves better performance compared to the existing approaches.

Keywords: Classification Algorithms, Class Imbalance Learning, SMOTE, Resampling and Mahalanobis Distance.

1. Introduction

In most of the real-world data, the class imbalance problem is persistent and causing trouble to a large section of the data mining society. This problem is prevalent in many applications such as fraud and intrusion detection, risk management, text classification, medical diagnosis and monitoring, and many other [1]. In classification, when the representative examples of one class are more frequent than that of another class, then this data set is represented as an imbalanced dataset. In class-imbalanced data sets, the number of instance of some classes appears more frequently than the other. The class with more number of instances are labelled as majority class and the class with less number of instances as a minority class. The primary concern with the imbalanced learning problem is the ability of skewed data which significantly compromise the performance of most standard learning algorithms.

Most standard algorithms expect balanced class distributions or equal misclassification costs. Therefore, the evaluation criterion, which guides the learning procedure, can lead to ignore minority class examples by treating them as noise and provide unfavourable accuracies across the classes of the data as a result. Moreover, datasets with skewed class distribution usually suffer from class overlapping, small sample size or small disjuncts, which difficult standard classifier learning algorithms [2] [3]. Recently, the class imbalance problem emerges as one of the challenging problems in data mining community [4]. This problem has been widely discussed by the research community and many techniques have been developed to address the class imbalance problem.

From the learning viewpoint, the class with lesser instances is usually the class of interest [1]. The standard classifier learning algorithms assume that imbalanced datasets are equally distributed and show bias towards majority classes thus generate inaccurate classification model performance. Class imbalance involves a series of difficulties in learning such as small sample size, class overlapping, and small disjuncts. To address the class imbalance

problem, a various number of techniques have been proposed by the research community. In general, these techniques are broadly categorized into 1. Data level approach [5][6][1], 2. Algorithm level approach [7][8][9] 3. Cost-sensitive learning approach [10][11] 4. Ensemble method [12][13].

- Data level approach also known as an external approach. It employs pre-processing to re-balance the class distribution of imbalanced data sets. The pre-processing is done either by under-sampling or over-sampling techniques to reduce the imbalance ratio in the data set.
- Algorithm level approaches also known as an internal approach. It modifies the classification algorithm to bias the learning towards the minority class. These algorithms require knowledge to learn from the imbalance data distribution before training the classifier.
- Cost-sensitive learning approach combines both data level and algorithm approaches to incorporate different misclassification cost for each class.
- Ensemble method uses the ensembles of classifiers. It increases the accuracy of a classifier by training different classifiers and combines their result to generate a single class label.

At data level, a pre-processing technique is applied to balance the imbalanced data sets. Resampling techniques such as under-sampling, oversampling and hybrid method are used for generating synthetic data.

However, most of the oversampling techniques at data level may generate data samples very much similar to existing samples by considering only the nearest neighbour samples. To overcome these problems, we propose a novel approach to solve the class imbalance problem at the data level. The main motivation behind this method is to balance the training data by removing noise lying

in the data in the form of outliers after Synthetic minority over-sampling techniques (SMOTE). SMOTE generates synthetic data using k-nearest neighbour algorithm [6][14][15]. This technique selects the data instances that are the nearest neighbours using Euclidean distance. After synthetic sample generation, noise and outliers usually present in the data instances. However, selecting only those data instances that are nearer may pose the potential challenge of generating noise and sparse data instances.

We conducted empirical experiments to show the performance of the proposed approach with SMOTE[6], SMOTE+ENN[14] and SMOTE+Tomek Link[15] using 9 imbalance datasets from KEEL repository. we evaluated the resampled datasets by using C4.5, K-NN, SVM, RIPPER and NB classification algorithms. Based on the experiment, we observed that our proposed algorithm shows significant improvement on C4.5, k-Nearest Neighbor (K-NN) and Naive Bayes (NB) algorithms in terms of Precision, Recall, F-Measure, G-Mean and AUC.

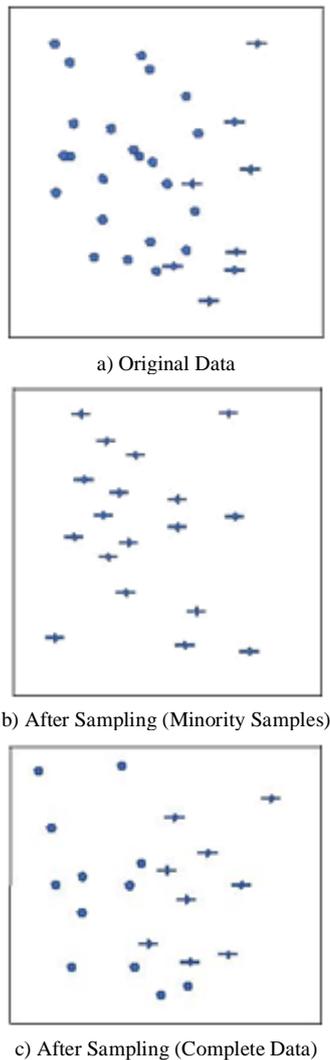


Fig. 1: Class Imbalance Data; (a) Original samples (b) Synthetic samples generated for minority samples (c) Resampled data

The organization of the paper is as follows. Section 2 discusses the concerns over oversampling techniques and the motivation behind our approach. Section 3 provides the proposed method and its application to generate synthetic samples. The data description is presented in Section 4. Section 5 presents the experimental evaluation. Finally, section 6 contains the final conclusion.

2. Motivation

Synthetic minority oversampling method (SMOTE) generates synthetic data using the k-nearest neighbour algorithm. [6][14][15]. This technique selects the data instances that are the nearest neighbour's using Euclidean distance. After synthetic sample generation, some problems usually present in the data instances. However, selecting only those data instances that are nearer to the existing samples may pose the potential challenge of generating noise and sparse data instances. The figure 1 presents examples of data samples before sampling and after sampling. It also displays the data samples generated after resampling were in some samples may tend to fall outside the boundary of minority class as outliers which may lead to bias in classification. To overcome this problem, the main motivation behind this method is to balance the training data by removing noise lying in the data in the form of an outlier after SMOTE. The technique we used is Mahalanobis distance[16] which is known to be useful for identifying outliers.

Given a skewed data set and a particular data point, a basic concern is about the extremeness of the data point relative to the other data points. For univariate data, Euclidean distance work better in identifying the data points, but for multivariate data, we must modify in order to relate distance and skewness. That modification using Mahalanobis distance enables a powerful technique for detecting multivariate outliers. In our proposed method Mahalanobis distance is used to remove outliers appeared in the data after generating the synthetic samples. Mahalanobis distance measure is considered as unit-less measure and provides a relative measure of an instance distance and helps in detecting outliers. Considering two data instances $x = (x_1, x_2, x_3, \dots, x_n)^T$ and $y = (y_1, y_2, y_3, \dots, y_n)^T$, the Mahalanobis distance between them is defined as

$$d_M(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)} \quad (1)$$

where S^{-1} is the covariance matrix. We use this measure to help rank and sort the data samples according to their distance in a decreasing order. By sorting the data, we are able to distinguish data samples that are far or close from the central data instance. It works well for multivariate datasets and also overcomes the inherent scale and correlation problems, associated with Euclidean distance. The removal of outlier samples might provide a better performance on classifiers.

3. Proposed Framework

The intuition behind our approach is to remove the outliers existing in the data samples after generating the synthetic samples for minority classes. The proposed approach is compared with the common Synthetic Minority Oversampling Techniques (SMOTE) proposed by Chawla et al.[6]. The SMOTE technique oversamples the minority classes by generating synthetic data by introducing data samples along the line segments that join any of the k nearest neighbour's minority class sample.

Our proposed method comprises two stages. Stage one is preprocessing stage and the second stage is for model generation. To generate the synthetic data for minority samples, in the first stage we divided the data samples into minority and majority data samples based on their class label. Then, for minority samples, we generated synthetic data using SMOTE. The synthetic data samples are generated to balance the class samples. Then, we combine the data samples of both minority and majority class which represent a balance data sets. Now, we find the outliers in the data and measure the diversity in the data sets using Mahalanobis distance [16]. This measure is adopted because it works well to eliminate the diversity existing in the data. It is adopted because of its mul-

tivariate effect size. Using this measure, we generate the ranks and sort the data instances in decreasing order to their distance. We eliminate the data samples that are far or close from the centre and consider the remaining data samples for the next stage [17-24].

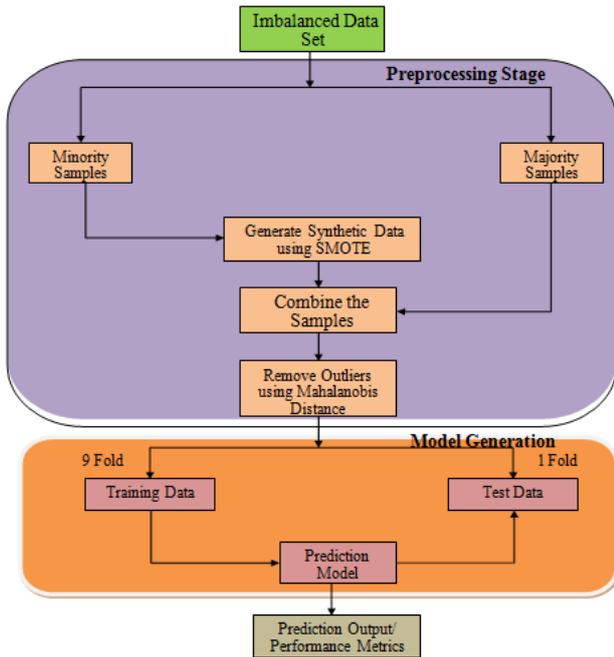


Fig.2: Framework of the proposed method

In the second stage, we performed model generation using the data samples produce from the first stage. We trained the data samples on various classification algorithms as mentioned in Table 4. We employed 10-fold cross-validation techniques, were in 2/3 of the data samples are picked randomly as training data and the remaining 1/3 samples as the testing data. The framework of the proposed method is showed in Fig 2.

4. Dataset Description

For the experiment, all of our datasets are from the KEEL data set repository [25]. We consider nine imbalanced datasets from various application domains. All the datasets are binary class problems. The name of the data sets used, no. of attributes and its imbalanced ratio (IR) is presented in Table 3.

Table 1: Datasets Used in the Experiment

Dataset	No. of Attributes	Imbalanced Ratio (IR)
glass 1	9	1.82
glass 6	9	6.38
haberman	3	2.78
iris0	4	2
new thyroid 1	5	5.14
new_thyroid 2	5	5.14
Pima	8	1.87
vehicle0	18	3.25
wisconsin	9	1.86

5. Experimental Evaluation

For the experiments, we have used five classification algorithms to evaluate the effects of the synthetic datasets generated by using baseline SMOTE, SMOTE+ENN, SMOTE+Tomek Link and proposed SMOTE+MD. The algorithms such as C4.5, k-NN, NB,

Ripper, and SVM are used. We have considered 10-fold cross validation during model construction.

The overall performance is measured by Precision, Recall, F-Measure, G-Mean and ROC (AUC) in our experiment. The results are summarized in Tables 4 in terms of Precision, Recall, F-measure, G-mean, Root Mean Square Error(RMSE), Accuracy on different classifiers. From the above Table 7, we find that our method performed well on iris0, new thyroid 1, new thyroid 2, Wisconsin datasets with high precision, recall, f measure, G mean, and AUC using C4.5 classifier. Our method shows low RMSE for new thyroid 1 and Wisconsin datasets for the C4.5 classifier.

With K-NN classifier as shown in Table 8, our method performed well on iris0 with 100% accuracy and 99% accuracy for new thyroid 1. In terms of precision, recall, F-measure, G-mean and AUC our method outperformed in iris0, new thyroid 1, new thyroid 2, Wisconsin datasets. Table 9 shows the performance of our method with NB classifier. The results show a better performance with 100%, 97.81%, 97.81%, 97.78% accuracy on iris0, new thyroid 1, new thyroid 2, and Wisconsin datasets respectively. We also observe a high precision, recall, F-measure, G-mean, and AUC on iris0, new thyroid 1, and new thyroid 2, Wisconsin datasets. We noticed a much low RMSE for iris0, new thyroid 1, new thyroid 2, Wisconsin datasets. Using RIPPER classifier as presented in Table 10, our method shows better AUC on new thyroid 2 datasets. Our method also showed better performance on glass1, iris0 datasets in terms of precision, recall, F-measure, G-mean and AUC using SVM classifier as shown in Table 11. Observing the result, we conclude that our method significantly outperformed most of the datasets using all the five classification algorithms. In most of the cases, our method yields better results on iris0, new thyroid 1, new thyroid 2, Wisconsin datasets. Our method showed an outstanding performance while working with C4.5, k-NN, NB classification techniques.

6. Conclusion

In this research, we proposed a novel oversampling approach with outlier removal. The proposed method solves the class imbalance problem at data level by integrating both SMOTE with Mahalanobis Distance technique. The experimental studies demonstrated best performs on class imbalance datasets using C4.5, NB and k-NN classifiers. Our method mainly addresses on outlier removal after generating the synthetic data. First, we apply SMOTE approach to create synthetic samples for minority class. Second, we combined the minority synthetic samples with the majority samples. Third, we generated Mahalanobis distance for each instance and order them in decreasing order. Then, we fixed a 5% removal of the data instances as outlier. Finally, we trained the different classification algorithms on these data instances. The experimental results show the effective performance of our method on baseline method. In future, we will apply our method on many other class imbalanced datasets.

Algorithm	C4.5		kNN		Naïve Bayes		RIPPER		SVM	
	S M O T E	S M O T E D	S M O T E	S M O T E D	S M O T E	S M O T E D	S M O T E	S M O T E D	S M O T E	S M O T E D
glass1	0.783	0.62	0.79	0.82	0.751	0.758	0.79	0.736	0.651	0.778
	0.77	0.759	0.77	0.82	0.66	0.693	0.77	0.736	0.65	0.703

	ll	79		97		89		88		99	
	f-measure	0.779	0.759	0.797	0.802	0.669	0.673	0.788	0.736	0.562	0.681
	G-mean	0.781	0.76	0.797	0.802	0.719	0.725	0.788	0.736	0.624	0.74
	AUC	0.788	0.773	0.897	0.897	0.695	0.71	0.824	0.781	0.599	0.703
glass6	precision	0.973	0.94	0.971	0.968	0.993	0.947	0.996	0.958	0.994	0.952
	recall	0.973	0.94	0.977	0.968	0.992	0.947	0.995	0.957	0.999	0.947
	f-measure	0.973	0.94	0.977	0.968	0.992	0.947	0.995	0.957	0.999	0.947
	G-mean	0.973	0.94	0.977	0.968	0.992	0.947	0.995	0.957	0.994	0.949
	AUC	0.973	0.929	0.994	0.993	0.998	0.948	0.994	0.957	0.993	0.947
	newthyroid1	precision	0.986	0.989	0.983	0.988	0.996	0.993	0.998	0.996	0.997
recall		0.986	0.989	0.983	0.988	0.996	0.993	0.998	0.996	0.997	0.989
f-measure		0.986	0.989	0.983	0.988	0.996	0.993	0.998	0.996	0.997	0.989
G-mean		0.986	0.989	0.983	0.988	0.996	0.993	0.998	0.996	0.997	0.989
AUC		0.991	0.995	0.994	0.999	0.999	0.995	0.999	0.999	0.998	0.989
newthyroid2		precision	0.976	0.985	0.979	0.975	0.996	0.989	0.995	0.996	0.999
	recall	0.976	0.985	0.979	0.973	0.996	0.989	0.992	0.996	0.999	0.996
	f-measure	0.976	0.985	0.979	0.972	0.996	0.989	0.992	0.996	0.999	0.996
	G-mean	0.976	0.985	0.979	0.973	0.996	0.989	0.993	0.996	0.999	0.996
	AUC	0.986	0.991	0.999	0.993	0.999	1	0.999	0.999	0.997	0.996
	iris0	precision	0.994	0.994	1	1	1	1	1	0.987	1
recall		0.999	0.993	1	1	1	1	1	0.987	1	1
		4									
	f-measure	0.994	0.993	1	1	1	1	1	1	0.987	1
	G-mean	0.994	0.993	1	1	1	1	1	1	0.987	1
	AUC	0.994	0.993	1	1	1	1	1	1	0.993	1
newthyroid1	precision	0.986	0.989	0.983	0.988	0.996	0.993	0.998	0.996	0.997	0.989
	recall	0.986	0.989	0.983	0.988	0.996	0.993	0.998	0.996	0.997	0.989
	f-measure	0.986	0.989	0.983	0.988	0.996	0.993	0.998	0.996	0.997	0.989
	G-mean	0.986	0.989	0.983	0.988	0.996	0.993	0.998	0.996	0.997	0.989
	AUC	0.991	0.995	0.994	0.999	0.999	0.995	0.999	0.999	0.998	0.989
	newthyroid2	precision	0.976	0.985	0.979	0.975	0.996	0.989	0.995	0.996	0.999
recall		0.976	0.985	0.979	0.973	0.996	0.989	0.992	0.996	0.999	0.996
f-measure		0.976	0.985	0.979	0.972	0.996	0.989	0.992	0.996	0.999	0.996
G-mean		0.976	0.985	0.979	0.973	0.996	0.989	0.993	0.996	0.999	0.996
AUC		0.986	0.991	0.999	0.993	0.999	1	0.999	0.999	0.997	0.996
pima		precision	0.719	0.747	0.75	0.755	0.735	0.755	0.735	0.742	0.738
	recall	0.718	0.745	0.748	0.753	0.734	0.753	0.734	0.741	0.738	0.766

	f-measure	0.718	0.745	0.748	0.753	0.734	0.741	0.738	0.763	0.752	0.766
	G-measure	0.718	0.746	0.749	0.754	0.734	0.741	0.738	0.765	0.752	0.766
	AUC	0.747	0.759	0.781	0.818	0.811	0.824	0.758	0.792	0.752	0.766
vehicle0	precision	0.952	0.958	0.927	0.935	0.881	0.795	0.957	0.957	0.957	0.961
	recall	0.952	0.957	0.925	0.934	0.765	0.766	0.956	0.957	0.956	0.96
	f-measure	0.952	0.957	0.924	0.934	0.775	0.753	0.955	0.957	0.955	0.96
	G-measure	0.952	0.957	0.926	0.934	0.778	0.777	0.956	0.957	0.956	0.96
	AUC	0.963	0.968	0.986	0.985	0.881	0.812	0.971	0.973	0.956	0.96
wisconsin	precision	0.958	0.975	0.978	0.994	0.961	0.978	0.968	0.978	0.973	0.991
	recall	0.958	0.975	0.977	0.994	0.961	0.978	0.968	0.978	0.973	0.991
	f-measure	0.958	0.975	0.977	0.994	0.961	0.978	0.968	0.978	0.973	0.991
	G-measure	0.958	0.975	0.977	0.994	0.961	0.978	0.968	0.978	0.973	0.991
	AUC	0.965	0.982	0.984	1	0.985	0.996	0.971	0.978	0.973	0.991

Mining, 2008. ICDM'08. Eighth IEEE International Conference on, IEEE, 2008, pp. 143{152.

[4] Q. Yang, X. Wu, 10 challenging problems in data mining research, International Journal of Information Technology & Decision Making 5 (04) (2006) 597{604.

[5] G. E. Batista, R. C. Prati, M. C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD explorations newsletter 6 (1) (2004) 20{29.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-technique, Journal of artificial intelligence research 16 (2002) 321{357.

[7] J. R. Quinlan, Improved estimates for the accuracy of small disjuncts, Machine Learning 6 (1) (1991) 93{98.

[8] B. Zadrozny, C. Elkan, Learning and making decisions when costs and probabilities are both unknown, in: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2001, pp. 204{213.

[9] G. Wu, E. Y. Chang, Kba: Kernel boundary alignment considering imbalanced data distribution, IEEE Transactions on knowledge and data engineering 17 (6) (2005) 786{795.

[10] N. V. Chawla, D. A. Cieslak, L. O. Hall, A. Joshi, Automatically countering imbalance and its empirical relationship to cost, Data Mining and Knowledge Discovery 17 (2) (2008) 225{252.

[11] A. Freitas, A. Costa-Pereira, P. Brazdil, Cost-sensitive decision trees applied to medical data, in: International Conference on Data Warehousing and Knowledge Discovery, Springer, 2007, pp. 303{312.

[12] L. Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33 (1-2) (2010) 1{39.

[13] R. Polikar, Ensemble based systems in decision making, IEEE Circuits and systems magazine 6 (3) (2006) 21{45.

[14] P. Hart, The condensed nearest neighbor rule (corresp.), IEEE transactions on information theory 14 (3) (1968) 515-516.

[15] G. E. Batista, R. C. Prati, M. C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD explorations newsletter 6 (1) (2004) 20-29

[16] P. C. Mahalanobis, On the generalized distance in statistics, National Institute of Science of India, 1936.

[17] R. C. Team, et al., R: A language and environment for statistical computing.

[18] I. H. Witten, E. Frank, L. E. Trigg, M. A. Hall, G. Holmes, S. J. Cunningham, Weka: Practical machine learning tools and techniques with java implementations.

[19] T. Oommen, L. G. Baise, R. M. Vogel, bias and class imbalance in maximum-likelihood logistic regression, Mathematical Geosciences 43 (1) (2011) 99{120.

[20] J. R. Quinlan, C4. 5: programs for machine learning, Elsevier, 2014.

[21] K. P. Murphy, Naive bayes classifiers, University of British Columbia 18.

[22] L. E. Peterson, K-nearest neighbor, Scholarpedia 4 (2) (2009) 1883.

[23] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273{297.

[24] J. Furnkranz, G. Widmer, Incremental reduced error pruning, in: Machine Learning Proceedings 1994, Elsevier, 1994, pp. 70{77.

[25] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework., Journal of Multiple-Valued Logic & Soft Computing 17.

References

[1] N. V. Chawla, N. Japkowicz, A. Kotcz, Special issue on learning from imbalanced data sets, ACM Sigkdd Explorations Newsletter 6 (1) (2004) 1{6.

[2] V. García, R. A. Mollineda, J. S. Sánchez, On the k-nn performance in a challenging scenario of imbalance and overlapping, Pattern Analysis and Applications 11 (3-4) (2008) 269{280.

[3] D. A. Cieslak, N. V. Chawla, Start globally, optimize locally, predict globally: Improving performance on imbalanced data, in: Data