An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets

Bee Wah Yap[1], Khatijahhusna Abd Rani[2], Hezlin Aryani Abd Rahman[1], Simon Fong[3], Zuraida Khairudin[1], Nik Nairan Abdullah[4]

[1,2] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Selangor,Malaysia
[3]Faculty of Science and Technology, University of Macau, China
[4] Faculty of Medicine, Universiti Teknologi MARA, Selangor, Malaysia
[1] {beewah, hezlin, zuraida_k}@tmsk,.uitm.edu.my, [2] ejahhusna@gmail.com, [3]ccfong@umac.mo, [4]niknairan@yahoo.com

Abstract. Most classifiers work well when the class distribution in the response variable of the dataset is well balanced. Problems arise when the dataset is imbalanced. This paper applied four methods: Oversampling, Undersampling, Bagging and Boosting in handling imbalanced datasets. The cardiac surgery dataset has a binary response variable (1=Died, 0=Alive). The sample size is 4976 cases with 4.2% (Died) and 95.8% (Alive) cases. CART, C5 and CHAID were chosen as the classifiers. In classification problems, the accuracy rate of the predictive model is not an appropriate measure when there is imbalanced problem due to the fact that it will be biased towards the majority class. Thus, the performance of the classifier is measured using sensitivity and precision Oversampling and undersampling are found to work well in improving the classification for the imbalanced dataset using decision tree. Meanwhile, boosting and bagging did not improve the Decision Tree performance.

Keywords- Bagging, Boosting, Oversampling, Undersampling, Imbalanced data

1 Introduction

In recent years, there have been great interests in mining imbalanced datasets. In data mining classification problems, most classifiers such as logistic regression, decision tree and neural network work well when the class distribution of the categorical target or response variable in the dataset is balanced. However, for real problems such as document classification [1], loan default prediction [2], fraud detection [3] or medical classification [4] which involve a binary response variable, the dataset are often highly imbalanced. For a binary response variable with two classes, when the event of interest (eg: 'Died' due to a certain illness) is underrepresented, it is referred to as the positive or minority class. Thus, the number of cases for the negative or majority class is very much higher than the minority cases. When the percentage of the minority class is less than 5%, it is known as a rare event [5]. When a dataset is

T. Herawan et al. (eds.), *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, Lecture Notes in Electrical Engineering 285, DOI: 10.1007/978-981-4585-18-7_2,

imbalanced or when a rare event occurs, it will be difficult to get a meaningful and good predictive model due to lack of information to learn about the rare event. There are three approaches to handling imbalanced datasets: data level, algorithmic level and combining or ensemble methods. The data level approach involves resampling to reduce class imbalance. The two basic sampling techniques include random oversampling (ROS) and random undersampling (RUS). Oversampling randomly duplicates the minority class samples, while undersampling randomly discards the majority class samples in order to modify the class distribution. It has been reported that oversampling may lead to overfitting as it makes exact copies of the minority samples while undersampling may discards potential useful majority samples [6-10]. The algorithmic level approach is when machine learning algorithms are modified to accommodate imbalanced data while combining methods involve mixture-of-experts approach [6]. Meanwhile, [11] categorized the approaches as algorithm level, data level, cost-sensitive approach [12] and ensemble methods. Cost-sensitive methods combine algorithm and data approaches to incorporate different misclassification costs for each class in the learning phase. The two most popular ensemble-learning algorithms are boosting and bagging. Bagging stands for "**B**ootstrap **Agg**rega**ting**" whereby bootstrap samples are drawn randomly with replacement. Meanwhile, Boosting algorithms tries to improve the accuracy of a classifier by a reweighting of misclassified samples ([5], [13-14]).

This study examined the predictive performance of three decision tree (DT) algorithms: CART (Classification and Regression Tree), C5 and CHAID (Chi-Square Automatic Interaction Detection) after using oversampling, and undersampling techniques for a cardiac surgery imbalanced dataset. The DT performances are also compared using the bagging and boosting technique.

The rest of this paper is structured as follows: Section 2 reviews some past studies on comparison and applications of methods in handling imbalanced datasets. The ROS, RUS, Bagging and Boosting methods are explained in Section 3. The results are presented in Section 4 and Section 5 concludes the paper.

2 Literature Reviews

The class imbalance problem has been reported as a major obstacle to the induction of a good classifier in Machine Learning algorithms [15]. Most studies on comparisons of methods for handling imbalanced datasets used several different data sets, several different approaches and several classifiers such as Logistic Regression, C4.5, neural network and SVM (Support Vector Machine). This section reviews some of these studies.

In a study by [16] on nosocomial infection risk, the dataset comprises of 683 patients, whereby only 75 (11%) were infected or positive and 89% were negative cases. The difficulty to recognize the minority class took them to propose resampling techniques. They used a new resampling approach in which both oversampling of rare positives and undersampling of the noninfected majority rely on synthetic cases (prototypes) generated via class-specific subclustering. They reported that their novel

resampling approach performs better than classical random resampling. The predictive performance of Support Vector Machine (SVM) Decision tree, Naïve Bayes, Adaptive Boosting (Adaboost) and Instance-Based Learner (IB1) improved with their new sampling approach. Their results also shows that support vector algorithm in which asymmetrical margins are tuned to improve recognition of rare positive cases are effective for nosocomial infection detection. [17] implemented three different algorithms, namely, Logistic Regression (LR), Neural Network (NN) and Chi-squared Automatic Interaction Detection (CHAID) to a marketing dataset which consist of 2826 (17%) who bought the product (positive examples) and 14130 (83%) who did not buy the product (negative examples). The three classifiers performance were based on accuracy, hit rate and AUC and were compared for various imbalance datasets generated from the original dataset. They reported that hit rate (precision) is a better measure of classifier performance for imbalanced dataset and CHAID can be used to develop marketing models. Meanwhile, [1] implemented undersampling and cost sensitive learning in handling imbalanced data in biomedical document classification. They concluded that both undersampling and cost sensitive learning can improve the performance of Bayesian Network classifier. The measures of performance used were sensitivity rate, precision rate, F-score and false positive rate (FPR). The Synthetic Minority Oversampling Technique (SMOTE) was proposed by [18] and involves generation of synthetic samples. Their experiment involves nine different imbalanced datasets and three classifiers, which are decision tree classifier, Ripper classifier and a Naïve Bayes Classifier. They found that combination of SMOTE and undersampling performs better than only undersampling the majority class. The methods were evaluated using area under Receiver Operating Characteristics Curve (AUC), accuracy of minority class and accuracy of majority class.

Several studies have compared the Bagging and Boosting methods. Boosting has been shown to be promising in handling imbalanced data. The case study by [5] on predicting customer attrition risk showed that combination of boosting and case sampling can improve logistic regression performance. A good explanation on bagging and boosting algorithm is given by [19]. They implemented these techniques on two datasets and showed the significant performance of boosting. Recently, the hybrids of bagging and boosting techniques such as RUSBoost and UnderBagging are reported to achieve higher performances than many other complex algorithms [11]. Meanwhile [14] also investigated four boosting and bagging techniques: SMOTEBoost, RUSBoost, EBBBag and RBBag. Their experiments showed that bagging generally outperforms boosting for noisy and imbalanced data. They recommended bagging without replacement techniques for handling imbalanced data. Recently, [20] reported that combining under-sampling, classification threshold selection, and using an ensemble of classifiers can improve the Naive Bayes classifier to overcome the imbalance problem

Although there have been various developments for handling imbalanced data especially in the ensemble methods, the new variants or hybrid approach are quite complex, not yet available in data mining software and may be difficult for practitioners. Besides, there is still no conclusive evidence as to which is the best approach although undersampling and oversampling remain popular as it is much easier to implement. The next section explains the application to a real dataset using the sampling, bagging and boosting techniques.

3 Methodology

3.1 Cardiac Surgery Data

In this study, we only focus on the binary (or two classes) classification problems. The positive instances belong to the minority class and the negative instances belong to the majority class. The Cardiac Surgery data were obtained from a local hospital. The data contain cases from a study on prediction of survival of cardiac surgery patients. The response variable has two classes: alive and died. The cardiac surgery dataset comprises of 4976 cases with 4.2% who had 'Died' after surgery and 95.6% 'Alive' cases. For this study, eight independent variables were selected: gender (f,m); Age Group (18-40, 40-60, above 60); Comorbidities (Hypertension, Diabetes, Both, None); Surgery Type (CABG only, CABG and Valve Surgery, Others); Chest Reopen (Yes, No); Atrial Fibrillation (Yes, No), Wound Infection (Yes, No); EUROScore. There were no problems of imbalanced data for the categorical predictors.

3.2 Undersampling and Oversampling

IBM SPSS Modeler 15.2 was used for random undersampling and oversampling of the imbalanced data. The supernode was used to perform these sampling techniques. First, we need to determine the distribution of two classes before we proceed to balance out the data. In undersampling, the majority classes are eliminated randomly to achieve equal distribution with the minority class. On the other hand, in oversampling the minority classes are replicated to achieve equal distribution with the majority class. Thus for undersampling the class distribution of minority to majority cases is 209:209 while for oversampling it is 4767:4767.

3.3 Bagging and Boosting

The bagging method proposed by [21] is a bootstrap ensemble method that can be applied to enhance model stability. In the Bagging approach, all instances in the training dataset have equal probability to be selected. All samples were replicates based on bootstrap approach. The replicates are samples drawn with replacement and with the same size as the training sample. For each bootstrap set, one model is fitted. The final predictions of the cases are produced using the voting approach.
Consider a training dataset with N samples belonging to two classes. The two classes are labeled as $y \in \{0,1\}$. The steps involved in the Bagging process ([13], [19-21]).
Are as follows:

1. For iterations $t=1, 2, \ldots, T$: # by using T=10

a) Randomly select a dataset with N samples from the original training with replacement.

b) Obtain a learner, *f(x) (predictive model or classifier)* from the resample dataset

c) By using the model, $f(x)$ predicts the cases.

2. Combine all predicted model $f^{t}(x)$ into an aggregated model $f^{A}(x)$

3. By using voting approach, return class that has been predicted most often.

The adaptive Boosting algorithm, named AdaBoost is available in IBM SPSS Modeler 15.0. Consider a training dataset with *N* samples belonging to two classes. The two classes are labeled as $y \in \{0,1\}$. The steps involved in the Boosting process are as follows by [19, 22-23]:

1. Assign initial equal weights to each samples in the original training set:

$$w_i^1 = 1/N, i = 1,2,\ldots,N$$

2. For iterations *t*=1, 2, . . . , *T:* # by using T=10

a) Randomly select a dataset with *N* samples from the original training set using weighted resampling. The chance for a sample to be selected is related to its weight. A sample with a higher weight has a higher probability to be selected.

b) Obtain a learner, *f(x) (predictive model or classifier)* from the resampled dataset.

c) Apply the learner *f(x)* to the original training dataset. If a sample is misclassified, its error=1, otherwise=0.

d) Compute the sum of the weighted errors of all training samples.

$$error^t = \sum_{i=1}^{N} (w_i^t \times error_i^t)$$

e) Calculate the confidence index of the learner *f(x)*:

$$\alpha^t = \frac{1}{2} \ln\left(\frac{1 - error^t}{error^t}\right)$$

The confidence index of the learner *f(x)* depends on the weighted error.

f) Update the weights of all original training samples:

$$w_i^{t+1} = w_i^t \exp(-\alpha^t * error_i^t)$$

If samples are correctly classified, the weights are unchanged, while the weights for misclassified samples are increased.

g) Then, renormalize weight, $w_i^t = \frac{w_i^t}{\sum_i^N w_i^t}$ so that, $\sum_i^N w_i^{t+1} = 1$

h) *T*=*t*+*1* , if error<0.5, and *t*<*T,* repeat steps (a)-(g); otherwise, stop and T=t-1.

i) After *T* iterations, *t=1,2…T,* there are T predicted model $f^t(x), t = 1,2,\ldots,T$. The final prediction for case j, is obtained by the combined prediction of the T models using voting approach:

$$y_j = sign\sum_{t=1}^{T}\alpha^t f^t(x)$$

Figure 1 displays the modelling flow using IBM SPSS Modeler 15.0. The original data set is connected to the TYPE node which is connected the PARTITION node for splitting the data into Training (70%) and Testing (30%) samples. The CART model nodes are then connected to the PARTITION node. The diamond shaped gold nuggets are the generated models. The performance measures are then obtained for the training and testing samples. The process is repeated for C5 and CHAID algorithms.

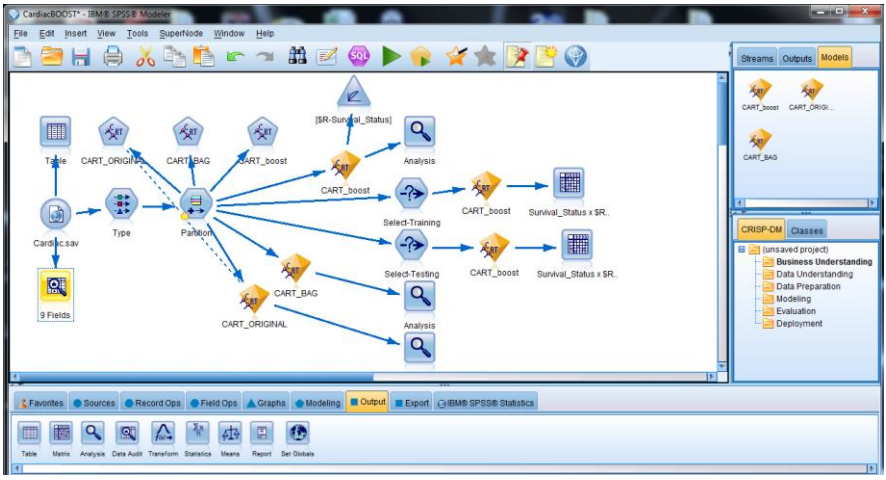

Fig 1. Bagging and Boosting using CART as classifier

3.4 Model Performance Measures

The classification accuracy rate (Acc), sensitivity (Sen), specificity (Spec) and precision rate (Pre) were chosen as the criteria in measuring the performance of the Decision Tree model.

Table 1. *Confusion Matrix*

Actual Class	Predicted Class	
	Positive ('Died')	Negative ('Alive')
Positive('Died')	True Positive (TP)	False Negative (FN)
Negative ('Alive')	False Positive (FP)	True Negative (TN)

Based on Table 1, the calculations are as follows:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad Sen = \frac{TP}{TP + FN},$$

$$Spec = \frac{TN}{TN + FP}, \quad \text{Pre} = \frac{TP}{TP + FP}$$

4 Results

The first column in Table 2 shows the performance measures for the original dataset. As expected the specificity is high (100%) and sensitivity is 0%. The results in the second and third column shows that with oversampling and undersampling, the sensitivity for the testing set has increased to 69.4% and 68.7% respectively. Oversampling has been reported to be prone to overfitting but in this study there was no problem of overfitting. The CART_Bagg results are similar to CART model for the original data set and CHAID. Meanwhile, CART_Boost improves with testing sensitivity (27.9%) and precision (42.2%). Taking into consideration that the small sample of minority class will result in much smaller number of minority cases in the training and testing samples, the CART, CHAID AND C5 algorithms were applied to the original data without any data partitioning. Both CART and CHAID classified all 209 minority cases into the majority group (sensitivity=0%) while C5 correctly classified 28 (13.4%) minority cases.

In Table 2, the results for CHAID are similar with CART for Original dataset with 0% sensitivity. Results for C5, C5_Boost and CHAID_Bagg are also similar with testing sensitivity 25% and precision 48.6%. Bagging is not available for C5 in IBM SPSS Modeler. The results in Table 2 and Table 3 show that sampling approach

performs better than bagging and boosting methods. Boosting and bagging did not improve the sensitivity of the decision tree classifiers.

Table 2. Results for CART as base classifier

		CART_Original	Os	Us	CART_Bagg	CART_Boost
Acc	Training	95.9	79.1	81.9	95.9	96.2
	Testing	95.5	76.7	71.1	95.5	95.1
Sen	**Training**	**0.0**	**71.4**	**76.7**	**0.0**	**34.7**
	Testing	**0.0**	**69.4**	**68.7**	**0.0**	**27.9**
Spe	Training	100.0	86.4	87.2	100.0	98.8
	Testing	100.0	84.5	73.5	100.0	98.2
Pre	**Training**	**0.0**	**83.6**	**85.8**	**0.0**	**55.7**
	Testing	**0.0**	**82.8**	**71.9**	**0.0**	**42.2**

Notes: **Acc**: Accuracy, **Sen**: Sensitivity, **Spe**: Specificity, **Pre** : Precision, **Os**: Oversampling, **Us**: Undersampling, **Bagg**: Bagging, **Boost**: Boosting

Table 3. Results for C5 and CHAID as base classifiers

		CART_ Original	C5	C5_Boost	CHAID	CHAID-Bagg	CHAID_ Boost
Acc	Training	95.9	96.2	96.2	95.9	96.2	96.5
	Testing	95.5	95.4	95.4	95.5	95.4	94.6
Sen	Training	**0.0**	**31.9**	**31.9**	**0.0**	**31.9**	**35.5**
	Testing	**0.0**	**25.0**	**25.0**	**0.0**	**25.0**	**23.5**
Spe	Training	100.0	98.9	98.9	100	98.9	99.1
	Testing	100.0	98.7	98.7	100	98.7	97.9
Pre	Training	**0.0**	**56.9**	**56.9**	**0.0**	**56.9**	**63.3**
	Testing	**0.0**	**48.6**	**48.6**	**0.0**	**48.6**	**34.8**

Notes: **Acc**: Accuracy, **Sen**: Sensitivity, **Spe**: Specificity, **Pre** : Precision, **Os**: Oversampling, **Us**: Undersampling, **Bagg**: Bagging, **Boost**: Boosting

5 Conclusion

Sampling approaches are much easier to implement for improving prediction of the minority case of a two-class classification problem. The random undersampling advantage is that all the minority cases are maintained as replication of minority case in oversampling will cause overfitting since it makes duplicates copy of the existing data. Besides, most classifiers assume that all cases are independent. The application of bagging and boosting in this study shows that they do not perform better than the random sampling strategies. For future research, a simulation study should be carried out whereby data are generated and then the different approaches are compared so as to obtain a conclusive decision on the best strategy to handle imbalanced data. The simulation study could investigate the effect of different methods of handling imbalanced data with different percentage of imbalance and for different classifiers. It is also important to note that the classifiers performance depend on data quality All datasets should be cleaned and imbalanced problems in categorical predictors (or features) should be determined so as to obtain a good predictive model with results that can be generalized.

Acknowledgments We thank the Research Management Institute (RMI) Universiti Teknologi MARA and the Ministry of Higher Education (MOHE) Malaysia for the funding of this research under the Malaysian Fundamental Research Grant, 600-RMI/FRGS 5/3 (16/2012).

References

1. Laza, R., Pavón, R., Reboiro-Jato, M., Fdez-Riverola, F.: Evaluating the effect of unbalanced data in biomedical document classification. Journal of integrative bioinformatics, 8(3):177, (2011). Doi:10,2390/biecoll-jib-2011-177
2. Brown, I., & Mues, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications, 39*(3), 3446-3453, (2012). doi: 10.1016/j.eswa.2011.09.033
3. Wei, W., Li, J., Cao, L., Ou, Y., Chen, J.: Effective detection of sophisticated online banking fraid on ectremely imbalanced data. World Wide Web (2013) 16:449–475. doi: 10.1007/s11280-012-0178-0
4. Rahman, N.N., Davis, D.N.: Addressing the Class Imbalance Problems in Medical Datasets. International Journal of Machine Learning and Computing, 3(2), 224-228, (2013).
5. Au, T., Chin, M.-L., & Ma, G.: Mining Rare Events Data by Sampling and Boosting: A Case Study. In S. Prasad, H. Vin, S. Sahni, M. Jaiswal & B. Thipakorn (Eds.), Information Systems, Technology and Management (Vol. 54, pp. 373-379): Springer Berlin Heidelberg, (2010).
6. Kotsiantis, S. B., Pintelas, P. E., Kanellopoulus, D.: Handling imbalanced datasets: A review. GESTS International Transactions on Computer Science and Engineering, Vol.30, (2006).
7. Drummond C., Holte, R. C.: C4.5, Class Imbalance and Cost-Sensitivity: Why Undersampling beats Oversampling, Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, (2003).

8. Drummond C., Holte, R. C.: Severe Class Imbalance: Why Better Algorithms Aren't the Answer. Proceedings of 16th European Conference of Machine Learning, LNAI 3720, 539-546, (2005).
9. Weiss, G. M.:Mining with rarity: a unifying framework. Sigkdd Explorations, *6*(1), 7-19 (2004).
10. Chawla, N. V.: Data mining for imbalanced datasets: An overview Data mining and knowledge discovery handboo*k* (pp. 853-867): Springer, (2005).
11. Galar. M., Fern´andez, A., Barrenechea, E., Bustinc, H., Herrera, F.: A review on Ensembles for Class Imbalanced Problems: Bagging-, Boosting- and Hybrid Based Approaches. IEEE Transactions on Systems. Man, .and Cybernetics-Part C. Applications and Reviews. Vol.42, No.4, 463-484 (2012).
12. Chawla, N. V., Cieslak, D. A., Hall, L. O., Joshi, A.: Automatically countering imbalance and its empirical relationship to cost. Data Mining and Knowledge Discovery 17, 2, 225-252 (2008).
13. Kotsiantis, S., Pintelas, P.: Combining bagging and boosting. International Journal of Computational Intelligence, 1(4), 324-333 (2004).
14. Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: Comparing Boosting and Bagging techniques with Noisy and Imbalanced Data, IEEE Transactions on Systems. Man, .and Cybernetics-Part A. Systems and Humans. Vol.41,No.3, 552-568 (2011).
15. Batista, G. E., Prati, R. C., Monard, M. C.: A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter, 6*(1), 20-29,(2004).
16. Cohen, G., Hilario, M., Sax, H., Hugonnet, S., Geissbuhler, A.: Learning from imbalanced data in surveillance of nosocomial infection. Artificial Intelligence in Medicine, 37(1), 7-18 (2006).
17. Duman, E., Ekinci, Y., Tanriverdi, A.: Comparing alternative classifiers for database marketing: The case of imbalanced datasets. Expert Systems with Applications, *39*(1), 48-53 (2012).
18. Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P.: SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, 16, 321-357 (2002).
19. Cao, D.-S., Xu, Q.-S., Liang, Y.-Z., Zhang, L.-X., Li, H.-D.: The boosting: A new idea of building models. Chemometrics and Intelligent Laboratory Systems, *100*, 1-11(2010). doi: http://dx.doi.org/10.1016/j.chemolab.2009.09.002
20. Klement, W., Wilk, S., Michaowski, W., Matwin, S.: Classifying severely imbalanced data. C. Butz and P. Lingras (Eds.): Canadian AI 2011, LNAI 6657, pp. 258–264 (2011).
21. Breiman, L.: Bagging predictors. Machine learning, 24(2), 123-140 (1996).
22. Freund, Y., Schapire, R. E.: A desicion-theoretic generalization of on-line learning and an application to boosting Computational learning theory (pp. 23-37): Springer,(1995).
23. IBM SPSS Modeler 15 Algorithms Guide. IBM Corporation (2012).