IJETIE International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

Minority Oversampling Technique for Imbalanced Dataset Learning Using Agglomerative Clustering

R.Anitha[1]
[1]PG Scholar, Computer science and Engineering,
P.S.R.Engineering College, Sivakasi, Tamilnadu.
mailtoaniharshad@gmail.com

S.Santhi[2]
[2]Assistant Professor, Computer science and Engineering,
P.S.R.Engineering College, Sivakasi, Tamilnadu.
santhi@psr.edu.in

Abstract--- Unequal distribution of data examples is due to imbalanced learning problem. To solve this problem synthetic oversampling methods were introduced. These methods generate synthetic samples to balance the distribution among different classes. Many of the synthetic sample generation methods produce wrong samples. A new method namely Minority Oversampling Technique for Imbalanced Dataset Learning Using Agglomerative Clustering. This method first finds the hard-to-learn informative minority samples and then assigns weight to the minority class samples based on the Euclidean distance which are nearer to the majority class samples. Then the synthetic samples are generated for the informative minority class samples which are weighted by using clustering approach.

Index terms--- Oversampling, clustering, imbalanced learning, under sampling, synthetic sample generation

1 INTRODUCTION

IMBALANCED learning problem will cause unequal distribution of data samples between various classes where most of the samples belong to one class and few samples belong to another class if the classes are two. If most of the samples are from one class that class is said to be majority class. The other is called as minority. A data set is called imbalanced if it contains many more samples from one class than from the rest of the classes. For research community learning from imbalanced data is important since it is used in many real world problems such as medical diagnosis [1], information retrieval systems [2], detection of fraudulent telephone calls [3], detection of oil spills in radar images [4], data mining from direct marketing [5]. In fraud detection problem [3], the ratio between the number of majority class and minority class samples is about as high as 1:100. This ratio is known as imbalanced ratio. To reduce the classification error is the main goal of any classifier. That is to maximize the overall accuracy of the classifier.

An imbalanced learning problem creates a big challenge to the classifier, because it is very hard to learn the minority class samples [6], [7], [8]. Thus the classifier favors the majority class samples, resulting in a large classification error over minority class samples from the imbalanced data [9]. The identification of the minority class samples is crucial and it makes a high cost [1], [2], [3], [4], [5]. Hence the classifier learned from the imbalanced data has to perform equally on both the majority and minority class samples.

An imbalance that occurs between the samples of two classes is called as between-class imbalance. The bad performance of classifiers on the minority class samples is not only due to between-class imbalance. It also depends on the within class imbalance and small disjunts problems [9], [10], [11]. The complexity of data samples can be another factor for the poor performance of the classifier [9]. Most popular approaches that deal with imbalanced learning problems are based on the synthetic oversampling methods [10], [11].

The synthetic oversampling methods deal with the imbalanced learning problems [12]. From the overall view of many synthetic methods it is clear that all methods failed to give correct synthetic minority samples. These samples will make the learning task so difficult. The minority oversampling technique for imbalanced dataset learning using agglomerative clustering algorithm generates useful synthetic samples. This algorithm includes three steps:

1. Selects the appropriate subset of original minority class samples.
2. Assigns weight to the samples based on their importance.
3. Then useful synthetic samples are generated using clustering approach.

2 RELATED WORKS

More number of works is done in imbalanced dataset learning. There are four methods to do a work. They are,

1. Sampling-based methods
2. Cost-based methods
3. Kernel-based methods
4. Active learning-based methods.

A single method itself does generate the synthetic samples. Now-a-days sampling methods are very useful in generating the synthetic samples. Sampling methods are very successful in recent years.

Sampling methods mainly concentrate on the distribution of samples among majority and minority classes. The information get by a classifier from a balanced dataset is more than from an imbalanced dataset. It is

IJETIE International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

difficult for good classification by a classifier. In the survey there are two different sampling methods are present. They are oversampling and under sampling. Under sampling methods reduces the majority class samples. If the reduction of samples done in random it is called as random under sampling. If it is done by using statistical knowledge then it is called as informed under sampling. Some of the informed under sampling methods also uses data cleaning techniques to refine the majority class samples.

Over sampling methods adds the minority class samples to the imbalanced dataset. The over sampling is categorized into two, Random over sampling and Synthetic over sampling. Random over sampling is a non-heuristic method which adds samples through the random replication of minority class samples. This type of over sampling technique creates very specific rules which lead to over fitting. In other hand the synthetic over sampling technique add samples by generating the synthetic minority class samples. It can add required information to the original dataset using the generated samples which in turn improves the classifier's performance. Various methods are there to generate the synthetic samples like SMOTE [12], Borderline-SMOTE [13], and ADASYN [14].

Over sampling method can cause extra bias to the classifiers, which in turn leads to decrease in the classifiers' performance. This can be solved by introducing boosting methods. It is clear that the over sampling method is more useful than under sampling. It also improves the classifiers' performance for complex data [12]. Other methods rather than sampling also work comparably well. Hence there is no a single method for all scenarios. Comparing over sampling with under sampling there occur a difference. In under sampling, the method may remove the essential information from the original data whereas not in over sampling.

3 MOTIVATION

Synthetic over sampling methods are successful in dealing with imbalanced data [12], [13], [14], [15]. Although it is successful there exists some insufficiencies and inappropriateness of the existing methods that may occur in many different scenarios. We describe them in this section.

The synthetic oversampling methods, for example, Borderline-SMOTE [13], are used to identify the border-line minority class samples (also called the seed samples). The oversampling method uses these seed samples for generating the synthetic samples because they are most likely to be misclassified by a classifier. So that Borderline-SMOTE generates the synthetic samples in the neighborhood of the border. A minority class sample is identified as a borderline sample [13], if the number of the majority class samples, m, among its k-nearest neighbors satisfies

$$\frac{k}{2} \leq m < k \qquad (1)$$

Where k is a user-specified parameter. The criterion mentioned above (1) may fail to identify the borderline samples.

Over sampling methods like ADASYN [14] and RAMOBoost [15] try to avoid the synthetic sample generation problem by adaptively assigning weights to the minority class samples. They enhance the chance for the minority class sample as a participant in the synthetic sample generation process. To assign weight here parameter δ is used to define the number of majority class samples among the k-nearest neighbors of the minority class sample. The use of δ for assigning the weights may cause some problems like:

1. The parameter δ is inappropriate for assigning weights to the minority class samples located near the decision boundary.
2. The parameter δ is insufficient to distinguish the minority class samples with regard to their importance in learning.
3. The parameter δ may favor noisy samples.

The existing synthetic over sampling methods [12], [13], [15] uses the k-nearest neighbor approach. To generate synthetic samples for a minority class sample x, the k-NN approach randomly selects another sample y from the k-nearest of x. This approach then generates a synthetic sample g and it is expressed as

$$g = x + (y - x) \times \alpha \qquad (2)$$

Where α (2) is a random number in the range 0 to 1. The above equation says that g will lie in the line segment between x and y. However, in many cases, the k-NN-based approach may generate wrong minority class samples.

4 PROPOSED METHOD

Considering the above mentioned problems a new minority over sampling method is proposed. The objective of this method is i) to improve the efficiency of sample selection scheme and ii) to improve the efficiency of synthetic sample generation scheme. The algorithm consists of three steps,

1. Selection of an appropriate subset of the original minority class samples.
2. Assigning weights to the selected samples according to their importance in the data.
3. Using a clustering approach (agglomerative) for generating the useful synthetic minority class samples.

Algorithm:

Input:

1. Smaj: Set of majority class samples.
2. Smin: Set of minority class samples.
3. N: Number of synthetic samples to be generated.
4. k1: Number of neighbors used for predicting noisy minority class samples.
5. k2: Number of majority neighbors used

IJETIE
International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

for constructing informative minority set.

6. k3: Number of minority neighbors used for constructing informative minority set.

Procedure:

1. For each minority example $x_i \in S_{min}$, compute the nearest neighbor set, NN (x_i). NN (x_i) consists of the nearest k_1 neighbors of x_i according to euclidean distance.
2. Construct the filtered minority set, S_{minf} by removing those minority class samples which have no minority example in their neighborhood: $S_{minf}= S_{min} -\{x_i \in S_{min}: NN(x_i)$ contains no minority example$\}$
3. For each $x_i \in S_{minf}$, compute the nearest majority set, $N_{maj}(x_i)$. $N_{maj}(x_i)$ consists of the nearest k_2 majority samples from x_i according to euclidean distance.
4. Find the borderline majority set, S_{bmaj}, as the union of all $N_{maj}(x_i)$s, i.e.,$S_{bmaj}=Ux_i \in S_{minf}N_{maj}(x_i)$.
5. For each majority example yi $\in S_{bmaj}$, compute the nearest minority set, $N_{min}(y_i)$. $N_{min}(y_i)$ consists of the nearest k_3 minority examples from yi according to euclidean distance.
6. Find the informative minority set, S_{imin}, as the union of all $N_{min}(y_i)$s, i.e., $S_{imin} = U_{yi \in Sbmaj}N_{min}(y_i)$.
7. For each $y_i \in S_{bmaj}$ and for each $x_i \in S_{imin}$, compute the information weight, $I_w(y_i,x_i)$.
8. For each $x_i \in S_{imin}$, compute the selection weight $S_w(x_i)$ as $S_w(x_i)=\sum_{yi \in Sbmaj} I_w(y_i,x_i)$.
9. Convert each $S_w(x_i)$ into selection probability $S_p(x_i)$ according to $S_p(x_i)= S_w(x_i)/ \sum_{zi \in Simin} S_w(z_i)$.
10. Find the clusters of S_{min}. Let, M clusters are formed which are L_1; L_2; . . . ; L_M.
11. Initialize the set, $S_{omin} = S_{min}$.
12. Do for j = 1 . . .N.
    a) Select a sample x from S_{imin} according to probability distribution $\{S_p (x_i)\}$. Let, x is a member of the cluster L_k, 1<=k<=M.
    b) Select another sample y, at random, from the members of the cluster L_k.
    c) Generate one synthetic data, s, according to
    s= x +α*(y-x),
    where α is a random number in therange [0,1].
    d) Add s to S_{omin}: $S_{omin} = S_{omin} U \{s\}$.
13. End Loop

4.1 Construction of the set S_{imin}

The algorithm filters the original minority set, S_{min}, to find a filtered minority set, S_{minf}. To do this, compute NN (x_i) for each $x_i \in S_{min}$. Then remove each x_i if its NN (x_i) contains only the majority class samples. The removed minority class sample is likely to be noisy because it is surrounded only by the majority class samples. This removal prohibits the noisy minority class sample to take part in the synthetic sample generation process. Thus, the algorithm will be able to remove existing noisy samples from the given data.

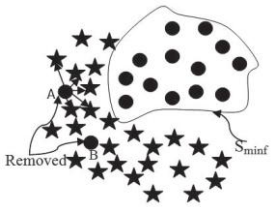

Fig 4.2.1 Noise removed

For each $x_i \in S_{minf}$, construct $N_{maj}(x_i)$.The samples in $N_{maj}(x_i)$ will be the borderline majorities and expected to be located near the decision boundary when k_2 is small. Then combine all the $N_{maj}(x_i)$'s to form the borderline majority set,S_{bmaj}.

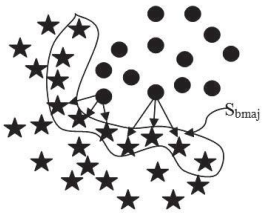

Fig 4.2.2 Borderline majority set

For each $y_i \in S_{bmaj}$, construct $N_{min}(y_i)$ and combine all such $N_{min}(y_i)$'s to form S_{imin}

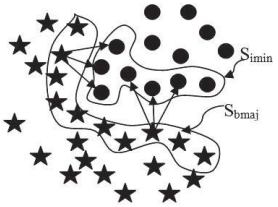

Fig 4.2.3 Informative minority set

4.3 Finding the Weights for the Members of S_{imin}

The weight is calculated for each minority class sample $x_i \in S_{imin}$ and it is called the selection weight, $S_w(x_i)$. This paper computes this weight based on the following three important observations:

Observation 1: Samples close to the decision boundary contain more information than those of further. This observation indicates that the closer samples should be given a higher weight than the further ones.

IJETIE International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

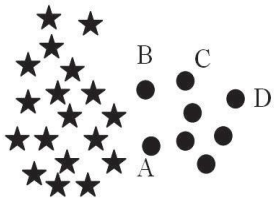

Fig 4.3.1 Weightage for closer samples

In the above figure samples A and B are closer to the decision boundary compared to those of C and D. Therefore, A and B are more informative than C and D. Similarly, C is more informative than D. It implies that A and B need to be given a higher Sw than C and D. At the same time, C is to be given a higher S_w than D.

Observation 2: The minority class samples in a sparse cluster are more important than those in a dense cluster. From the perspective of the synthetic sample generation, the members of a sparse cluster are more important than those of a dense cluster. This is due to the fact that the dense cluster contains more information than the sparse cluster. Thus, the sparse cluster requires more synthetic samples to increase its size for reducing within-class imbalance. This is why the members of the sparse cluster deserve higher S_w's than those of the dense cluster.

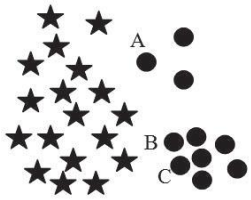

Fig 4.3.2 Weightage for sparse cluster

In the above figure sample A is more important than B and C though all of them are close to the decision boundary. This is obvious because A is a member of the sparse cluster, while B and C are the members of the dense cluster.

Observation 3: The minority class samples near a dense majority class cluster are more important than those near a sparse majority class cluster.

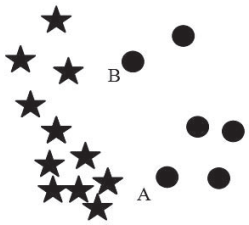

Fig 4.2.3 Weightage to samples near dense majority cluster

In the above figure the density of the majority class neighbors near to A is higher than that to B. This relative imbalance will make difficulty for a classifier to correctly learn A. Because of this A is more important for the synthetic sample generation than B and its S_w will be higher than that of B.

It is now understood that S_w is to be computed by considering the aforementioned observations. This paper considers them and employs the majority class set S_{bmaj} in computing S_w, which can be described as follows:

1. Each majority class sample $y_i \in S_{bmaj}$ gives a weight to each minority class sample $x_i \in S_{imin}$. This weight is called the information weight, $I_w(y_i, xi)$.
2. For x_i, we sum up all the $I_w(y_i, x_i)$s to find its selection weight, $S_w(x_i)$. This can be expressed as,

$$S_w(x_i)=\sum_{yi \in Sbmaj} I_w(y_i, x_i) \quad (3)$$

4.4 Synthetic Sample Generation

This paper finds the clusters of the minority data set, S_{min}, using a modified hierarchical clustering algorithm to overcome issues in k-NN . Then the oversampled minority data set, S_{omin}, is initialized by S_{min}.Finally the synthetic minority class samples are generated using the above equation and they are added to S_{omin}. Although the sample generation approach described here and the k-NN-based approach used in the existing methods employ the same equation our approach differs with the others in the way how y in the equation is chosen. Here they choose y from the members of the x's cluster ,while the k-NN-based approach randomly selects it from the k-nearest neighbors. It is obvious that if x and y reside in the same cluster, then according to equation the generated synthetic samples will also lie inside the same cluster. This is beneficial in the sense that the generated samples will never erroneously fall in the majority class region. Another benefit of cluster-based approach lies when the synthetic samples are generated from the noisy minority class samples. If x is a noisy sample, then it will likely form an isolated cluster, consisting of only one member (itself). In this case, y selected by this approach will be the same as x and the equation will then generate a duplicate sample, i.e., x. This is much better than the k-NN-based approach that may generate the noisy minority class samples, which in turn will enlarge the minority class region. An erroneously enlarged region is likely to overlap with the majority class region, creating difficulty in learning.

4.5 Clustering S_{min}

In the fourth step, our algorithm uses average-linkage agglomerative clustering, a hierarchical clustering process. Agglomerative clustering does not require the number of clusters to be fixed a priori. It generates clusters in a bottom up fashion.

5 CONCLUSION

Furthermore, it is able to generate correct synthetic samples. The samples closer to decision boundary are given higher weights than others. Similarly, the samples of the small-sized clusters are given higher weights for reducing within-class imbalance. The synthetic sample generation

technique uses a clustering approach. The aim of using clustering is to ensure that the generated samples must reside inside the minority class area for avoiding any wrong or noisy synthetic sample generation.

6 FUTURE WORK

Data sets with continuous features only are used here.So, the algorithm can be generalized to handle features of any type.It can be investigated whether some other clustering mechanism can give better performance for the algorithm. A number of parameters are involved in this algorithm which can be optimized for the best performance depending on the specific problem at hand.

7 REFERENCES

[1] P.M. Murphy and D.W. Aha, "UCI Repository of Machine Learning Databases," Dept. of Information and Computer Science,Univ. of California, Irvine, CA, 1994.
[2] D. Lewis and J. Catlett, "Heterogeneous Uncertainty Sampling for Supervised Learning," Proc. Int'l Conf. Machine Learning, pp. 148-156, 1994.
[3] T.E. Fawcett and F. Provost, "Adaptive Fraud Detection," Data Mining and Knowledge Discovery, vol. 3, no. 1, pp. 291-316, 1997.
[4] M. Kubat, R.C. Holte, and S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images," Machine Learning, vol. 30, no. 2/3, pp. 195-215, 1998.
[5] C.X. Ling and C. Li, "Data Mining for Direct Marketing: Problems and Solutions," Proc. Int'l Conf. Knowledge Discovery and Data Mining, pp. 73-79, 1998.
[6]G.M. Weiss, "Mining with Rarity: A Unifying Framework," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 7-19, 2004.
[7] R.C. Holte, L. Acker, and B.W. Porter, "Concept Learning and the Problem of Small Disjuncts," Proc. Int'l Joint Conf. Artificial Intelligence, pp. 813-818, 1989.
[8] J.R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.
[9] N. Japkowicz and S. Stephen, "The Class Imbalance Problem: A Systematic Study," Intelligent Data Analysis, vol. 6, no. 5, pp. 429-449, 2002.
[10] 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," Proc. Mexican Int'l Conf. Artificial Intelligence, pp. 312-321, 2004.
[11] T. Jo and N. Japkowicz, "Class Imbalances versus Small
Disjuncts," ACM SIGKDD Exploration Newsletter, vol. 6, no. 1,pp. 40-49, 2004.
[12]N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer,"SMOTE: Synthetic Minority oversampling Technique," J. Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.
[13] H. Han, W.Y. Wang, and B.H. Mao, "Borderline-SMOTE: A New Oversampling Method in Imbalanced Data Sets Learning," Proc. Int'l Conf. Intelligent Computing, pp. 878-887, 2005.
[14] H. He, Y. Bai, E.A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," Proc. Int'l Joint Conf. Neural Networks, pp. 1322-1328, 2008.
[15] S. Chen, H. He, and E.A. Garcia, "RAMOBoost: Ranked Minority Oversampling in Boosting," IEEE Trans. Neural Networks, vol. 21, no. 20, pp. 1624-1642, Oct. 2010.

International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org