A NOVEL OVERSAMPLING TECHNIQUE TO HANDLE IMBALANCED DATASETS

Ayat Mahmoud
Faculty of Computer Sciences
October University for Modern Sciences and Arts
Cairo, Egypt
E-mail: eng.ayat@gmail.com

Ayman El-Kilany
Information Systems Department
Faculty of Computers and Information, Cairo University
Cairo, Egypt
Email: a.elkilany@fci-cu.edu.eg

Farid Ali
Information Technology Department
Faculty of Computers and Information
Beni-suef University, Egypt
E-mail: farid.cs@gmail.com

Sherif Mazen
Information Systems Department
Faculty of Computers and Information, Cairo University
Cairo, Egypt
Email: s.mazen@fci-cu.edu.eg

KEYWORDS
Imbalance, Oversampling, Classifier.

ABSTRACT

With the amount of data is growing extensively in different domains in the recent years, the data imbalance problem arises frequently. A dataset is called imbalanced when the data of a certain class has significantly more instances than that of other classes of the same dataset. This imbalanced nature of the data negatively affects the performance of a classifier since misclassification of data may cause data analysis results to be inaccurate and hence leads to wrong business decisions. This paper presents a study of the different techniques that are used to handle the imbalanced dataset, and finally proposes a novel oversampling technique to tackle the binary classification of imbalanced dataset problem.

INTRODUCTION

In today's world of internet, there is huge amounts of data generated every day. Therefore, it becomes important to advance the deep understanding of knowledge discovery (KD) and analysis of raw data to support decision-making in businesses. An evolution has been done on classification of data through the learning process. The problem gets more complex when the dataset is imbalanced. A dataset is said to be imbalanced if the class distribution is not uniform (Fernández et al., 2017). In this situation, there are instances from one class are higher in number than the other class. The class having a greater number of samples is named "majority class", and the class having relatively a smaller number of instances is named "minority class".

Researches in the field of machine learning proved that using an uneven distribution of class instances can cause a bias in the performance of the used learning algorithm (Herland et al., 2018). In other words, the classifier gives high accuracy on the majority class, while giving poor accuracy on the minority class. This happens because traditional training measures such as the overall success are inclined by the larger number of instances from the majority class. In many real-world applications, the minority classes are more important as in cancer diagnosis in the medical field applications. That is why classifying the imbalanced datasets has a growing attention from both academia and industry (Chu et al., 2016).

Most of the traditional methods of machine learning had limitations when applied to imbalanced datasets. They do not work well for imbalanced data classification because they assume equal costs for each class. Therefore, the data classification results may be biased and inaccurate. The reason is that traditional machine learning algorithms aimed at enhancing accuracy by reducing the error without taking into consideration the class distribution or balance (Zhang et al., 2018). Several solutions to the misclassification of imbalanced datasets problem were previously proposed in many researches like oversampling, undersampling and cost-sensitive learning.

Classifiers encounter imbalanced distribution in many real-world applications. For example, the number of legal credit card transactions is significantly greater than that of illegal transactions. Most medical fields have similar conditions, where the number of patients needing special care (e.g. rehabilitation or treatment) is significantly lower than the number of those who don't. In many other fields, such as oil deposits identification in satellite images (Cai et al., 2018), there were also class imbalances.

The common problem of these real-life applications is that every class contains completetly different number of instances. Since traditional classifiers were designed to get higher accuracy regardless of the distribution of classes, they gave a less attention to the minority class and therefore caused a noticed bias towards the majority class.

Communications of the ECMS, Volume 34, Issue 1,
Proceedings, ©ECMS Mike Steglich, Christian Mueller,
Gaby Neumann, Mathias Walther (Editors)
ISBN: 978-3-937436-68-5/978-3-937436-69-2(CD) ISSN 2522-2414

In this paper, we argue that oversampling techniques can yield better results to handle imbalanced dataset problem if the majority data was considered during the oversampling process. The proposed technique tackles the imbalanced dataset problem by using a sample of the majority data to create a better new sample of minority data. The evaluation results show that such oversampling technique outperform the standard oversampling algorithm.

The rest of this paper is organized as follows: Section 2 contains the related work and explains the basic techniques that used to solve the problem of imbalanced dataset. In Section 3, we introduce our proposed technique to handle imbalanced data problem while Sections 4 and 5 show the details of the performance evaluation and conclusion remarks.

RELATED WORK

There are three main approaches to tackle imbalanced data problems. Data-level methods -also called external methods- that work on the data in a way to adapt the number of data instances to more balance the distribution. On the other hand, the algorithm-level methods (also called internal methods) that adapt the traditional algorithms of learning to minimize the bias, increase accuracy, and get benefit of mining data that have skew distributions. Hybrid methods combine both data and algorithm level methods. All approaches are summarized in Figure1 and further explained in the next sections.

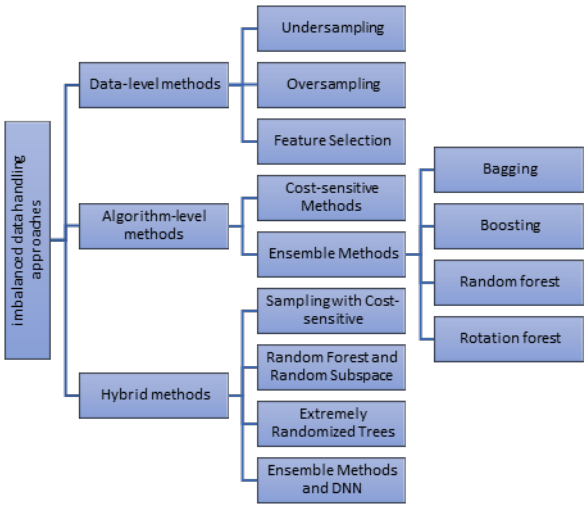

Figure 1: Taxonomy of Imbalanced Data Learning Approaches

Data-level Methods

In data-level methods, the goal is to modify the dataset to make it more suitable to apply a traditional learning algorithm. Three sub-approaches are used to modify datasets, undersampling, feature selection, and oversampling. Undersampling is to remove samples from majority class whereas oversampling is to generate new objects for minority class (Devi & Purkayastha, 2017; Ha & Lee, 2016; Ng et al., 2014). Feature selection means the algorithms that output a subgroup of the input feature set that are more relevant and help a classifier to enhance its performance (Pant & Srivastava, 2015). Traditional approaches use random techniques to select the target samples for sampling (Lin et al., 2017). But this frequently leads to exclusion of important samples or appearance new samples that are meaningless. Consequently, more adapted approaches were proposed to try to maintain structure of classes and generate new data samples that conform to the original distribution. These new adapted algorithms contain also some methods for cleaning the overlapping samples and removing dirty samples that may affect learning process in a negative way (Hu et al., 2015).

In contrast to undersampling, the oversampling adds synthetic samples to the minority class with the aim of balancing the distribution of the classes (Abdi & Hashemi, 2015). The simplest method of oversampling is replication of instances of minority class (Liu et al., 2007). This reduces the class imbalance but on the other hand, it can cause the problem of overfitting. SMOTE is a fundamental oversampling approach that uses data synthesis. SMOTE algorithm implements an oversampling approach to rebalance the training dataset. As an alternative to applying a simple duplication of the minority class examples, the main idea of SMOTE is to generate synthetic instances. This novel data instance is created by interpolation between several minority class instances that are inside a well-defined neighborhood. Therefore, the process is said to be concerned about the feature space instead of *on* the data space, in other words, the algorithm is based on the values of the features and their relationship, in place of considering the data points as a whole (Sáez et al., 2016).

Sampling is the most widely used approach to deal with the problem of imbalanced datasets. The sampling of data turns the unbalanced distribution into a better class distribution. This is achieved by generating data samples or by deleting them. As stated before, there are two key techniques for sampling namely oversampling and undersampling (Bunkhumpornpat & Sinapiromsaran, 2017). The benefits and drawbacks of each will be discussed here.

There are considerable drawbacks of oversampling that can lead to the issue of overfitting, to that the time required to create the classifier or even to harm the learning process. Undersampling approach makes the balancing by the removing class samples. While the particular space can be defined, data loss can be caused by the reducing data size. The bias in the datasets, which influences on minority groups more than a majority, is also an important issue for sampling. Researchers should understand the scope of the problem being tackled and the appropriate classifier for this situation. When supplemented by sampling methods, several classifiers achieve better performance.

Algorithm-level Methods

They modify traditional learning algorithms to lessen the bias found towards the majority class. To achieve this, a good understanding of the learning algorithm is needed, and a clear analysis of reasons for its failure in learning from imbalanced datasets. The most prevalent division is cost sensitive approaches. In cost sensitive approaches the traditional learning algorithm is adapted to include varying penalty for each of class of samples (Khan et al., 2017). This is done by giving a higher cost to the group of instances that is less represented in the dataset. We increase its significance through the mining process (Cheng et al., 2017).

While the approaches of sampling try to achieve more balanced dataset, considering the representativeness of the class instances in the data, cost-sensitive methods take into account the cost of misclassified samples (Khan et al., 2017). Cost-sensitive learning addresses the problem of imbalanced datasets by using different cost formulae that assign some cost for a particular data sample (Sáez et al., 2016).

There are alot of methods to ensemble algorithms such as bagging, boosting, random forest and rotation forest. Till now several approaches have been developed and improvements to traditional methods have been designed to solve the issue of imbalanced distributions (Cai et al., 2018). Bagging, is a machine learning approach that is used to improve accuracy while reducing variance in classifying samples. Boosting means that poor classifications can be combined to create a more correct decision. That is to say, boosting means a number of algorithms that use weights to make weak learners more accurate. Unlike bagging that has run separately from each classifier and at last merges the output without any classifier being preferred.

Hybrid Methods

Sampling-based Approaches with Cost-Sensitive Learning

In these methods, a preprocessing is done to the data samples with imbalanced distribution. this is done by using over or undersampling at first, and then using cost-sensitive approach. Some remarkable researches of this area are Akbani et al. and L'opez et al. (Akbani et al., 2004; López et al., 2012).

Random Forest and Random Subspace Methods

Random trees are still growing, mostly because of their flexibility and good performance. The random forest is treated as a simple to tune technique, unlike other techniques (e.g. GBM) which require careful tuning. Random forests utilize a large number of integrated decision trees.

Extremely Randomized Trees

Extremely random trees (Geurts et al., 2006) use randomness in the training stage in order to produce different sets. In addition to the random subgroup of attributes that choose the most distinctive feature, defining attributes are randomized when extremely random trees are applied.

Ensemble Methods and Deep Neural Networks

In the area of machine learning, DNNs are considered nowadays a major force. In many fields such as speech recognition and object detection and many other fields, DNN has improved dramatically in the last few years (LeCun et al., 2015). DNNs are made up of many layers of un-linear procedures. they allow us to examine complicated patterns, and if used with big data, it can learn high-level concepts.

The research in (Batista et al., 2004) presented a respectable study of sampling methods. Several strategies of over and under sampling and dynamic / hybrid processes have been tested and examined carefully on thirteen datasets. While most of them had improved performance, in all experimental datasets there has been no method overwhelming others. The experiment findings have revealed that random over-sampling yielded great findings relative to more complicated approaches. Additional research argued that the improvements made by undersampling and oversampling approaches have a greater impact on the highly imbalanced distribution datasets (Japkowicz & Stephen, 2002).

PROPOSED TECHNIQUE

The proposed technique main focus is to generate new synthetic minority samples that lessen the difference in number between majority and minority data. Towards this goal, majority data samples are considered while generating the new minority data samples. The detailed steps of the proposed technique is described in Algorithm 1.

ALGORITHM 1: Proposed Technique (MODIFIED SMOTE)

1: **Function Modified_SMOTE (K,N,A,B,T,R)**
   Input: K,N,A,B,T,R where K:#neighbors, N:Percentage of required oversampling, A:Majority class samples, B:Minority class samples, T:# Minority samples, R:#iterations
   Output: Original Data + (N/100) * Minority class samples
2: **If** N<100 **then**
3: N=100
4: **End if**
5: **For** i=1 to T **do**
6: y← B(i)
7: Get k-nearest neighbors of y from A along with their distances
8: minA← the nearest neighbor of A to y
9: Get k-nearest neighbors of y from B along with their distances
10: minB← the nearest neighbor of B to y
11: x← Randomly select one of the nearest neighbors of y from B
12: dist_x_minA← calculate difference in distance between x and minA

```
13:     dist_x_minB← calculate difference in distance
        between x and minB
14:     While r<R do
15:         For j=1 to T do //loop to generate samples
16:             s(j) ← x +(minA - minB)*α
17:         End for
18:         dist_x_s←calculate distance between s
            and x
19:         If dist_x_s < dist_x_minA and
            dist_x_s > dist_x_minB then
20:             Go to 24
21:         End if
22:         r = r +1
23:     End while
24:     synth ← concatenate s with B
25: End for
26: Write file
27: End
```

The algorithm starts by considering samples in minority class. For each sample, we get the k-nearest neighbors of it from the majority class and also the k-nearest neighbors from minority class. We randomly select one of the minority neighbors and calculate the distance between it and both the nearest majority neighbor and the nearest minority neighbor multiplied by random number. We then generate the new synthetic sample depending on the randomly selected minority neighbor adding to it the difference in distance between the nearest majority neighbor and the nearest minority neighbor. Before writing the new generated sample to the generated samples file, we calculate the difference in distance between it and the randomly selected minority neighbor to make sure that it is already in the area between the closest majority neighbor and the closest minority neighbor.

PERFORMANCE EVALUATION

The objective of the evaluation is to prove the effectiveness of the proposed oversampling technique which considers the majority data samples while generating new minority data samples. Towards this goal the proposed technique is evaluated on different datasets over different classifiers against SMOTE method which is the standard oversampling approach in the literature. The following subsections describes the evaluation details.

Datasets

For our experiment, we used three numerical datasets which can be downloaded from (Alcalá-Fdez et al., 2011). The first dataset is Poker dataset which contains 1485 objects for 2 classes, the first class is 25 (values 1) and the second class is 1460 (value 0). Each object contains 11 attributes. The second dataset is Yeast which contains 514 objects for 2 classes, the first class is 51 (values 1) and the second class is 463 (value 0). Each object contains 9 attributes. The third dataset is Cleveland which contains 173 objects for 2 classes, the first class is 13 (values 1) and the second class is 163 (value 0). Each object contains 9 attributes For each dataset, cross validation procedure was used to split the data into training and testing sets. The number of cross validation folds was set to 10 folds.. Details of the datasets is summarized in Table 1.

Table 1: Number of Samples in each Dataset Before and After Oversampling

	Poker		Yeast		Cleveland	
	Before	After	Before	After	Before	After
Minority	25	650	51	459	13	169
Majority	1460	1460	463	463	160	160

Evaluation Method

Six evaluation matrices were used for performance evaluation, which are, accuracy, sensitivity, specificity, precision, f-score, and error. Sensitivity measures the proportion of actual positives that are correctly identified as such, while Specificity measures the proportion of actual negatives that are correctly identified as such. Sensitivity, therefore, quantifies the avoidance of false negatives and specificity does the same for false positives. Evaluation metrics were obtained after training of three different classifiers on the three datasets in different settings. Classifiers were trained once on the imbalanced datasets without applying oversampling and once trained on the datasets after applying traditional SMOTE algorithm for oversampling and finally trained on the datasets after applying the proposed technique for oversampling. Three different classifiers were used which are K-Nearest Neighbors, Fuzzy K-Nearest Neighbors and Support Vectors Machines classifiers. The KNN classifier takes only one parameters and the best results were when k= 10 While the FKNN classifier takes two parameters k and m, with k=10 and m=0.5.

Evaluation Results

Results of the proposed technique against SMOTE algorithm are summarized in Figures 2, 3, and 4.

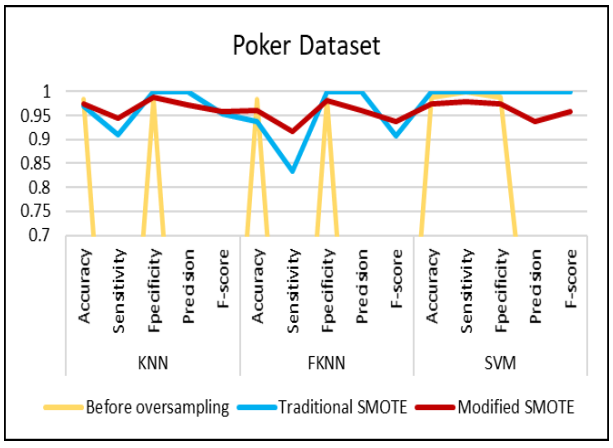

Figure 2: The Evaluation Metrics of Poker Dataset with the Three Classifiers

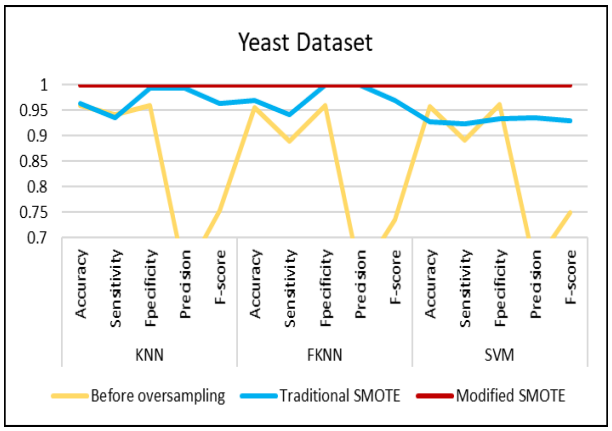

Figure 3: The Evaluation Metrics of Yeast Dataset with the three Classifiers

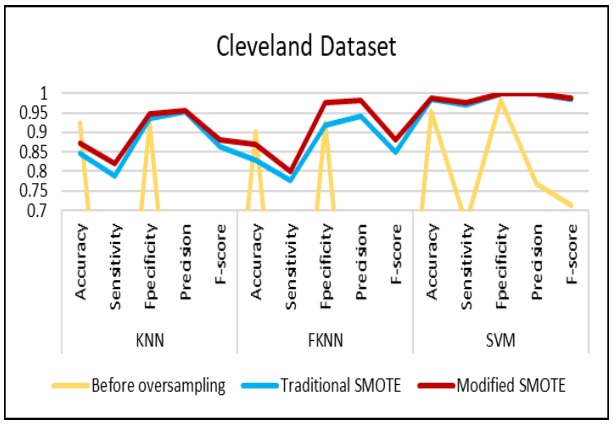

Figure 4: The Evaluation Metrics of Cleveland Dataset with the Three Classifiers

We can see that the proposed technique got almost better performance in all three datasets with different classifiers. We may notice that before oversampling, the classifiers showed high value in the accuracy metric as you see in Figure 2,3 and 4. Generally speaking, the accuracy metric measures the fraction of all instances that are correctly categorized. But here, this metric value is fake if used alone for evaluation (Akosa, 2017). As due to the imbalance of data, the classifiers tend to get all majority samples as correct and all minority samples as incorrect. The other metrics especially precision which is low explains this.

The outperformance of the proposed technique against SMOTE algorithm may be related to the difference between how the two methods work. While SMOTE algorithm generates new samples in the space of the minority data space, the Modified SMOTE algorithm generates new samples in the space separating minority and majority data. Those new samples were able to explain the difference between majority and minority points in a better way and consequently lead to better classifications results after using them during training.

CONCLUSION AND FUTURE WORK

The paper explored the nature of the imbalanced data and its current real-life applications. We provided a taxonomy for the solutions found in the literature. Then, we presented a comparative study for the efforts done with the aim of addressing the challenge of the classification of imbalanced data. At last, we introduced our proposed technique for handling the imbalanced data problem along with our experiment which showed a noticeable higher performance results. In future, we aim to further apply it to categorical datasets as we applied it with only numerical ones. Another direction is to apply our approach to multiclass datasets.

REFERENCES

Abdi, L. Hashemi, S. (2015). To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE transactions on Knowledge and Data Engineering, 28*(1), 238-251.

Akbani, R., et al. (2004). *Applying support vector machines to imbalanced datasets.* Paper presented at the European conference on machine learning.

Akosa, J. (2017). *Predictive accuracy: a misleading performance measure for highly imbalanced data.* Paper presented at the Proceedings of the SAS Global Forum.

Alcalá-Fdez, J., et al. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing, 17*.

Batista, G. E., et al. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter, 6*(1), 20-29.

Bunkhumpornpat, C. Sinapiromsaran, K. (2017). DBMUTE: density-based majority under-sampling technique. *Knowledge and Information Systems, 50*(3), 827-850.

Cai, T., et al. (2018). Breast cancer diagnosis using imbalanced learning and ensemble method. *Applied and Computational Mathematics, 7*(3), 146-154.

Cheng, F., et al. (2017). Large cost-sensitive margin distribution machine for imbalanced data classification. *Neurocomputing, 224*, 45-57.

Chu, X., et al. (2016). *Data cleaning: Overview and emerging challenges.* Paper presented at the Proceedings of the 2016 International Conference on Management of Data.

Devi, D. Purkayastha, B. (2017). Redundancy-driven modified Tomek-link based undersampling: a solution to class imbalance. *Pattern Recognition Letters, 93*, 3-12.

Fernández, A., et al. (2017). An insight into imbalanced big data classification: outcomes and challenges. *Complex & Intelligent Systems, 3*(2), 105-120.

Geurts, P., et al. (2006). Extremely randomized trees. *Machine learning, 63*(1), 3-42.

Ha, J. Lee, J.-S. (2016). *A new under-sampling method using genetic algorithm for imbalanced data classification.* Paper presented at the Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication.

Herland, M., et al. (2018). Big data fraud detection using multiple medicare data sources. *Journal of Big Data, 5*(1), 29.

Hu, Y., et al. (2015). An improved algorithm for imbalanced data and small sample size classification. *Journal of Data Analysis and Information Processing, 3*(03), 27.

Japkowicz, N. Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis, 6*(5), 429-449.

Khan, S. H., et al. (2017). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems, 29*(8), 3573-3587.

LeCun, Y., et al. (2015). Deep learning. *nature, 521*(7553), 436-444.

Lin, W. C., et al. (2017). Clustering-based undersampling in class-imbalanced data. *Information Sciences, 409*, 17-26.

Liu, A., et al. (2007). *Generative Oversampling for Mining Imbalanced Datasets.* Paper presented at the DMIN.

López, V., et al. (2012). Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. *Expert Systems with Applications, 39*(7), 6585-6608.

Ng, W. W., et al. (2014). Diversified sensitivity-based undersampling for imbalance classification problems. *IEEE transactions on cybernetics, 45*(11), 2402-2412.

Pant, H. Srivastava, R. (2015). A survey on feature selection methods for imbalanced datasets. *International Journal of Computer Engineering and Applications, 9*(2).

Sáez, J. A., et al. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition, 57*, 164-178.

Zhang, C., et al. (2018). A cost-sensitive deep belief network for imbalanced classification. *IEEE transactions on neural networks and learning systems, 30*(1), 109-122.