Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5 – No.4, June 2016 – www.caeaccess.org

Various Approaches for Multiclass Imbalance Learning Issues with MLP

Ranjana Singh
ME, Student
Dr. D Y Patil School of Engineering and Technology
Savitribai Phule Pune University, Pune

Roshani Raut (Ade)
Associate Professor
Dr. D Y Patil School of Engineering and Technology
Savitribai Phule Pune University, Pune

ABSTRACT
Imbalance data is a major issue, which can be either binary or multiclass. Oversampling, Undersampling, SMOTE, SMOTEboost, Adaboost, OSS (One Sided Selection) and many other algorithms are there to deal with binary or multiclass imbalance issues. Software Defect Predictors (SDPs) and Software Cost Estimations (SCEs) are tools that used to classify the software elements into certain factors which helps in studying the imbalance problem. First take into consideration the SDPs to predict the defect prone part of software so that project can be completed with expected quality. In the same way for SCEs, certain factors will need to be considered for overall cost estimation in a way that financials can be managed and software elements can be done neatly. Class imbalance learning challenges, supervised learning difficulties where some classes have significantly more samples than others, i.e. dataset having a set of majority and minority samples. To make imbalance data balanced, most of the present study focused only on binary-class cases. In this paper, Adaboost.NC method is introduced and its result will be analyzed with proposed Dynamic Sampling method-Multilayer Perceptrons ((DyS)-MLP) for multiclass imbalance problem.

General Terms
Information retrieval through Data Mining.

Keywords
Multiclass Imbalance Learning; Multilayer Perceptrons (MLP); Dynamic Sampling (DyS); Software Defect Prediction.

1. INTRODUCTION

1.1 Estimation and prediction
To improve the version of software quality and to improvise the system in other word to go in detail just take an example of software cost estimation here cost is estimated before to manage the finance status and let the schedule be manage as almost all related terms are relate with finance either you are talking about resources availability, computer ,human resource first thing is budget accordingly can move further this the way finance status is managed and in the same way prediction of certain fault in software before it can reached to some major loss mode that's why many algorithm been mention to handle this issues but the major issues is imbalance nature of data because so many factors are there to consider. Some can have inor data and some can have major data so while classification might be minor will get neglected and only major will come under consideration so here again classification cannot be done with accuracy. So to maintain the quality of software and to manage the budget with proper cost estimation.to manage this models properly some algorithm have been tried and found that was mainly working for binary so moving for multiclass with various algorithm. To get the accurate classification. Some of them are random forest, some are naïve bayes, some supervised and unsupervised form.

1.2 Class Imbalance Learning
Here in this it can be explain in way that suppose you are considering the dataset car and as per the facilities have been provided by the provider of the car cost will be estimated.

So again no of factors will be there for cost consideration in car like comfort, look, color, brand in this cases there can have many minor and major factors will have to be considered for overall cost estimation of car by various brands.so it's a big issue to manage with multiclass imbalance data as here number of factors are there to consider for cost finalizing for car .so various algorithm are being proposed by various author to deal with binary and multiclass imbalance issues some of them are random forest, naïve bayes, bayes theorem, smote boost ,random oversampling and under sampling and many more but maximum of them are limited to binary class imbalance issues . so here in this paper proposed is to deal with multiclass imbalance issues with dynamic sampling algorithm with multilayer perceptron .back propagation and front propagation are the error correcting nodes and every time error is corrected by back propagation likewise error will be corrected and by adaboost.nc algorithm 1 also same dataset will be classified and comparison and analysis will be done for accuracy of each classifier. As per algorithm 2 every time correctly classified instances value is calculated and as per condition if satisfied will go under iteration and rest will be considered as correctly classified. In the same way for each sample correctly classified value will be calculated that delta value as per algorithm and result will calculated and at the end cross validation will be used for result evaluation will be used.

The rest of paper is organized as follows: Section II discussed in short about literature survey. Section III addresses Input. Section IV introduced input architecture. Section V had algorithms and practically exercised results are described in section VI. Section VII wrap up the paper.

2. LITERATURE REVIEW

2.1 Estimation with prediction.
As per author [2], concentration on prediction of student career with some psychometric test to some sort of training to judge the student's caliber in few specific areas, as talking

about student database so incremental learning approach is must and again for classification some algorithm been referred mainly used by author naïve bayes, Kstar and SVM with Voting's.

In [3], the author concentrated on number of fault in systematic way by generating algorithm. As at present fault in software can have disaster impact not just in quality of software but can also adversely impact the overall budget and financials. Some times because of inadequate information to mask the fault is not easy in software development perspective. To deal with that here genetic programming approach is used. The author has taken 10 different fault dataset from promise repository. The Error rate, Recall and Completeness of the fault prediction model are used to evaluate the performance of the proposed approach.

2.2 Pre-Sampling Methods

In [8], author concentrated on cost sensitive software defect prediction model by considering binary classes defected and not defected as if effected class is misclassified it can have disaster effect on cost of specific software so here main algorithm has been referred are threshold-moving i.e. moving to non-fault prone modules and boosting as it is known for weighting rule, boosting is done for minority set. The performances of the algorithms grading are evaluated by using some specific datasets from NASA projects in terms of a singular measure, the Normalized Expected Cost of Misclassification (NECM).and has been evaluated that threshold moving is working best compare to other algorithm being mentioned on cost sensitive software defect prediction system.

2.3 Class Imbalance Learning

As per [9], the author approached for student's classification in batch wise with some specific areas like area of interest, enrollment number, performance etc to guide them for carrier choice or for further move in some specific field. so here again multiclass data sets are there to consider while classification so mainly algorithm has been referred .are 1B1/1BK, NNge, Kstar, naïve bayes etc been exercised.

As per [10] author highlighted the issue of extracting decision maker data from (raw data) and massive data, again here the major issues been discovered are imbalance nature of data mainly in massive data it is huge problem. Author considered field in this are surveillance, security, finance which has huge impact in present world .so here imbalance problem and remedy related to that particular issues(algorithm) been reviewed.

As per author [13], class imbalance is quite complex problem to accumulate the accuracy of various dataset and in solution of that various algorithm has been referred to overcome the imbalance issues. As per author specific system "Credit card fraud with stationary and non-stationary environment" can have major imbalance issues , few may have supervised where label is known with accurate class label and some will be incorrectly classified so here again many binary and multiclass algorithm is exercised in incremental learning approach.

3. PROPOSED SYSTEM

As most of existing system is dealing binary issues and various algorithms are there which are limited to deal with binary issues so here it has been a trying to move for multiclass to improve the system with no of factors to be considered for various field like software cost estimation, software fault alarm. some algorithms are there to deal with multiclass imbalance data that is by merging all minor in one class and another class will be as major and again evaluation and analysis will be done but again there will be loss of some facts that can be important for some analysis so to avoid that shorts of constraints here planning to work with multiclass imbalance data with multilayer perceptron with three hidden layers and by back propagation and front propagation error will be detected and corrected for each epoch .As existing system are Adaboost.NC so result will be evaluated and again compared with output of dynamic sampling with MLP.

4. PROPOSED ARCHITECTURE

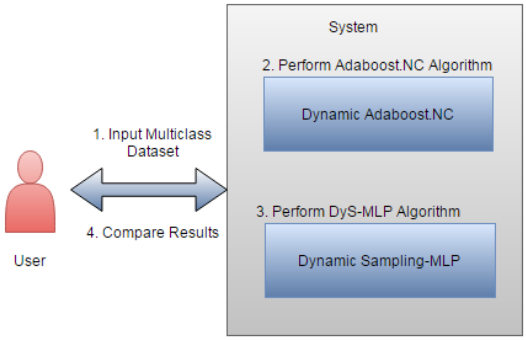

Fig 1: Architecture for Multiclass Imbalance

As in figure 1, Two Multi-class imbalance learning algorithms for software defects prediction and software cost estimation are evaluated and compare for analysis. One most important thing to consider by means of class imbalance learning method is to finalize the parameters which should be considered for evaluation as data is imbalance(as minor and major sets), and misclassification cost of classes, can be time-consuming and problem-dependent to tune.

4.1 Dynamic Adaboost.NC

Boosting is way to get accuracy while classification. Meant it boost the weaken part of data that in the sense provides strength to the part which can affect the overall accuracy and classification of data with evaluation of result .This dynamic adaboost algorithm is a way to solve the issues of binary class imbalance. In detail just take some examples of Software defect prediction, software cost estimation where to get prediction of defects or to estimate the cost, Adaboost adaptively adjust the parameters dynamically and negative correlation works in vice versa and simplify the training procedure, objective is to develop a better solution that combines the strength of AdaBoost.NC without the parameter setting issue. Dynamic version of AdaBoost.NC adjusts its parameter automatically during training based on a performance criterion. But AdaBoost.NC is less aggressive in finding defects, as it tries to maintain the performance balance between classes.

4.2 Dynamic Sampling MLP

Here multilayer perceptron performed a best and accumulated accuracy with dynamic sampling algorithm and as it is well known that oversampling and under sampling and many more. way is there to handle the issues of binary and multiclass imbalance but in dynamic sampling in place of over sampling and under sampling and other tactics here duplication of

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5 – No.4, June 2016 – www.caeaccess.org

dynamic samples will have to be done so that imbalance data can get balance then multilayer perceptron will be applied.

5. SAMPLING ALGORITHAM

5.1 Dynamic Adaboost.NC Algorithm 1

Initialization

D: $\{(X_1,Y_1),(X_2,Y_2)\ldots\ldots,(X_m,Y_m)\}$

A chosen performance criterion = ACC

Data weights $D_1(X_i)=1/m$, where m is number of instances

Penalty term $P_1(X_i)=1$

Penalty strength $\lambda= 9$

For training epoch=$\{1,2,\ldots T\}$

Process:

1. Train Weak Classifier f_t using distribution D_t
2. Get weak classifier $f_t : X \rightarrow R$
3. Calculate the penalty value for every example : X_i

   $P_t(x_i)=1|amb_t(x_i)|$
4. Calculate f_t's weight by error and penalty for discrete label outcome

$$\alpha_{t=}1/2_{\log\sum_i y_i = h_{t(x_i)}D_{t(x_i)(p_{t(}xi))}{}^{\lambda}/\sum_i y_i = h_{t(x_i)}D_{t(x_i)(p_{t(x_{i))}}{}^{\lambda}}$$

Where, α_t =error

5. If $Acc(f_t)>=Acc(f_t\text{-}1)$, then $\lambda= \lambda+1$

   Else $\lambda= \lambda\text{-}1$.
6. Update data weights D_t , and obtain new weights D_t+1 by error and penalty:

   $D_{t+1}(x_i)=(P_t(X_i))^{\lambda}D_t(X_i)\exp(-\alpha_t f_t(x_i)y_i)/z_t$

   Where z_t, is a normalization factor

Output:

The final ensemble:

$H(x)=sign(\sum\alpha_t f_t(x))$

5.2 Dys-MLPAlgorithm 2

Initialization:

Training set = TS

$epoch$ Denotes the number of epochs

m Denotes the number of classes in TS

n Denotes the number of examples in TS

Ratio of class $r_c = n_c / n$, where n_c is the number of examples belonging to class c

Normalizing factor $Z = \min_i \{r_i\}$

Majority class ratio $r_{\max} = \max_i \{r_i\}$

Minority class ratio $r_{\min} = \min_i \{r_i\}$

Process:

1. For every epoch randomly duplicate the examples to make balance training data.
2. Predict label using MLP to every example.
3. The probability that an example belonging to class c will be used to update the MLP is estimated as

$$p = \begin{cases} 1, & if \delta \le 0 \\ \exp\left(-\delta.\frac{r_c}{r_{\min}}\right) & otherwise \end{cases}$$

4. When ep^{th} epoch $ep > 2$ then duplicate ratio for class is heuristically attenuated to $a_c / Inep$, where a_c denotes duplicate ratio for class c
5. Sampled data $y_i = f(x_i)$

6. EXPERIMENTAL EVALUATIONS

6.1 Dataset

Proposed Dys-MLP algorithm and existed Adaboost.NC for comparison and have been evaluated on SDP software cost estimation binary or multiclass imbalanced data sets.

6.2 Experimental Results

Cocomo and Desharnais are two multiclass datasets for software cost estimation that are used for this evaluation. For SDP there are kc1, kc2, mc2 dataset available on PROMISE repository. To make them multiclass the defected class is divided into two classes for experiment. The following table shows the experimental result on these dataset in case of accuracy difference between DyS-MLP and Adaboost.NC.

Table 1. Accuracy difference between DyS-MLP and Adaboost.NC

Dataset	DyS-MLP (Accuracy in %)	Adaboost.NC (Accuracy in %)
Cocomo	78.89	82.15
Desharnais	84.55	80
Kc1	75.26	75.33
Kc2	86.25	84.55
Mc2	89.22	88.25

7. CONCLUSION

A simple dynamic sampling method, DyS, has been proposed which is effective for sampling imbalance data. DyS manages to dynamically select the training data to be used in each epoch of MLP. Further this proposed algorithm is compare with Adaboost.NC algorithm that overcomes the parameter setting issue. From comparative analysis it is found that proposed algorithm gives good results as compare to Adabbost.NC. Future work of this paper would be to study other classifiers (like MLP) to improve results of multiclass

imbalance problem resolution from software cost estimation or software defect prediction.

8. REFERENCES

[1] Shuo Wang, and Xin Yao, "Using Class Imbalance Learning for Software Defect Prediction", member IEEE, IEEE Transaction on Reliability, vol. 62, no. 2, June 2013.

[2] Ade, Roshani, and P. R. Deshmukh. "An incremental ensemble of classifiers as a technique for prediction of student's career choice." Networks & Soft Computing (ICNSC), 2014 First International Conference on. IEEE, 2014.

[3] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A systematic review of fault prediction performance in software engineering," IEEE Trans. Software Eng., vol. 38, no. 6, pp. 1276–1304, Nov.-Dec.2012.

[4] S. Wang and X. Yao, Negative Correlation Learning for Class Imbalance Problems School of Computer Science, University of Birmingham,2012.

[5] Ade, Roshani, and P. R. Deshmukh. "Classification of students by using an incremental ensemble of classifiers." Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), 2014 3rd International Conference on. IEEE, 2014.

[6] C. Catal, "Software fault prediction: A literature review and current trends," Expert Syst. Appl., vol. 38, no. 4, pp. 4626–4636, 2010.

[7] Ade, Roshani, and P. R. Deshmukh. "Efficient Knowledge Transformation System Using Pair of Classifiers for Prediction of Students Career Choice."Procedia Computer Science 46 (2015): 176-183.

[8] J. Zheng, "Cost-sensitive boosting neural networks for software defect prediction," Expert Syst. Appl., vol. 37, no. 6, pp. 4537–4543, 2010.

[9] Ade, Roshani, and P. R. Deshmukh. "Instance-based vs Batch-based Incremental Learning Approach for Students Classification." International Journal of Computer Applications 106.3 (2014).

[10] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowledge Data Eng., vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[11] Ade, Roshani, and Prashant Deshmukh. "Efficient knowledge transformation for incremental learning and detection of new concept class in student's classification system." Information Systems Design and Intelligent Applications. Springer India, 2015. 757-766.

[12] T. Mu, J. Jiang, Y. Wang, and J. Y. Goulermas, "Adaptive data embedding framework for multiclass classification," IEEE Trans. Neural Netw. Learn. Syst., vol. 23, no. 8, pp. 1291–1303, Aug. 2012.

[13] Kulkarni, Pallavi Digambarrao and Roshani Ade. "Learning from Unbalanced Stream Data in Non-Stationary Environments Using Logistic Regression Model: A Novel Approach Using Machine Learning for Assessment of Credit Card Frauds." Handbook of Research on Natural Computing for Optimization Problems. IGI Global, 2016. 561-582. Web. 9 Jun. 2016. doi:10.4018/978-1-5225-0058-2.ch023