Applying Ensemble Learning Approach in Imbalanced Data Learning of Accounting Fraud Detection

Shuqi Mo (ID: 17318086)
Sun Yat-sen University
Machine Leaerning and Data Mining Course
2020-2021 Fall Term

Abstract

Accounting fraud is a worldwide problem and is difficult to detect. Considering the accounting fraud detection as a supervised learning task, machine learning could be an alternative. Previous works used financial variables and linguistic variables as input and the output is 1 for fraudulent and 0 for nonfraudulent. Imbalanced data learning problem is challenging when applying traditional machine learning algorithms to this task. This paper applied a cluster-based under-sampling technique *CUSBoost* to alleviate the imbalanced data set problem. Firstly, CUSBoost performs under-sampling based on the k-means clustering result of majority instances. Then, CUSBoost conducts boosting algorithm to train a prediction model. Experiment results demonstrate that CUSBoost improves both AUC and NDCG@k evaluation metrics against the state-of-the-art fraud prediction model RUSBoost, indicating that CUSBoost will an effective and efficient tool for fraud prediction.

1 Introduction

Financial statements reveal the financial health of a company and thus provide valuable information for the market to trade. Accounting fraud can be defined as material omissions or misrepresentations resulting from an intentional failure to report financial information in accordance with generally accepted accounting principles (Nguyen 2010). If not detected and prevented on a timely basis, it can cause significant harm to the stakeholders of fraudulent firms as well as the stakeholders of many nonfraudulent firms indirectly (Hung, Wong, and Zhang 2015). In practice, accounting fraud is usually detected by auditors. Professional auditors conduct specific audit procedures based on accounting principles (i.e. IFRS, PRC GAAP, US GAAP) and their experience on accounting estimates. Both the complexity of audit precedures and subjectivity of estimation make accounting fraud difficult to detect. Hence, efficient and effective methods of accounting fraud detection would offer significant value to auditors and the whole market.

Considering the accounting fraud detection as a supervised learning task, machine learning could be an alternative for this problem. After inputting the publicly provided financial statement records and fraud records as the training data, machine learning algorithm could automatically learn a model that maps from financial statement records to fraudulent-or-nonfradulent issues. However, the main challenge of applying machine learning in this scenario is the highly *class-imbalanced problem*, i.e. the real-world fraudulent companies are outnumbered by the nonfraudulent companies. Traditional machine learning algorithms will be inefficient in the class-imbalanced problem due to a lack of interest addressed to the rare class. The trained model will have inappropriate inductive bias and tend to favor classifying examples as belonging to the majority class (Weiss 2004).

To alleviate the problem of class imbalance, previous researches proposed methods into three categories: (1) data-based methods, (2) algorithm-based methods and (3) hybrid methods. Data-based methods use data sampling techniques to conduct a balanced data set, or assign more misclassification cost to the minority instances in the objective functions. Algorithm-based methods (e.g. ensemble learning) modify the existing algorithms to reduce their sensitiveness to the class imbalance when learning from the imbalanced data. Hybrid method is a combination of data-based method and algorithm-based method. Under the categories discussed above, RUSBoost (Seiffert et al. 2009) is hybrid method which proposes random under-sampling with ensemble learning algorithm. Bao et al. applied the RUSBoost approach in publicly traded U.S. firms data sets and proved that this approach could perform better than their benchmarks of SVM and logistic regression evaluated by AUC and NDCG@k (Bao et al. 2020). However, the under-sampling method with random sampling of the majority class might suffer from the loss of potentially useful training instances, which may limit their model performance.

Inspired by this, I applied another model, *CUSBoost* (Rayhan et al. 2017), which prevents this potential information loss by cluster-based random under-sampling. Both RUSBoost and CUSBoost belong to hybrid methods that combining under-sampling with ensemble learning algorithm. The difference between RUSBoost and CUSBoost is that CUSBoost does not randomly select a portion of majority instances directly. Instead, CUSBoost firstly clusters

the majority instances into several clusters and then randomly select a portion of instances from each cluster. The intuition of CUSBoost is that it considers examples from all subspace of the majority instances. I tested the performance of CUSBoost with RUSBoost, the best machine learning algorithm in fraud detection so far, on the dataset provided by Bao et al.. Based on the experimental result, the CUSBoost method outperforms RUSBoost at most cases. Furthermore, supplement analysis is conducted to prove that the original dataset is clusterable, which validates the correctness of the CUSBoost method in the specific application of fraud detection.

The remainder of this paper is organized as follows. Section 2 presents related works on using machine learning to detect fraud. Section 3 describes the imbalanced data learning methods and the CUSBoost algorithm. Experiment details and result are provided in Section 4. Section 5 is the conclusion.

2 Related Work

Accounting fraud detection can be considered as a supervised learning task, or a binary classification problem more specifically, under the categories of machine learning. Regarding the input variables utilized, previous studies employed two kinds of variables: (1) financial variables and (2) linguistic variables.

Financial Variables Financial variables are usually financial indicators taken from financial statements. This is due to the fact that the unusual values of financial variables may indicate the need to meet targets or hide losses (Lin et al. 2015). This incentive to commit fraudulent practice increases the potential for fraud. Therefore, the financial variables used in prior studies covered all aspects of firms' financial performances, such as profitability, activity, asset structure, liquidity, business situation, leverage, and market value (Dechow et al. 2011). Based on the quantitative financial statement and stock market variables as predictive factors, machine learning algorithms can train a detection model to predict that a company is fraudulent-or-nonfraudulent. Previous works applied three mainstream machine learning algorithms in the specific task of accounting fraud detection: logistic regression, support vector machine (SVM) and ensemble learning. Ratio-based logistic regression (Dechow et al. 2011) uses financial ratios as predictors, which the ratios are often identified by human experts based on theories. Rather than using the financial ratios identified by human experts alone, another fraud detection model based on SVM with a financial kernel (Cecchini et al. 2010) that maps raw financial data into a broader set of ratios within the same year and changes in ratios across different years. The latest work in this area addressed the class-imbalanced problem and implemented the RUSBoost algorithm based on raw financial data items instead of financial ratios (Bao et al. 2020). Bao et al. (2020) find that the RUSBoost outperforms the traditional ratio-based prediction models.

Linguistic Variables Linguistic variables are extracted from textual analysis and text mining methods. Previous studies suggested that firm-related textual documents may contain misleading statements. MD&A section of annual reports is usually used as the input text, because it provides investors with the opportunity to receive qualitative information on a firm's performance and prospects from the manager's perspective. According to previous works, fraudsters are more likely to use negative and uncertain words (Newman et al. 2003). This has theoretical foundations in an individual's nonverbal behavior during deception, suggesting that deceivers often include statements indicating aversion or negative mood. In addition to measuring the proportion of negative and uncertain words, a recent research applied the LDA topic modeling to quantify the thematic content of annual report filings and the attention devoted to each topic (Brown, Crowley, and Elliott 2020).

This paper will focus on accounting fraud detection by financial variables and address the imbalanced data problem when training the model.

3 Imbalanced Data Learning Method

The solution to imbalanced data sets can be divided into data-based, algorithm-based and hybrid categories. The data-based methods change the distribution of the imbalanced data sets, and then the balanced data sets are provided to the learner to improve the detection rate of minority class. As one of the representative of the algorithm-based methods, ensemble learning can modify the existing data mining algorithms (or base learners) to resolve the imbalance problem. The hybrid methods combine data-based methods with algorithm-based methods.

3.1 Data-based Imbalance Learning

The data-based methods proposed different forms of re-sampling methods (Batista, Prati, and Monard 2004). The simplest re-sampling methods are random under-sampling and random over-sampling. The former randomly takes away some majority instances, while the latter augments the minority instances by exactly duplicating the examples of the minority class. However, random under-sampling may cause information loss when removing the majority instances. Random over-sampling may make the decision regions of the learner smaller and more specific, thus cause the learner to over-fit. Under-sampling methods generally work better than over-sampling methods as long as the imbalance ratio of the dataset is not very high (Farid, Nowé, and Manderick 2016).

Another data-based method is cost sensitive learning. This method tackles the class imbalance problem by assigning more misclassification cost to minority class instances for the underlying classifier (Rayhan et al. 2017). The intuition here is to make the classifier pay more attention to the misclassified minority instances. However, how to assign the cost is a difficult question when applying this method in practice.

3.2 Algorithm-based: Ensemble Learning

The algorithm-based methods operate on the algorithms other than the data sets. Ensemble learning methods com-

bine multiple base learners which may be the same or different. This usually increases the predictive capability of the individual base learners, thus making it adaptive to different data sets. Bagging and boosting are two basic ensemble learning algorithms. Bagging creates multiple sub-sets of the original dataset through sampling with replacement or without replacement. These sub-sets are used by the base learners while considering each instance with equal weight. The output of the individual base learners is considered as votes to determine the final prediction.

Boosting is similar to bagging in that it combines multiple base learners to obtain a result based on voting technique. The difference lies in that boosting assigns weight to instances according to how hard they are to classify and assigns weight to base learners according to how well they are to classify. Adaboost (Freund, Schapire, and others 1996) is a standard boosting algorithm, which increases the weights of misclassified examples and decreases those correctly classified using the same proportion.

3.3 Hybrid method: Sampling with Boosting

The hybrid methods can combine the data-based methods with algorithm-based methods. One of the combinations is to combine sampling with boosting, which the former belongs to data-based and the latter belongs to algorithm-based. Firstly, the training data set will be divided into majority instances and minority instances. Secondly, sampling methods, i.e. under-sampling or over-sampling, will be applied to reconstruct a balanced training set. Finally, boosting algorithm will train a model based on its base learners and the balanced training set. This model can be used to complete the classification or regression tasks on the testing data set.

SMOTE (Synthetic Minority Over-sampling Technique) algorithm (Chawla et al. 2002) creates synthetic minority class instances rather than by over-sampling with replacement. SMOTEBoost (Chawla et al. 2003) combines this intelligent oversampling technique (SMOTE) with AdaBoost, resulting in a highly effective hybrid approach to learning from imbalanced data. Another hybrid algorithm is RUSBoost (Seiffert et al. 2009), which applies random under-sampling (RUS) with AdaBoost algorithm. RUS randomly removes the majority class instances to form a balanced data. CUSBoost (Rayhan et al. 2017) is based on combination of cluster-based sampling and Adaboost algorithm. It is similar to RUSBoost and SMOTEBoost with the critical difference occuring in the sampling technique. SMOTEBoost uses SMOTE method to oversample the minority class instances, while RUSBoost uses random under-sampling on the majority class. In comparison, CUSBoost uses cluster-based under-sampling instead of random under-sampling on the majority instances. The key intuition here is that CUSBoost uses clustering to consider every subspace of the majority instances and ensures that the boosting algorithm can make use of examples from all regions of the data set. Firstly, CUSBoost separates the majority and minority instances from the training data set. Secondly, CUSBoost clusters the majority instances into k clusters using k-means clustering algorithm. Finally, CUSBoost performs random under-sampling on each of the created cluster, reconstructs a balanced training set, and applies the AdaBoost algorithm.

Algorithm 1 CUSBoost Algorithm

Given: Imbalanced data D, k clusters, number of iterations T, $WeakLearner$ C4.5 decision tree induction algorithm.
CUS: Use cluster-based under-sampling to create balanced data set:

1: separate the data set D to majority instances and minority instances;
2: use k-means algorithm to cluster the majority instances into k clusters;
3: apply random under-sampling on each cluster;

Boosting: Train an ensemble model based on the balanced dataset:

1: initialize weight, $x_i \in D$ to $\frac{1}{d}$;
2: **for** i = 1 to T **do**
3: create balanced dataset D_i with distribution D by cluster-based under-sampling;
4: derive a weak learner M_i from D_i employing C4.5 algorithm;
5: compute the error rate of M_i, $error(M_i)$ in Eq. 1;
6: **if** $error(M_i) \geq 0.5$ **then**
7: go back to step 3 and try again;
8: **end if**
9: **for** each $x_i \in D_i$ that correctly classified **do**
10: multiply weight of x_i by α_i in Eq. 2;
11: **end for**
12: normalise the weight of each instances, x_i;
13: **end for**

Specifically, CUSBoost considers a series of decision trees using C4.5 algorithm and combines the votes of each individual tree to classify new instances in the boosting process. Each instance is initialized with an equal weight $\frac{1}{N}$ at the first stage, where N is the total number of training instances. The weights of instances are adjusted according to how they were classified. If an instance was correctly classified then its weight is decreased, or if misclassified then its weight is increased. The weight of an instance reflects how difficult it is to classify. To compute the $errorrate$ of the weak leaner M_i, the weights of misclassified instances in D_i are summed up as shown in Eq. 1. If an instance x_i is misclassified, then $err(x_i)$ equals to one. Otherwise, $err(x_i)$ equals to zero when x_i is correctly classified.

$$error(M_i) = \sum_{i=1}^{N} w_i * err(x_i), \tag{1}$$

If an instance x_i in the ith iteration is correctly classified, its weight is multiplied by α_i as shown in Eq. 2. For normalization, the weights of all instances are multiplied by the sum of old weights and then divided by the sum of new weights. As a result, the weights of misclassified instances are increased and the weights of correctly classified instances are decreased. If the $errorrate$ of weak learner M_i exceeds 0.5, it will be abandoned and a new M_i will be derived by generating a new sub-data set D_i. The CUSBoost algorithm is

summarized in Algorithm 1.

$$\alpha_i = \frac{error(M_i)}{1 - error(M_i)}, \tag{2}$$

4 Experiments

4.1 Data set and Baseline

This paper used the SOTA fraud prediction model RUSBoost (Bao et al. 2020) from related work as baseline. For better comparison, this paper also used the same data source as Bao et al. (2020). Bao et al. (2020) used raw financial data items taken directly from financial statements as fraud predictors. The list of raw financial data items is selected based on Cecchini et al. (2010) and Dechow et al. (2011). Their accounting fraud sample comes from the SEC's AAERs provided by the University of California-Berkeley Center for Financial Reporting and Management (CFRM). According to this data set, there were 1171 detected fraudulent firm-years in total over 1979-2014, but the frequency of detected fraud is very low, typically less than 1% of all firms per year. The rarity of detected accounting fraud highlights the ongoing challenge of fraud prediction based on imbalanced data learning.

4.2 Evaluation Metrics

Following Bao et al. (2020), this paper uses AUC and NDCG@k as performance evaluation metrics. AUC is equivalent to the probability that a randomly chosen fraud will be ranked higher by a classifier than will a randomly chosen nonfraud observation. The AUC for random guesses is 0.50. Therefore, any reasonable fraud prediction model must have an AUC higher than 0.50. AUC can evaluate the effectiveness of the fraud prediction model, while the NDCG@k is used for the consideration that regulators and other monitors seek to investigate the smallest number of observations with the highest predicted likelihood of fraud. Intuitively, NDCG@k assesses the ability of a fraud prediction model to identify actual fraud by picking the top k observations in a test year that have the highest predicted probability of fraud. In the experiment, this paper pick a k that equals the top 1% of the observations.

4.3 Results and Analysis

This paper proposes two research questions: (**RQ1**) Can CUSBoost outperform state-of-the-art fraud prediction RUSBoost method? (**RQ2**) Do the cluster-based under-sampling idea contribute to the better performance of CUSBoost?

In order to answer the research questions, this paper reports on two experiment results. The first experiment aims to evaluate the performance of CUSBoost and RUSBoost in the same experiment setup (i.e. use same data set and same hyperparameter). This paper repeated the experiment in Bao et al. (2020) which using the year 1991-2001 data as training set and year 2003 data as testing set. According to the result shown in Table 1, CUSBoost can outperform RUSBoost in most hyperparameter settings evaluated by AUC and NDCG@k, especially in the perspective of NDCG@k.

Models	Depth	Iterations	AUC	NDCG@k
RUSBoost	2	20	0.23	0.00
	2	30	0.72	0.03
	2	40	0.64	0.05
	12	20	0.72	0.01
	12	30	0.73	0.06
	12	40	0.74	0.08
CUSBoost	2	20	**0.71**	**0.07**
	2	30	0.67	**0.07**
	2	40	**0.73**	**0.06**
	12	20	**0.75**	**0.08**
	12	30	**0.76**	**0.15**
	12	40	**0.76**	0.06

Table 1: Evaluation results of RUSBoost and CUSBoost under same hyperparameter setting. Depth: the maximum depth of C4.5 decision tree. Iterations: the iteration times of the boosting algorithm.

Therefore, it is believed that CUSBoost can achieve the state-of-the-art performance in fraud prediction and has the ability to predict accounting fraud both *effectively* and *efficiently*.

The second experiment aims to analyze the contribution of cluster-based under-sampling, which is the core idea of CUSBoost. Rayhan et al. (2017) mentioned that CUSBoost does not always achieve better performance than RUSBoost. Theoretically, CUSBoost will perform best when the data set is highly cluster-able. Therefore, the second experiment is conducted to evaluate whether the data set of fraud prediction is highly cluster-able. The Silhouette Coefficient was created as a measure of cluster density and separation (Rousseeuw 1987). In the second experiment, the Silhouette Coefficient is calculated before every iteration of the boosting algorithm. According to the experiment result, the average values of Silhouette Coefficient on different hyperparameter settings can exceed 0.76, which indicates that the data set of fraud prediction is highly cluster-able and the cluster-based under-sampling idea contributes to the better performance of CUSBoost in Table 1.

5 Conclusion

Previous works addressed the data imbalanced learning problem in accounting fraud detection. This paper applied a hybrid method CUSBoost, which combines the cluster-based sampling method with boosting algorithm, to the task of fraud prediction. The experiment result shows that CUSBoost can achieve state-of-the-art performance in fraud prediction evaluated by AUC and NDCG@k. In addition, the data set of fraud prediction is highly-clusterable due to the high Silhouette Coefficient, which proves that CUSBoost can perform well in the scenario of fraud prediction. Therefore, CUSBoost is an effective and efficient tool in accounting fraud detection.

References

Bao, Y.; Ke, B.; Li, B.; Yu, Y. J.; and Zhang, J. 2020. Detecting accounting fraud in publicly traded us firms using a machine learning approach. *Journal of Accounting Research* 58(1):199–235.

Batista, G. E.; Prati, R. C.; and Monard, M. C. 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter* 6(1):20–29.

Brown, N. C.; Crowley, R. M.; and Elliott, W. B. 2020. What are you saying? using topic to detect financial misreporting. *Journal of Accounting Research* 58(1):237–291.

Cecchini, M.; Aytug, H.; Koehler, G. J.; and Pathak, P. 2010. Detecting management fraud in public companies. *Management Science* 56(7):1146–1160.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16:321–357.

Chawla, N. V.; Lazarevic, A.; Hall, L. O.; and Bowyer, K. W. 2003. Smoteboost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, 107–119. Springer.

Dechow, P. M.; Ge, W.; Larson, C. R.; and Sloan, R. G. 2011. Predicting material accounting misstatements. *Contemporary accounting research* 28(1):17–82.

Farid, D. M.; Nowé, A.; and Manderick, B. 2016. A new data balancing method for classifying multi-class imbalanced genomic data. In *25th Belgian-Dutch Conference on Machine Learning (Benelearn)*, 1–2.

Freund, Y.; Schapire, R. E.; et al. 1996. Experiments with a new boosting algorithm. In *icml*, volume 96, 148–156. Citeseer.

Hung, M.; Wong, T.; and Zhang, F. 2015. The value of political ties versus market credibility: Evidence from corporate scandals in china. *Contemporary Accounting Research* 32(4):1641–1675.

Lin, C.-C.; Chiu, A.-A.; Huang, S. Y.; and Yen, D. C. 2015. Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments. *Knowledge-Based Systems* 89:459–470.

Newman, M. L.; Pennebaker, J. W.; Berry, D. S.; and Richards, J. M. 2003. Lying words: Predicting deception from linguistic styles. *Pers Soc Psychol Bull* 29(5):665–675.

Nguyen, K. 2010. *Financial statement fraud: Motives, methods, cases and detection*. Universal-Publishers.

Rayhan, F.; Ahmed, S.; Mahbub, A.; Jani, R.; Shatabda, S.; and Farid, D. M. 2017. Cusboost: Cluster-based undersampling with boosting for imbalanced classification. In *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, 1–5. IEEE.

Rousseeuw, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20:53–65.

Seiffert, C.; Khoshgoftaar, T. M.; Van Hulse, J.; and Napolitano, A. 2009. Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 40(1):185–197.

Weiss, G. M. 2004. Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter* 6(1):7–19.