Handling Class Imbalance in Customer Behavior Prediction

Nengbao Liu, Wei Lee Woon, Zeyar Aung, Afshin Afshari

Institute Center for Smart and Sustainable Systems (iSmart)
Masdar Institute of Science and Technology
Abu Dhabi, UAE
Email: {nliu, wwoon, zaung, aafshari}@masdar.ac.ae

***Abstract*—Class imbalance is a common problem in real world applications and it affects significantly the prediction accuracy. In this study, investigation on better handling class imbalance problem in customer behavior prediction is performed. Using a more appropriate evaluation metric (AUC), we investigated the increase of performance for under-sampling and two machine learning algorithms (weight Random Forests and RUSBoost) against a benchmark case of just using Random Forests. Results show that under-sampling is the most effective way to deal with class imbalance. RUSBoost, as a specific algorithm designed to deal with class imbalance problem, is also effective but not as good as under-sampling. Weighted Random Forests, as a cost-sensitive learner, only improves the performance of appetency classification problem out of three classification problems.**

***Keywords*—Class imbalance, Random forests, RUSBoost, Under-sampling, Customer behavior, Prediction**

I. Introduction

In the real world applications, there are various problems associated with the data, which make some data mining algorithms lose their power. One common problem is the class imbalance problem, which means that one class is represented by a large number of examples while the other is represented by only a few [1]. In the context of data mining, the minority class is often of great interest and value. However, this class imbalance problem causes the minority class much harder to identify and also many algorithms have great difficulty dealing with this problem [2].

In the presence of class imbalance, a number of problems may arise. Firstly, commonly used evaluation metric may not be suitable any more. For example, classification accuracy, which is the percentage of correctly classified examples among all examples, is biased towards to the majority class as the majority class has more impact on accuracy than minority class. This is also confirmed by an empirical study by Wesis and Provost [3]. So a more appropriate evaluation metric is in great need which would take class imbalance into account and better guide the data mining process and interpret the results. Secondly, algorithms without considering class imbalance problem may fail when classifying the minority class. This is of great importance when the minority class is the problem of interest, which is often the case in the context of data mining. As the same problem with accuracy, those algorithms would be biased towards the majority class. In the literature, various techniques have been proposed to particularly deal with this problem.

One application where class imbalance is a big problem is in the prediction of customer behavior. Very often, the types of behavior that are of particular interest are a lot less common. For example, in normal situations the number of customers who would continue one service from a company are much more than that of those who would change the service to another company. Customer behavior prediction is one of the most important issues for companies and one of the core parts in Customer Relationship Management (CRM). Companies predict customer behaviors so that they can take corresponding actions, such as if a company predicts a customer is prone to churn, then the company can take actions to retain this customer before he or she leaves the company. Many data mining techniques have been applied on this issue while those with good performance are the ones companies need.

While many techniques have been proposed in the literature [2], it still seems that there is a shortage of works which compare the relative efficacies of these methods on an industrial dataset. In this study we hope to address this apparent shortcoming. Our study has the following key objectives and characteristics:

- Three commonly used methods will be considered: weight Random Forests, RUSBoost and under-sampling.
- Use of the AUC as the evaluation metric - this is a good choice as it does not emphasize one class over the other [4].
- Application on CRM data, which is of particular interest to many Big Data players.

We believe that there is currently no other work in the literature which combines all three elements.

The rest of the report is organized as follows: in section II-A, we give a general description of the dataset used in this study and some data preprocessing techniques. After that, the methods used in this study are presented in section II-B. Results and discussions are included in section III. Finally the conclusion part is in section IV.

II. Experimental Methodology

A. Data Description and Preprocessing

In this subsection, firstly a description of the dataset used in this study is presented. Then we analyze some problems in the dataset and describe their corresponding data preprocessing techniques applied in this study.

1) Data Description: The dataset used in this paper is from KDD Cup 2009 [5], which is a telecommunication industry dataset from Orange, a telecom company in France. There are two versions of datasets, big dataset and small dataset. The small dataset has several special characteristics: a large number of training examples (50,000) with a large number of missing values (about 60%), imbalanced class proportions (fewer than 10% of the examples of the positive class), noisy data, presence of categorical variables with many different values [6]. The big dataset contains much more variables, but the number of examples is the same as the small dataset. And it also has same characteristics as listed above. In addition, the dataset has other important properties that should be considered, and one of these properties is that: many of its features are irrelevant or redundant. The second thing is that the datasets are heterogeneous, in which it has numerical and categorical features, and the number of different categories varies on different categorical features. These features may range from two to more than ten thousands [7]. In this study, only the training set of the small dataset is used and it is split into two parts with 70% and 30% respectively, which are used to train and test classifiers.

2) Missing Value: As there are large number of missing data [6], proper dealing with them is very important and can significantly affect the performance of classifiers. For missing values, one possible solution is to remove the corresponding records. Another good way is to fit the missing values with their most probable values, in which some models, such as regression, decision tree may be used [8]. This way has an advantage that it utilizes all the information present in the data. In this study, methods to handle missing data are as follows: features with large amount of missing values are discarded, examples with large amount of missing features are discarded too. Missing categorical values will be set as a standalone value, missing numerical values will be fit with their mean, but binary features to indicate the missing status are added, that is, a 0/1 indicator is added for any feature with at least one missing value [9]. It shows that this approach is very effective and improves the test performance considerably [7].

3) Imbalanced Dataset: The fraction of positive/negative examples is significantly skewed and thus normal prediction models would be biased to favor those classes with more examples. The proportion of examples in different classes in the training dataset is as below [6]:

- Churn problem: 7.3% positive examples (3672/50000 on train).
- Appetency problem: 1.8% positive examples (890/50000 on train).
- Up-selling problem: 7.4% positive examples (3682/50000 on train).

In the literature, there are several popular ways to deal with this imbalance class problem. One is to use cost-sensitive learning, which can exploit the fact that the value of correctly identifying the positive (rare) class outweighs the value of correctly identifying the common class [4]. In this study, weighted Random Forests are utilized to deal with class imbalance problem. In addition, RUSBoost, a boosting algorithm which was designed to particularly deal with imbalanced data, is also adopted to compare its performance against weighted Random Forests.

Another approach is to use sampling method, whose basic idea is to eliminate or minimize rarity by altering the distribution of training examples [4]. Considering the simplicity and effectiveness of sampling method, under-sampling method is used to deal with the imbalanced dataset, which are also used by those papers [4] [10].

4) Categorical Features: Another challenge in this dataset is that there are categorical features which contains thousands of values for each. One possible solution is to convert each categorical feature to a single integer feature, where each integer value corresponds to a single categorical value [11]. However, this approach has an obvious drawback which introduces certain ordering into categorical features and later it failed to give good performance. Another way to deal with this challenge found in the literature contains transforming each categorical feature to several binary ones. A binary feature corresponds to a possible value of the categorical feature [7]. However, this approach would generate a substantially large number of additional features, so one possible modification is to encode only those most common values of each categorical feature [9]. In this study, as both Random Forests and RUSBoost can deal with categorical features directly, we did not transform categorical features to binary features. However, we applied the following transformation: for categorical features with many categorical values, the most 30 common categorical values are retained while the rest categorical values are aggregated into a single categorical value.

Another problem in the dataset is that there are some numerical features which contain only a few unique values. Keeping them as numerical features would be problematic as they would introduce some ordering which may do not exist. So in this study, these features are treated as categorical features.

B. Proposed Methods

In this subsection, the methods used in this study are presented, which are weighted Random Forests, RUSBoost and under-sampling.

1) Weighted Random Forests: When one uses Random Forests classifier to perform classification on imbalanced data, it would have the same problem as many other algorithms, which tends to be biased towards the majority class. One way to improve it is to assign a larger penalty for misclassifying the minority class. Weights are assigned to each class with the minority class having a higher weight (i.e. higher misclassification cost). Class weights are essential parameters to tune to get the desired classifier configuration.

2) RUSBoost: RUSBoost was firstly introduced by Seiffert et al. in 2008 to deal with skewed data [12], and becomes a popular approach for imbalanced classification problems. The effectiveness of handling class imbalance behind RUSBoost is that it combines two traditional methods, sampling and boosting algorithm, which are all effective to handle imbalance problems. RUSBoost is based on another similar algorithm SMOTEBoost but replaces its intelligent sampling with random selection. The random example selection is proved to outperform the SMOTEBoost. RUSBoost is adopted in our study to address the highly imbalanced dataset problem, which is a common characteristic in many real world applications.

3) Under-sampling: One of the most common ways to deal with class imbalance is sampling. The basic idea behind under-sampling is to eliminate or minimize rarity by altering the distribution of training examples [4]. In this study, under-sampling is used, which eliminates majority class examples in order to get the classes balanced. However, this under-sampling method has some drawbacks. One is that under-sampling potentially drops useful majority class examples and thus may degrade the classifier performance. However, usually it's reported to be effective in dealing with class imbalance problem [4] and its effectiveness is tested in this study.

III. Results and Discussion

In this section, firstly the evaluation criteria used in this paper is described. After that, results from the benchmark case of just using normal Random Forests and three other methods, which are under-sampling, weighted Random Forests, and RUSBoost will be presented, followed with some comparisons and discussion.

A. Evaluation Criteria: AUC

Receiver Operating Characteristic (ROC) is a good way to evaluate a classifier as it is independent on the threshold. Each binary classifier is represented by a point (1-specificity, sensitivity) and by varying the threshold of the probabilistic classifiers, a set of points can be obtained to construct the ROC curve [4]. However, ROC technique is sometimes hard to use in real world when comparing between different modeling techniques or different feature spaces. To overcome this problem, Area under ROC curve (AUC) is proposed as a measurement of the overall quality of a classifier. A random classifier would have an AUC of 0.5 while a perfect classifier would have an AUC of 1. So the AUC value lies between 0.5 and 1, preferably close to 1. Statistically, AUC is equivalent to the probability that a random positive example has a higher assigned score than random negative example.

B. Benchmark Case: Random Forests

For classification using Random Forests, two steps are performed, 1) identify the optimal number of trees and optimal minimum number of examples in the leaf nodes, which determines the depth of the decision trees. Those two values are empirically chosen by examining values in a certain range. 2) after step 1, a benchmark base case is obtained with the optimal depth of the trees and optimal number of trees. Those two steps are applied for all three classification problems, appetency, churn and up-selling. The base cases for the three classification problems and their respective AUC values are showed in Table I

C. Weighted Random Forests

After obtaining the benchmark case, the performance of other methods dealing with class imbalance can be compared against the benchmark case. One method is to use weighted Random Forests, which is to vary the misclassification cost for the three classification problems. Table I shows the optimal AUC and their respective misclassification costs. The misclassification cost here stands for the cost of misclassifying minority class over misclassifying majority class, which is the cost of misclassifying class with fewer examples (e.g. appetency) to the class with more examples (e.g. non-appetency) over the cost of misclassifying the other way round.

D. Random Forests with Under-sampling

In this scenario, the data are under-sampled to have equal number of examples for both classes to deal with the imbalance problem and the algorithm used is normal Random Forests.

As the same with benchmark case, the first step is to identify the optimal minimum number of examples in the leaf nodes and the optimal number of trees. Those two values are empirically chosen by examining values in a certain range.

Then the base cases with under-sampling for the three classification problems are obtained and their respective AUC values are showed in Table I.

E. RUSBoost

RUSBoost is designed particularly to deal with class imbalance problem. For RUSBoost, three steps are performed, 1) identify the optimal minimum number of examples in the leaf nodes and optimal number of trees. Note that those two values are empirically chosen by examining values in a certain range. 2) after step 1, a base case is obtained with the optimal depth of the tree and optimal number of trees. 3) varying the learning rate of the algorithm to get the best cases. Table I shows the best AUCs with corresponding learning rates.

F. Comparison and Discussion

Table I shows the comparison of benchmark case with other three methods to deal with class imbalance problem.

TABLE I. Comparison of Benchmark Case with Other Three Methods

Methods	Appetency	Churn	Up-selling
Benchmark Case	0.6423	0.6210	0.7065
Weighted Random Forests	**0.7022**	0.6242	0.7067
Random Forests with Under-sampling	**0.7020**	**0.6559**	**0.7417**
RUSBoost	0.6866	**0.6580**	0.7229

From Table I, we can generally see that all three methods improve classification performance against the benchmark case. However, the improvement is different for each method. They are summarized as follows:

- Weighted Random Forests: it only increases AUC for appetency, while for other two classification problems, it does not improve much.

- Random Forests with under-sampling: it greatly improves all three classification problems.
- RUSBoost: it also improves all three classification performance greatly, but the improvement is less than that of Random Forests with under-sampling.

From Table I, we can generally conclude that under-sampling is the most effective way to deal with class imbalance, which improves AUC most. RUSBoost is the second best method and weighted Random Forests is the least effective method.

IV. Conclusion

The dataset used in this study posed several significant challenges and class imbalance is one of them. But this gives us a great chance to investigate the impact of proposed methods to handle this problem.

Under-sampling, a cost-sensitive learner (weighted Random Forests) and RUSBoost are all implemented. Using more appropriate evaluation metric (AUC), we investigated their respective performance.

Results show that under-sampling is the most effective way to deal with class imbalance. RUSBoost, as a specific algorithm designed to deal with class imbalance problem, is also effective but not as good as under-sampling. Weighted Random Forests only improves the performance of appetency classification problem.

In some ways, these findings are surprising since we would expect more recent methods like RUSBoost and Weighted Random Forests to yield better classification performances. The reasons for this need more study but it is clear that researchers working with unbalanced datasets need to be careful in selecting their classification methods since this can make a real difference in performance, and it can be difficult to predict in advance which method will be most appropriate for a given dataset.

For future work, it would be useful to explore over-sampling technique as it is also reported to be effective in dealing with class imbalance problem [1]. Problems arising in the presence of class imbalance can be divided into several categories [2]. It is also of great interest to explore what exactly the arising problems associated with this dataset are and try to deal with each accordingly. In addition, it would be interesting to compare methods used in this study to some state of art methods, such as support vector machines, Bayesian methods, etc.

References

[1] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI*, 2000, pp. 111–117.

[2] G. M. Weiss, "Mining with rarity: a unifying framework," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.

[3] G. M. Weiss and F. J. Provost, "Learning when training data are costly: the effect of class distribution on tree induction," *J. Artif. Intell. Res.(JAIR)*, vol. 19, pp. 315–354, 2003.

[4] J. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4626–4636, 2009.

[5] Kdd cup 2009: Customer relationship prediction. [Online]. Available: http://www.sigkdd.org/kdd-cup-2009-customer-relationship-prediction

[6] I. Guyon, V. Lemaire, M. Boullé, G. Dror, and D. Vogel, "Analysis of the kdd cup 2009: Fast scoring on a large orange customer database," *The 2009 Knowledge Discovery in Data Competition (KDD Cup 2009) Challenges in Machine Learning, Volume 3*, 2009.

[7] H.-Y. Lo, K.-W. Chang, S.-T. Chen, T.-H. Chiang, and C.-S. Ferng, "An ensemble of three classifiers for kdd cup 2009: Expanded linear model, heterogeneous boosting, and selective naive bayes," *The 2009 Knowledge Discovery in Data Competition (KDD Cup 2009) Challenges in Machine Learning, Volume 3*, p. 53, 2009.

[8] P. Doetsch, C. Buck, P. Golik, N. Hoppe, M. Kramp, J. Laudenberg, C. Oberdörfer, P. Steingrube, J. Forster, and A. Mauser, "Logistic model trees with auc split criterion for the kdd cup 2009 small challenge." *Journal of Machine Learning Research-Proceedings Track*, vol. 7, pp. 77–88, 2009.

[9] A. Niculescu-Mizil, C. Perlich, G. Swirszcz, V. Sindhwani, Y. Liu, P. Melville, D. Wang, J. Xiao, J. Hu, M. Singh *et al.*, "Winning the kdd cup orange challenge with ensemble selection," *The 2009 Knowledge Discovery in Data Competition (KDD Cup 2009) Challenges in Machine Learning, Volume 3*, p. 21, 2009.

[10] J. Xie, V. Rojkova, S. Pal, and S. Coggeshall, "A combination of boosting and bagging for kdd cup 2009-fast scoring on a large database." *Journal of Machine Learning Research-Proceedings Track*, vol. 7, pp. 35–43, 2009.

[11] D. Sorokina and C. EDU, "Application of additive groves ensemble with multiple counts feature evaluation to kdd cup?09 small data set," *The 2009 Knowledge Discovery in Data Competition (KDD Cup 2009) Challenges in Machine Learning, Volume 3*, p. 95, 2009.

[12] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: Improving classification performance when training data is skewed," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.