EFFICIENT METHODS TO SOLVE CLASS IMBALANCE AND CLASS OVERLAP

[1#]S. Lavanya, [2*]Dr. S. Palaniswami, 3**S.Sudha

1. Department of CSE, Anna University Regional Centre, Coimbatore, India

2. Government college of Engineering, Bodinayakanur, India

3. PG scholar , Department of CSE, Anna University Regional Centre, Coimbatore, India

***Abstract*-In machine learning and data mining class imbalance and class overlap are the two main problems; it is not only done on two dataset it is handled on multi class scenario. The imbalance problem can be treated as small disjunct problem which can be solved by using larger training dataset, but the overlap problem is different it affect the performance of classifier while using training data when overlap is present. Neural network is used to train the dataset, due to overlap and imbalance problem the performance of neural network is affected.in this paper various methods to overcome the class imbalance and class overlap problem is analyzed.**

Keyword: **Multi-class imbalance, Overlapping, Back-propagation, editing techniques, smote technique**

1. INTRODUCTION

The class imbalance problem means when the training data is more from one class compared to other class. Many classifier shown to give poor performance in identifying minority class in case where large imbalance present.[1]the imbalance problem is not a problem when the training set is sufficiently large[2,3]it suggest that small training set is the real cause of poor performance of classifier. The overlap problem occurs when similar number of training data present in a region of data space for each class.

Supervised machine learning system creates a model to predict class label for unlabeled training set.it is common that intrinsic disproportion present in each class. This is known as class imbalance problem it occurs whenever example of one class is higher than the example of other class. Minority class represent circumscribed concept and other represent the counterpart of that concept. Positive and negative labels are used to denominate the minority and majority classes, respectively.

It is difficult to identify in which situation skewed dataset may cause degradation of performance in order to develop new tool or redesign of algorithm to deal with this problem.to perform this task artificial data set may be used which can be easily controlled. For instance, using artificial data sets [2] showed that class imbalance is a relative problem depending on both the complexity of the concept and the overall size of the training set. Furthermore, in previous work [5] using artificial datasets, we showed that performance degradation of imbalanced domains is related to the degree of data overlapping between classes.

The other domain suffer from class imbalance are target detection, fault detection and fraud detection etc. The large number of approach has been previously proposed to face the class imbalance problem. The approach can be categorized as internal approach.it means modifying the existing algorithm or creating new algorithm to face the problem [23].

The external approach that uses unmodified existing algorithm but resample the data using the algorithm the external approach is categorized into two approaches. First, what best data is needed in training dataset? Second what best proportion of positive and negative example included in training data set

ISSN: 2278 – 7798

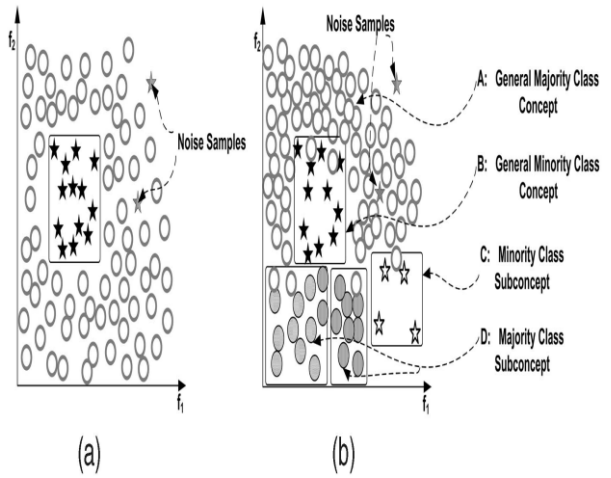

Fig.1 (a) A data set with a between-class imbalance. (b) A high complexity data set with both between-class and within-class imbalances, multiple concepts, overlapping, noise, and lack of representative data. [6]

The imbalance dataset exhibit highly imbalance is referred as between-class imbalance where in each case one class out represent the another class [6]. There are multi classes data in which imbalance exist between various classes] the direct result of the nature of data space is referred as intrinsic imbalance. Imbalanced dataset which depend on variable factor such as time and storage is referred as extrinsic imbalance.

In this paper further section deals with the efficient methods and how to implement the methods to solve the class imbalance and class overlap problem are briefly described.

2. METHODS TO FACE IMBALANCE

2.1 SMOTE

Propose an oversampling approach which is used to oversample the minority class by creating synthetic examples. This approach was successful in character recognition the synthetic example can be generated by operating in feature space not in data space [7]. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. The k nearest neighbor is randomly taken based on the amount of oversampling required. The synthetic example can be generated by the following steps calculate the difference between the feature sample and its nearest neighbor. Then multiply this difference by random number between 0 and 1, finally add it to feature sample this causes the selection of a random point along the line segment between two specific features. The machine learning algorithms used in smote techniques are:

C4.5: C4.5 release 8 is used as a base classifier [11], to compare the various combination of SMOTE and under-sampling with plain under sampling

Ripper: Ripper is used as a base classifier [8] to compare various combination of SMOTE and under-sampling with plain under-sampling We also varied Ripper's loss ratio from 0.9 to 0.001 (as a means of varying misclassification cost) and compared the effect of this variation with the combination of SMOTE and under-sampling[9][10] . To build a set of rules for the minority class the loss ratio can be reduced from 0.9 to 0.001

Naive Bayes Classifier: the prior of minority class can be varied to make the navie bayes classifier cost- sensitive. The prior of minority class can be varied from 1 to 50 times the majority class and compared with C4.5's SMOTE and under-sampling combination

2.2 NEURAL NETWORK APPROACH

Several neural models have been applied for this purpose [12] the error back propogation algorithm [14] is one of the widely used supervised classifier to train the multilayer perceptron network (MLP) the problem in the classifier are duration and reliability of training process. Several techniques are there in speeding up the training of MLP have been proposed [16][17] only few authors addressed the problem of training an MLP by using imbalanced data set [18][19].

The benefit of neural network is flexibility in modeling non–linear association between input variable and target variable. Neural network are modeled on human brain [20]. A MLP consists of input layer (for all input variable contain neuron) a hidden layer (consists of number of hidden neurons) and an output layer (one neuron).each neuron process its input and transmits its output value to the neuron in subsequent layer. Back propogation is the popular algorithm for learning process.in feed- forward network the weights are initialized to small random numbers [13]. Through the network each training instance sent and from each unit the output is computed. The target output is compared with the estimated output of network by calculating the error which is feedback through the network.

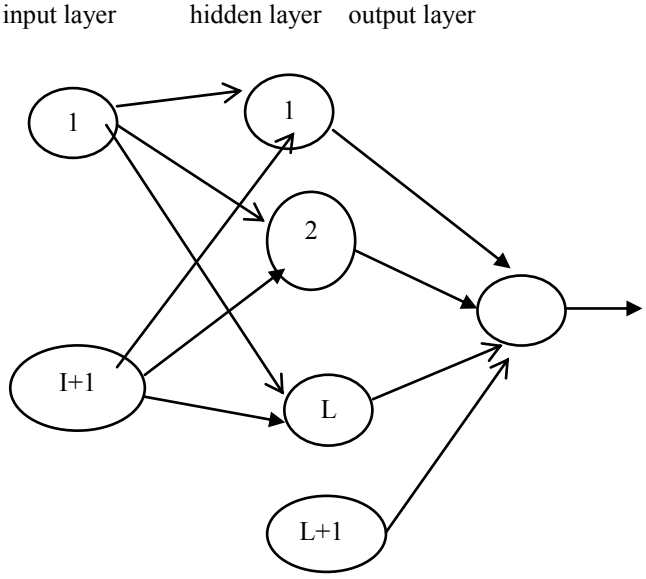

Fig 2. Architecture of feed forward network

ISSN: 2278 – 7798

The output of hidden neuron i is computed by activation function to the weighted inputs and its bias term b_i

$$h_i = f^{(1)}(b_i^{(1)} + \sum_{j=1}^{M} w_{ij} x_j) . \quad (1)$$

where W represents a weight matrix in which Wij denotes the weight connecting input j to hidden neuron i. in output layer the activation function can be predicted by logistic activation function to obtain response

$$f^{(2)}(x) = 1/(1+e^{-x})$$
$$\pi = f^{(2)}(b^{(2)} + \sum_{j=1}^{nh} vjhj) \quad (2)$$

here nh represent the hidden neuron and v is weight vector where vj represent the connection between hidden neuron and the output neuron.to minimize the objective function, eg : sum of squared errors the weights of network are randomly initialized and iteratively adjusted the error value can be calculated by

$$Ej(u) = 1/N \sum_{i=1}^{n_j} \sum_{p=1}^{j} (t^i_p - z^i_p)^2 \quad (3)$$

consider training dataset with two class(j=2),here N is number of samples in class,t_p is desired output and z_p is actual output in the network of sample. The overall mean square error can be determined by

$$E(u) = E_1(u) + E_2(u). \quad (4)$$

The cost function (ע) can be used to balance the error $E(u) = ע(1)\ E_1(u) + ע(2)\ E_2(u)$. The cost function ע(j) reduces its impact in the data distribution probability because the cost function value is diminished gradually.

3 METHODS TO SOLVE CLASS OVERLAP

3.1 EDITING TECHNIQUE

Class overlapping is defined as when the decision region intersect occur. It is highly misclassified by classifier. This problem can be addressed by instance selection methods [21].the edition algorithm is defined as using the instance approach to remove the noise or the points that do not agree with neighbors. The most popular editing methods are based on nearest neighbor rule

The proximity graph (PG) is used to obtain some editing algorithms. The kind of proximity graph are minimum spanning tree, Delaunay triangulation, Gabriel graph and relative neighborhood graph are used to solve geometric problem in various domains [22].

The proximity graph G = (V, E) is an u directed graph with set of vertices V=X where X={x1, x2…. xn} be a set of points in R^d, here n is number of prototype and d is dimensionality of the feature space, and E represent the set of edges (xi, xj) if and only if xi and xj satisfy some neighborhood relation. Let d(.,..) be the Euclidean distance in R^d

The gabriel graph is defined as

$$(xi , xj) \in E \Leftrightarrow d^2(xi,xj) \le d^2(xi,xk) + d^2(xj,xk)$$
$$\text{for all } x_k \in X,\ k \neq i,j$$

here xi and xj are said to be gabriel neighbors

The GGE can be summarized as follows:

1. Construct the Gabriel graph for each sample p in training dataset
2. If and only if all its graph neighbors are its same class then keep p in the training dataset
3. If p belongs to majority class, then discard p from training dataset
4. After eliminating all majority class sample then change the rule 2 by 5
5. To keep p in the TDS only if the majority graph neighbors are of its same class.

3.2 RESAMPLING APPROACH

Resampling was performed using the following strategy first the oversampling is performed by copying the existing training instance in random and added to the training dataset then under sampling is performed by removing existing training instance in random until it was fully balanced[23] .

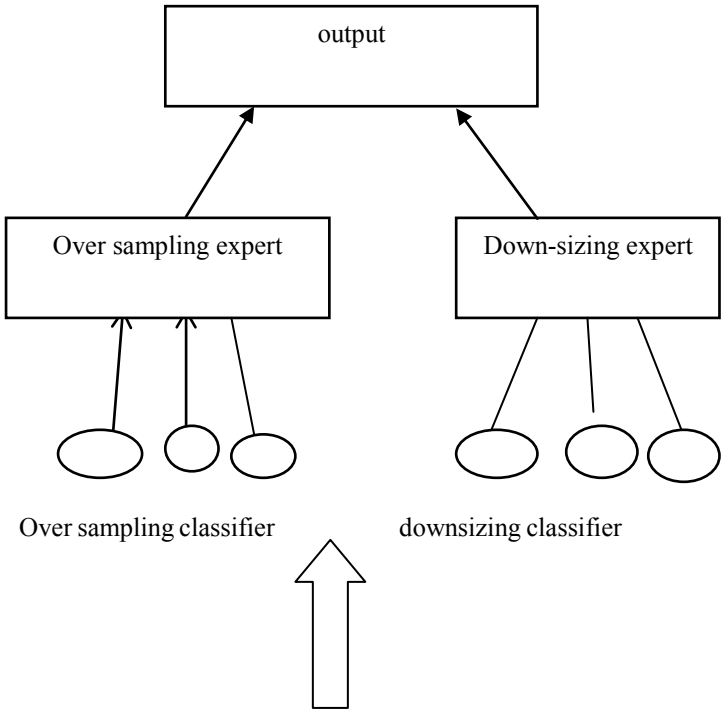

Fig 3.The architecture of multiple resampling methods [23]

ISSN: 2278 – 7798

4 PERFORMANCE MEASURE

The performance of learning classifier from imbalanced dataset can be measured they are 1. Minimum cost criterion 2. The criterion of maximum geometric mean 3. The criterion of maximum sum 4. The criterion of receiver operating characteristic analysis. [24]

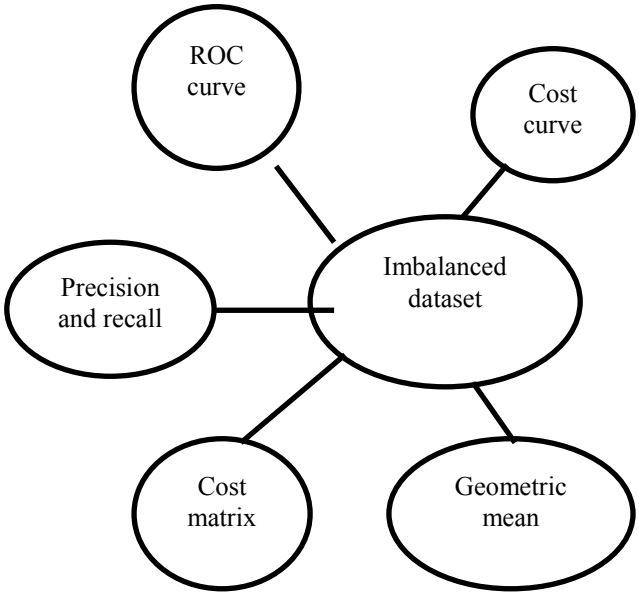

Fig 4. Performance metrics for imbalanced dataset

In the confusion matrix Table 1, *TN* is the number of negative examples correctly classified (True Negatives), *FP* is the number of negative examples incorrectly classified as positive (False Positives), *FN* is the number of positive examples incorrectly classified as negative (False Negatives) and TP is the number of positive examples correctly classified (True Positives).

Table 1: confusion matrix [25]

Actual/predicted	Predicted negative	Predicted positive
Actual negative	TN	FP
Actual positive	FN	TP

Accuracy = (TP+TN)/(TP+FN+FP+TN)
FP rate = FP/ (TN+FP)
TP rate = Recall = TP/ (TP+FN)
Precision = TP/ (TP+FP)

4.1 ROC Curves

Receiver operating characteristics(ROC) curve is used measure the performance of classifier over the range of true positive and false positive error rates.
On Roc curve,
X-axis represents %FP = FP/ (TN + FP)
and the,
Y-axis represents %TP =TP/ (TP + FN)

4.2 Precision and Recall

Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
recall and *precision* goals can be often conflicting, since when increasing the true positive for the minority class, the number of false positives can also be increased; this will reduce the precision [26].

4.3 Geometric Mean

It's one of standard performance measures used in an imbalanced dataset classifier. The reason of using Gmean is to balance the ratio of prediction between majority and minority class. The proportion of G-mean shows that how good an imbalanced dataset classifier predicts the classes. The G-mean is deliberate as follows:
G - Mean = $\sqrt{TNR * TPR}$
where,
TNR =TN / (TN + FP)
TPR =TP / (TP + FN)
where,
TP, *FN*, *FP* and *TN* can be defined as follows. True Positive (*TP*) refers to correctly prediction of the majority class. False Negative (*FN*) refers to wrongly prediction of the minority class as majority class. False Positive (*FP*) refers to wrongly prediction of majority class as minority class. True Negative (*TN*) refers to correctly prediction of minority class.

5. CONCLUSION

Thus this paper represent the characteristics of imbalanced dataset problem ,and provide a technique to solve the imbalance problem.in this paper it gives an idea of classification of imbalanced dataset And also we had gone through the evaluation metrics of the imbalance dataset.in this paper we discuss solution and algorithm to imbalanced dataset. Thus this paper might be very useful for the researchers to know about the imbalanced dataset problems and its solution

REFERENCE

1. Garcia, V., S´anchez, J.S., Mollineda, R.A.: An empirical study of the behaviour of classifiers on imbalanced and overlapped data sets. In Rueda, L., Mery, D.,Kittler, J., eds.: CIARP. Volume 4756 of Lecture Notes in Computer Science., Springer (2007) 397–406
2. Japkowicz, N.: Class imbalances: are we focusing on the right issue. In: Workshop on Learning from Imbalanced Data Sets II. (2003) 17–23
3. Jo, T., Japkowicz, N.: Class imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter 6(1) (2004) 4049
4. Denil, M., Trappenberg, T.P., 2010. Overlap versus imbalance. In: Canadian Conference on AI, pp. 220–231.
5. R. C. Prati, G. E. A. P. A. Batista, and M. C. Monard. Class Imbalances versus Class Overlapping: an Analysis of a Learning System Behavior. In 3rd Mexican International Conference on Artificial Intelligence (MICAI'2004), volume 2971 of LNAI,

ISSN: 2278 – 7798

pages 312–321, Mexico City, 2004. Springer-Verlag.
6. He, H., Garcia, E., 2009. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21 1263–1284.
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.
8. Cohen, W. W. (1995b). Fast Effective Rule Induction. In *Proc. 12th International Conference on Machine Learning*, pp. 115–123 Lake Tahoe, CA. Morgan Kaufmann.
9. Cohen, W. W., & Singer, Y. (1996). Context-sensitive Learning Methods for Text Categorization. In Frei, H.-P., Harman, D., Sch¨auble, P., &Wilkinson, R. (Eds.), *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pp. 307–315 Z¨urich, CH. ACM Press, New York, US.
10. Lewis, D., & Catlett, J. (1994). Heterogeneous Uncertainity Sampling for Supervised Learning. In *Proceedings of the Eleventh International Conference of Machine Learning*, pp. 148–156 San Francisco, CA. Morgan Kaufmann.
11. Quinlan, J. (1992). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.
12. Bruzzone, L., Serpico, S., 1997. Classification of imbalanced remote-sensing data by neural networks. Pattern Recognition Lett. 18, 1323–1328.
13. Brown, I., Mues, C., 2012. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Systems Appl. 39 (3), 3446– 3453.
14. Hertz, J., Krogh, A., Palmer, R.G., 1991. Introduction to the Theory of Neural Computation. Addison-Wesley, The Advance Book Program, Reading, MA.
15. Serpico, S., Bruzzone, L., Roli, F., 1996. An experimental comparison of neural and statistical non-parametric algorithms for supervised classification of remote-sensing images. Pattern Recognition Letters 17, 1331–1341.
16. Tollenaere, T., 1990. SuperSAB: Fast adaptive back propagation with good scaling properties. Neural Networks 3, 561–573.
17. Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T., Alkon, D.L., 1988. Accelerating the convergence of the back-propagation method. Biological Cybernetics 59, 257–263
18. Anand, R., Mehrotra, K.G., Mohan, C.K., Ranka, S., 1993. An improved algorithm for neural network classification of imbalanced training sets. IEEE Trans. Neural Networks 4, 962–969.
19. Anand, R., Mehrotra, K.G., Mohan, C.K., Ranka, S., 1995. Efficient classification for multiclass problem using modular neural networks. IEEE Trans. Neural Networks 6, 117–124.
20. Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford, UK: Oxford University Press.
21. Olvera-López, J., Carrasco-Ochoa, J., Martínez-Trinidad, J., Kittler, J., 2010. A review of instance selection methods. Artif. Intell. Rev. 34, 133–143.
22. Sánchez, J.S., Pla, F., Ferri, F.J., 1997. Prototype selection for the nearest neighbour rule through proximity graphs. Pattern Recognition Lett. 18 (6), 507–513
23. Estabrooks, A., Jo, T., Japkowicz, N., 2004. A multiple resampling method for learning from imbalanced data sets. Comput. Intell. 20 (1), 1836.
24. Nitesh V. Chawla," Data mining for imbalanced datasets: an overview", *IN 46530, USA*.
25. Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, "Handling imbalanced datasets: A review", GESTS International Transactions on Computer Science and Engineering, Vol.30, 2006.
26. Yongqing Zhang, Danling Zhang, Gang Mi, Daichuan Ma, Gongbing Li , Yanzhi Guo, Menglong Li,Min Zhu, "Using ensemble methods to deal with imbalanced data in predicting protein–protein interactions", Computational Biology and Chemistry 36 , 2012, 36–41.

ISSN: 2278 – 7798