A Survey on Methods for Solving Data Imbalance Problem for Classification

Arpit Singh
M.E. Computer Engineering
SGSITS Indore
India

Anuradha Purohit
Assistant Professor Comp. Tech. & Appl. Dept.
SGSITS Indore
India

ABSTRACT

The term "data imbalance" in classification is a well established phenomenon in which data set contains unbalanced class distributions. Dataset is called unbalanced if it contains at least one class which is presented by very few examples. A range of solutions have been proposed for the problem of data imbalance including data sampling, cost evaluation of model, bagging, boosting, Genetic Programming (GP) based methods etc. This paper presents a survey of various methods introduced by researchers to handle data imbalance problem in order to improve classification performance and further the comparison between the methods on the basis of their advantages and disadvantages is done.

General Terms

Survey on methods for data imbalance.

Keywords

Classification, data imbalance, genetic programming, boosting, bagging, sampling.

1. INTRODUCTION

Classification has numerous applications in a wide variety of mining and other applications, such as detecting faces from images dataset, recognising voice in data of speech [1] etc. Given the amount of data that needs to be classified, automated classification systems are highly desirable.

Classifiers classify datasets according to class labels. Classifiers perform well if dataset is balanced. Dataset is called imbalance if one or more classes are presented by only a few number of examples. These under represented classes are called minority classes [2]. In unbalanced datasets class ratio is significant enough that classifier became biased with some classes (majority classes). Performance bias means solutions give high accuracy on the majority classes and poor accuracy on the minority classes. Recent studies show that uneven distribution of class examples can reduce the performance of learning algorithms [3]. Important training criteria like overall success or error rate can be influenced by the larger number of examples from the majority class [4]. Accurate classification of examples from minority class can be as important as, and in some cases more important than, accurate classification of examples from the majority class [5].

In this paper, the problem of data imbalance in classification is discussed, the effective measures are taken by various authors to prevent and control data imbalance is presented and two main approaches (internal and external) to handle data imbalance problem is compared with each other.

2. DATA IMBALANCE PROBLEM

A data set is called imbalance [2] if class distribution among classes in dataset is not uniform. In this condition there is at least one class which is represented by only a small number of examples (minority class), other classes make up the rest of data set (majority class). Recent research in the machine learning showed that using an uneven distribution of class examples in the learning process can leave learning algorithms with performance bias. It means that classifier gives high accuracy on the majority class but it gives poor accuracy on the minority class. This is because traditional training criteria such as the overall success can be greatly influenced by the larger number of examples from the majority class. As the minority classes play an important role in many real world problems, as the accurately classifying examples from this class is also very important. Researchers have distinguished data imbalance problem into two main types: Binary class data imbalance and multi class data imbalance [6].

2.1 Binary Class Data Imbalance

Dataset which contains only two classes is called binary dataset. If in the binary dataset there exists a class which is represented by only a few numbers of examples, then it is called binary class data imbalance problem. In binary class dataset zero class thresholds is generally used to separate two classes, so there is no need to identify the boundaries of classes in dataset.

2.2 Multi Class Data Imbalance

Dataset which contains more than two classes is called multiclass dataset. Data imbalance problem create additional overheads in multiclass dataset. Simple and efficient zero class thresholds cannot be used in multiclass dataset. Complex methods like Static Search Selection or Dynamic Search Selection need to be used. Some times to classify dataset, multiclass problem is needed to be divided into many binary class problems.

3. WORK DONE IN DATA IMBALANCE

There are two main approaches which are used to develop methods to solve the data imbalance problem [3]. In first approach transformation or sampling from the original unbalanced data set to create a balanced class distribution is used. These are called "external" approaches because the external training data are rebalanced while the learning algorithm remains unchanged. The second approach uses many cost adjustment techniques within the learning algorithm to fully use the original imbalance data in the training process. These method are called as "internal" approaches. The work done using external and internal approaches are as discussed.

3.1 Work Done using External Approach

Sampling, bagging and boosting are popular external approaches which are used by the researchers to handle data imbalance problem. The details of these methods are as described.

3.1.1 Sampling

Sampling is a set of methods that changes the size of training sets by adding or removing features from datasets. Under-sampling and over-sampling change the training sets by sampling a smaller majority training set and repeating instances in the minority training set. In both methods the level of imbalance is reduced to more balanced training set which can give better results. Both sampling methods have been shown to be helpful in imbalanced problems. Training time in under sampling is short, but can ignore useful data. Over sampling increases the training set size, and thus requires longer training time. Over sampling many times leads to over fitting because it repeats minority class examples.

Orriols et al. [6] discussed that sampling technique is an effective technique to handle data imbalance problem. In sampling supervised learning is used. The algorithm maintains a distribution D variable over the examples to determine which features should be more likely to be selected next. D variable is calculated on the basis of predefined values. Predefined valued are decided according to hit and miss manner. Different values of D variable tested on datasets. A set of values selected according to results achieved in problem. Weights are assigned to all examples in dataset. D is updated by reducing the weights of those features that have been used in classification. This gives other features a better chance to be selected into the next feature subset. Performance of classifiers will be evaluated on different sets of example and finally those subset of datasets will be selected which will give good performance of classifiers. This approach is tested on binary unbalanced benchmark datasets. This method does not include number of features of dataset. Performance of this method decreases as number of features in dataset increases.

Garcia et al. [7] presented a new sampling approach which performs under sampling on the majority class by selecting a representative subset of the negative examples. In this technique all positive examples must be kept in the data set, even knowing that some of them can be noisy. It uses a nearest neighbor (NN) classifier and the geometric mean as performance measure. Despite the successful results, a problem remained unsolved. The problem which is common to all these sampling techniques is that they do not permit control on the number of examples to be removed. Consequently, eliminated examples can be too many or too few to solve the imbalance problem.

It is identified that by applying sampling, useful sample can be removed or some unnecessary samples can be added into datasets. As a result, dataset can be modified in such a way that problem can be changed.

3.1.2 Bagging

In bagging the original training set is divided into N subsets of the same size. Each subset is used to create one classifier (classifier learned from those subsets). A compound classifier is created as the aggregation of particular classifiers.

Breiman et al. [12] presented bootstrap aggregation, or bagging. It is a technique that can be used with many classification methods and it applies regression methods to reduce the variance associated with prediction which improves improve the prediction process. Prediction method is applied to each bootstrap sample and then the results are combined by averaging for regression and simple voting for classification to obtain the overall prediction. Tests on real and simulated data sets using classification and regression trees and subset selection in linear regression show that bagging can give substantial gains in accuracy. The element is the instability of the prediction method. If altering the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy.

3.1.3 Boosting

Boosting is a machine learning technique based on the observation that finding many rough rules is easier than finding highly accurate prediction rule.

Kerns et al. [13] introduced boosting approach to convert weak learners into strong learners. Boosting algorithm calls base learning algorithm repeatedly, each time feeding it a different subset of training examples. Each time it is called, the base learning algorithm generates a new weak prediction rule, and after many rounds, the boosting algorithm must combine these weak rules into a single prediction rule that will be much more accurate than any one of the weak rules. A weak learner is defined to be a classifier which is only correlated with the true classification (it labels examples better than random guessing). In contrast, a strong learner is well correlated with true classification values. Variant of this algorithm is as described.

W. Lee [14] presented boosting algorithm to solve the problem of unbalanced data. This approach trains multiple classifiers using smaller and usually balanced subsets of the original data, which are combined in final classification step in ensemble process. These subsets usually contain all minority instances and the same number of randomly selected majority instances. It focuses on those instances which are not already accurately learned using weights to decide the values of probability of selection.

Yoav et al. [15] presented AdaBoost algorithm to solve the problem of data imbalance. It is also called adaptive boosting. It focuses on difficult data points. Difficult data points are data points that have been misclassified most by the previous weak classifier. AdaBoost combines these weak classifiers into a comprehensive prediction by an optimally weighted majority vote of weak classifier. AdaBoost is fast, simple and easy to program. In AdaBoost there is no need to tune parameters of boosting. AdaBoost does not need prior knowledge about weak learner. It’s an effective method but it is vulnerable to uniform noise. AdaBoost uses weak classifiers which lead to low margins and over fitting.

Chris et al. [16] presented a hybrid approach to solve data imbalance problem which is called RUSBoost. It combines sampling and boosting algorithms to alleviate data imbalance problem. Let x be a point in the feature space X and y be a class label in a set of class labels Y. Each of the m examples in the data set (S) can be represented by the tuple (x,y). Let t be an iteration between one and the maximum number of iterations T (number of classifiers in the ensemble), ht be the weak trainer trained on iteration t, and ht(x) be the output of hypothesis ht, for instance, x. Let Dti be the weight of the ‘i’ example on iteration. In step 1, the weights of each example are initialized to 1/m, where m is the number of examples in the training data set. In step 2, T weak hypotheses are iteratively trained. In this step, RUS is applied to remove the majority class examples until N% of the new training data set S t belongs to the minority class. As a result, S assigned a new weight distribution D.D is passed to the weak classifiers. The pseudo loss t based on the original training data set S and weight distribution D is calculated in this step, the weight update parameter α is calculated as t/(1 − t). Next, the weight distribution for the next iteration D+1 is updated and normalized. After T iterations of step 2, the final hypothesis is returned as a weighted vote of the T weak hypotheses. Limitation of RUSBoost is that it does not include other learners and it does not consider performance matric.

It is identified in boosting that, classifiers are trained on usually balanced datasets, so there is no guaranty that final classifier is good solution for unbalanced dataset.

3.2 Work Done using Internal Approach

Various techniques have been presented by researchers by using internal approach. Common internal approach includes assigning different misclassification costs to incorrect class predictions [21] or developing improved training criteria that are more sensitive to the unbalanced class distributions compared to the standard overall accuracy or overall error rate. Improved training criteria include the average classification accuracy of the minority and majority classes. In internal approach only few methods have been suggested by researchers. One of the popular techniques which solve the data imbalance problem by using an internal approach is Genetic Programming (GP) [2]. GP is an evolutionary algorithm technique inspired from biological evolution to find computer programs that perform a user-defined task. In GP, solution to a problem is represented as a computer program. Darwinian principal of natural selection is used to evolve a population of computer programs towards an effective solution of specific problem. In [3], M. Zhang presented three fitness functions in GP to solve the data imbalance problem. These fitness functions are as follows.

(i) M. Zhang et al. [3] has described a way to control data imbalance problem by using area under curve (AUC) training criteria in GP. AUC is a useful metric to measure classifier performance, generating the AUC requires multiple performance points (performance thresholds) which are computationally costly to produce. Formula to represent performance point is given in (1):

$$\frac{\sum_{i=0}^{N_{min}} \sum_{j=0}^{N_{maj}} I(xi, yj)}{N_{min} * N_{maj}} \quad \text{… (1)}$$

In (1) N_{min} is number of examples in minority class. N_{maj} is number of examples in majority class. AUC conducts a series of pair wise comparisons on an example-by-example basis between minority class x and majority class y examples collecting “rewards” (1 point) for those cases in which indicator function I(x, y) enforces two constraints. The first constraint, x > 0, requires that the minority class example x is classified correctly. A minority class example is correctly classified if the genetic program output is a positive number. The second constraint, x > y, requires that the genetic program output for minority class example x is larger than the genetic program output for majority class example y. This constraint ensures that while majority class example y may not be classified correctly.

(ii) Next fitness function presented a variation on recently successful technique to handle data imbalance problem in GP. It described new training criteria to calculate individual class performance with overall performance.

$$\frac{hits_{min}}{N_{min}} + \frac{hits_{maj}}{N_{maj}} + \frac{hits}{N} \quad \text{… (2)}$$

In (2) $hits_{min}$ and $hits_{maj}$ represent the number of correctly classify examples in minority and majority class respectively. The idea of training criteria is that overall performance should be considered alongside improving the accuracy of both classes and not compromised by an improvement in only one class. N, N_{min}, N_{maj} represents the number of training examples in dataset, minority class, majority classes respectively.

(iii) Fitness function described in (3) evaluates the performance of population of classifier.

$$\frac{\sum_{i \in Min} \sum_{j \in Maj} I(P_i, P_j)}{Min * Maj} \quad \text{… (3)}$$

In (3) a series of pair wise comparisons are done between the genetic program outputs when evaluated on examples from the minority and majority classes. It effectively measures the ordering of minority to majority class outputs. It calculates fitness where P i and P j represent the outputs of a genetic program when evaluated on an example from the minority and majority classes, respectively. The indicator function returns 1 if P i > P j and P i ≥ 0. This enforces both the zero class thresholds and the required ordering of minority and majority class outputs in evolved solutions. The denominator ensures that function returns values between 0 and 1, where 1 indicates optimal AUC and 0 indicates poor AUC.

M. Zhang et al. [4] evolve diverse ensembles using GP for classification with unbalanced data. The evolved ensembles comprise of nondominated solutions in the population where individual members vote on class membership. A solution will dominate another solution if it is at least as good as the other solution on all the objectives and better on at least one. Solutions are nondominated if they are not dominated by any solution in the population. This function determines the importance of particular class in classification results. One of the key advantages of this approach is that the evolved Pareto front represents highly accurate classifiers, each with a different performance bias toward either class. However, as the front of nondominated solutions has as much information as any single individual, utilizing the combined classification ability of these solutions in a competitive voting or ensemble-based scenario proved beneficial. This strategy has proved successful in previous ensemble learning approaches.

Heywood et al. [9] developed fitness functions for a multiple minority class classification problem. This fitness functions focused on the number of correct classifications for each minority class. Fitness function assigned adjusting weights dynamically to certain examples as a reward for correctly classifying examples. A hierarchical two-tier evaluation process called “tie breaker” fitness and it resolves classifier performance when the two classifier have equal performance.

It is identified in work done in internal approach is that by applying no modification in datasets problem remains unchanged and main focus on learning from datasets.

4. COMPARISION OF METHODS TO HANDLE DATA IMBALANCE

External techniques described in Section III have been used successfully in data imbalance problems. Some of the advantages external techniques identified are as given.

(i) If process of balancing datasets is done with precaution, learning algorithms gives better results.

(ii) External techniques are free from the run and trial overheads of cost evaluation techniques because in some problems the cost is not monetary value.

(iii) External techniques provide goal oriented results since these are specific to particular problems.

External techniques are effective in solving data imbalance problem but there are two main limitations which are identified. These limitations are as follows.

(i) These techniques can add a computationally expensive overhead to the learning process. External techniques must be applied repeatedly. It can lead to over-fitting and poor classification by a classifier as potentially.

(ii) It is the possibility that useful training samples can be excluded from dataset so learning process will get effect.These approaches require a priori task-specific knowledge about the data.

Due to these limitations, researchers have focused on “internal” or algorithm-level approaches. In this technique learning algorithm is adjusted carefully according to the uneven distribution of class examples in the original data set. Some of the advantages of internal approach identified are as given.

(i) In various data imbalance contexts like medical diagnosis, intrusion detection, fraud detection and external approaches don’t perform well because in these types of datasets not only class distribution is imbalanced but cost of misclassification is also asymmetric.

(ii) Internal techniques provide general solutions for data imbalance problem because these are not specific to particular problems.

These approaches for cost adjustment are effective but there are two main limitations which are identified. These limitations are as follows.

(i) Misclassification costs must calculate priori. Misclassification costs have to calculate for particular problem which will require many trial process to decide.

(ii) Many new fitness functions are handcrafted to suit a particular classification problem which will require certain expertise about the dataset and which will affect the generalization of classifiers.

Conclusion of this survey paper is described in next section.

5. CONCLUSION

In this paper comparative study of different approaches described by various researchers to handle data imbalance problem has been done. Data imbalance problem hampers the performance of classifier. A lot of research has been done to find solutions to avoid and reduce the problem of data imbalance. For this various strategies have been presented in this paper.

Our contribution in the paper can be summarized as follows:

(i) This paper presents the theoretical concept of Data Imbalance problem.

(ii) This paper identifies different types of data imbalance problem.

(iii) This paper discusses about different methods to avoid and handle data imbalance in classification.

(iv) This paper presents comparison among various methods to handle data imbalance problem.

6. ACKNOWLEDGMENTS

My thanks to the Anuradha Purohit mam who have contributed towards development of the paper.

7. REFERENCES

[1] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press, 1992.

[2] Urvesh Bhowan, Mark Johnston and Mengije Zhang “Developing New Fitness Functions in Genetic Programming for Classification With Unbalanced Data” IEEE Transaction on system, man and cybernetics—part b, volume 42, pp 406-421 (2012).

[3] J. Eggermont, J. N. Kok, and W. A. Kosters, “Genetic programming for data classification: Partitioning the search space,” in Proc. ACM SAC, pp. 1001–1005, 2004.

[4] M. Zhang and W. Smart, “Multiclass object classification using genetic programming,” in Proc. Appl. Evol. Comput. vol. 3005, LNCS, 2004, pp. 369–378.

[5] U. Bhowan, M. Johnston, and M. Zhang, “A comparison of classification strategies in genetic programming with unbalanced data,” in Proc. 23rd Australasian Joint Conf. Artif. Intell. vol. 6464, LNCS, J. Li, Ed., 2010, pp. 243–252.

[6] A. Orriols, “evolutionary rule based system for dataset,” in springer verlag soft comput., 2008, pp. 213-225.

[7] A. G. García and M. J. Muñoz-Bouzo. Sampling-related frames in finite U-invariant subspaces. Appl. Comput. Harmon. Anal., 39:173-184, 2015

[8] G. Patterson and M. Zhang, “Fitness functions in genetic programming for classification with unbalanced data,” Proceedings of the 20th Australian Joint Conference on Artificial Intelligence, vol. 4830, pp. 769–775, December 2007.

[9] Doucette and M. I. Heywood, “GP classification under imbalanced data sets: Active sampling and AUC approximation,” in Proceedings of EuroGP 08, pp. 266–277, 2008.

[10] Song, M. Heywood, and A. Zincir-Heywood, “Training genetic programming on half a million patterns: an example from anomaly detection,” IEEE Transactions on Evolutionary Computation, vol. 9, pp. 225–239, June 2005.

[11] J. Eggermont, A. Eiben, and J. van Hemert, “Adapting the fitness function in GP for data mining,” EuroGP’99, LNCS, vol. 1598, pp. 193–202, 1999.

[12] L breiman “Bagging Predictores”, in Machine Learnin Springer, Vol. 24, pp. 123–140, 1996.

[13] Kerns. “Thoughts on hypothesis boosting” in Machine learning Project, Vol 12, pp. 1-9, 1988.

[14] W. Lee “Margin and Boosting”, Machine Learning proceeding of 14th international conference, pp. 1-9, 1997.

[15] Yoav., Robert E. “A short introduction to boosting”, A journal of Japanese Society for Artificial Intelligence, vol. 14, pp. 771-780, September, 1999.

[16] Chris, “RUSBoost: A Hybrid Approach to Alleviating Class Imbalance Problem”, IEEE transactions on systems, man, and cybernetics part a: systems and humans, vol. 40, pp.185,197, 2010