Novel Approach for Classification On Imbalanced Dataset

Amee Rajan[1] and Chetna Chand[2]

[1] PG Student ,Department of computer Engineering, Kalol Institute of Technology & Research Centre ,Kalol , Gujarat , India
[2] Asst. Prof. ,Department of computer Engineering , Kalol Institute Of Technology & Research Centre ,Kalol , Gujarat , India

Abstract

Real world application suffers from imbalanced dataset. There have been many attempts at dealing with classification of imbalanced data sets. Classification of imbalanced dataset is an evoloving trend in research area of data mining. we can categorise classification methods in three categories as data-level approach ,algorithm level approach and cost sensitive approach.Data level approach mainly synthesize or remove instances to force the sizes of each class comparable, which may change the inherent data structure or introduces noise to the source data . In this paper we have studied algorithm level approach Hybrid Coupled K Nearest Neioghbours and have work on improving its performance .Results show that our proposed method weighted HC-Knn gives better performance compared to HC-knn.

Keywords: *Imbalanced Data, undersampling, oversampling, cost-sensitive approach , algorithm level approach , HC-knn*

1.Introduction

Class imbalance problem is a hot topic being investigated recently by machine learning and data mining researchers. It can occur when the instances of one class is more than the instances of other classes. The class have more instances called the majority class while the other called minority class. However, in many applications the class has lower instances are the more interesting and important one. The imbalance problem heightens whenever the class of interest is relatively rare and has small number of instances compared to the majority class. Moreover, the cost of misclassifying the minority class is very high in comparison with the cost of misclassifying the majority class for example; consider cancer versus non-cancer or fraud versus un-fraud [1].The patient could lose his/her life because of the delay in the correct diagnosis and treatment. Similarly, if carrying a bomb is positive, then it is much more expensive to miss a terrorist who carries a bomb to a flight than searching an innocent person.

Many real world applications such as medical diagnosis, fraud detection (credit card, phone calls, insurance), network intrusion detection, pollution detection, fault monitoring, biomedical, bioinformatics and remote sensing (land mine, under water mine) suffer from these phenomena. In this scenario, classifiers can have good accuracy on the majority class but very poor accuracy on the minority class(es) due to the influence that the larger majority class has on traditional training criteria.

Many research papers on imbalanced data sets have commonly agreed that because of this unequal class distribution, the performance of the existing classifiers tends to be biased towards the majority class. The reasons for poor performance of the existing classification algorithms on imbalanced data sets are: 1).They are accuracy driven i.e.,their goal is to minimize the overall error to which the minority class contributes very little. 2). They assume that there is equal distribution of data for all the classes. 3).They also assume that the errors coming from different classes have the same cost. With unbalanced data sets, data mining learning algorithms produce degenerated models that do not take into account the minority class as most data mining algorithms assume balanced data set[2].

A number of solutions to the class-imbalance problem were previously proposed both at the data and algorithmic levels [2]. At the data level, these solutions include many different forms of re-sampling such as random oversampling with replacement, random undersampling, directed oversampling (in which no new examples are created, but the choice of samples to replace is informed rather than random), directed undersampling (where, again, the choice of examples to eliminate is informed),oversampling with informed generation of new samples, and combinations of the above techniques. At the algorithmic level, solutions include adjusting the costs of the various classes so as to counter the class imbalance, adjusting the probabilistic estimate at the tree leaf (when working with decision trees), adjusting the decision threshold, and recognition-based (i.e., learning from one class) rather than discrimination-based (two class) learning.

2. Related Work

One of the most famous over-sampling methods is SMOTE[3]. It over-samples the minority class by taking each minority class sample and introducing synthetic examples along the line segments joining all of the k minority class nearest neighbors, Cost-Sensitive Learning is a type of learning in data mining that takes the misclassification cost into consideration. For example, CCPDT[4], which is designed for imbalanced situation, is a modification of the decision tree algorithm. The cost-sensitive learning incorporate approaches at the data level, algorithmic level or at both levels, considering higher costs for the misclassification of examples of the positive class with respect to the negative class, and trying to minimize higher cost errors . Although *k*NN has been identified as one of the top ten most influential data mining algorithms[13], the standard *k*NN algorithm is not suitable for imbalanced class distribution. To improve the performance of *k*NN for imbalanced classification several modifications has done. Such as in 2011 Yuxuan Li and Xiuzhen Zhang presented kENN which proposed a training stage where exemplar positive training instances are identified and generalized into Gaussian balls as concepts for the minority class. This paper propose to identify exemplar minority class training instances and generalize them to Gaussian balls as concepts for the minority class [5].Again In 2011 Wei Liu and Sanjay Chawla proposed CCW-kNN a novel *k*-nearest neighbors (*k*NN) weighting strategy is proposed for handling the problem of class imbalance. CCW-kNN uses the probability of attribute values given class labals to weight prototypes in kNN[6].

In 2014 Chunming Liu et al. proposed HC-*k*NN which considers relationship between features when computing the similarity. It calculates sized membership using fuzzy theory to deal with imbalance data problem then assigns feature weight to every feature .hybrid coupled k-nearest neighbor classification algorithm (HC-*k*NN) works on mixed type data, by doing discretization on numerical features to adapt the inter coupling similarity as we do on categorical features, then combing this coupled similarity to the original similarity or distance,then *k* nearest neighbors that correspond to the *k* highest similarity values are choosen. The most frequently occurred class in the *k* neighbors is assigned [7].

Algorithm : Hybrid Coupled *k*NN Algorithm

```
Input: An instance $u_t$ without label and a source labeled dataset $D\{u_1; u_2; :::; u_n\}$
Output: The class label of $u_t$
For each class, initiate the sized membership of class using the fuzzy set theory
Do discretization on numerical features
Calculate the feature weight of every feature
Create the similarity matrix which contains both intra and inter similarity for dataset $D$
Calculate the distance of $u_t$ to every instance in dataset $D$ using the adapted similarity
Select top $k$ points which are close to the instance $u_t$
Return the class label of those $k$ neighbors which has the maximum number of instances
```

3. Proposed Work

Hc-Knn has disadvantage that it is sensitive to different values of k ,which degrades its performance.so to reduce the influence of the sensitivity of the selection of the neighborhood size *k* to some degree and yield the good performance in classification , we are assigning weight[8] based on similarity calculation to each k neighbors. The weight drops quickly from 1 at the distance of the first nearest neighbor to 0 at the distance of the furthest *k*-th nearest neighbor. Algorithm of our proposed Algorithm is as follows.

Algorithm of proposed Work

Input: An instance u_e without label and a source labeled dataset D{u_1; u_2; :::; u_n}
Output: The class label of u_e

1: For each class, initiate the sized membership of class using the fuzzy set theory

The Sized Membership of Class denotes the rate of a class c_l that belongs to the minority. The sized membership of class describes how small a class is.

$$\theta(c_l) = 1 - \frac{|c_l|}{m} \tag{1}$$

Where $|c_l|$ is the number of instances in classes c_l and *m* is the total number of instances in the data set.

2: Do discretization on numerical features

$$CAIM(C,D|F) = \frac{\sum_{r=1}^{n}(max_r^2/M_{+r})}{n} \tag{2}$$

where *n* is the number of intervals, *r* iterates through all intervals, and $max_r$is the maximum value within the r^{th} column of the quanta matrix, $M_{+r}$is the total number of continuous values of attribute F that are within the interval (d_{r-1}; d_r].

The algorithm starts with a single interval that covers all possible values of a continuous attribute, and divides it iteratively. From all possible division points that are tried it chooses the division boundary that gives the highest value of the CAIM criterion[9].Discretization results are used in following Feature weight and Inter-Similarity calculation.

3: Calculate the feature weight of every feature

The feature weight describes the importance degree of each categorical feature

$$\alpha_j = \begin{cases} \sum_{i=1}^{m} \frac{fre\left(x_{ij,}R^{c(u_i)}\right)}{m.\left|R^{c(u_i)}\right|} & if\ \left|Unique\left(f_j\right)\right| > 1 \\ 0 & if\ \left|Unique\left(f_j\right)\right| = 1 \end{cases} \tag{3}$$

where *m* is the total number of instances in the data set, x_{ij} is the *j* feature value for instance u_i , $R^{C(ui)}$ consists of all the instances which share the same class as instance u_i, and the according instance number is $|R^{C(ui)}|$, while Fre($x_{ij,}$ $R^{C(ui)}$) defines as a frequency count function that count the occurrences of x_{ij} in feature j of set $R^{C(ui)}$, and $|Unique(f_j)|$ returns the category number or discretization interval number in feature j.If all the values in a feature are the same, that is, $|Unique(f_j)|$ = 1,then this feature cannot be used in Classification task so we set the weight to be zero.

4: Create the similarity matrix which contains both intra and inter similarity for dataset *D*

In intra-similarity calculation on numerical features Euclidean distance is used, and if the inter-similarity calculation relates to numerical features, same strategy on its discretization result as we do on categorical features.

Intra Coupled Similarity

$$\delta^{Ia}\left(v_j^x, v_j^y\right) = \frac{RF\left(v_j^x\right).RF\left(v_j^y\right)}{RF\left(v_j^x\right)+RF\left(v_j^y\right)+RF\left(v_j^x\right).RF\left(v_j^y\right)} \tag{4}$$

where RF(v^x_j) and RF(v^y_j) are the relative occurrence frequency of values v^x_j and v^y_j in feature a_j , respectively. For numerical features, we use 1/*Euclidean* as the feature values' Intra-similarity

Inter Coupled Similarity

$$\delta_{i|j}^{Ie}\left(v_i^x, v_j^y\right) = \frac{f\left(v_p^{xy}\right)}{max\left(RF\left(v_i^x\right),RF\left(v_i^y\right)\right)} \tag{5}$$

Where $f\left(v_p^{xy}\right)$ is the co-occurrence frequency count function with value pair v_p^{xy} , and $RF(v_i^x)$ and $RF\left(v_i^y\right)$is the relative occurrence frequency in their features respectively.

Adapted Coupled Object Similarity

$$AS(u_{i1}, u_{i2}) = \sum_{j=1}^{n}\left[\beta.\, \alpha_j \delta_j^{Ia}\left(v_j^{i_1}, v_j^{i_2}\right) + (1-\beta).\sum_{k=1,k\neq j}^{n} \delta_{j|k}^{Ie}\left(v_j^{i_1}, v_k^{i_2}\right)\right] \quad (6)$$

where β ∈ [0 , 1] is the parameter that decides the weight of intra-coupled similarity, $v_j^{i_1}$ and $v_j^{i_2}$ are the values of feature j for instances u_{i1} and u_{i2} respectively.
δ_j^{Ia} and $\delta_{j|k}^{Ie}$ are the intra-coupled feature value similarity and inter-coupled feature value similarity.

5: Calculate the distance of u_e to every instance in dataset *D* using the adapted similarity and sort the distance.

$$IS(\,u_e, u_i) = \theta\big(c(u_i)\big).AS(u_e, u_i) \quad (7)$$

where u_e and u_i are the instances, respectively; $C(u_i)$ denotes the class of u_i; denotes the class of u_i; $\theta(\cdot)$ is the sized membership of class defined in Step (1); and $AS(\cdot)$ is the adapted coupled object similarity defined in Step (4).

6: Select top *k* points which are close to the instance

7: Calculating weight:

```
For i =1 to k do
    If $IS(u_e, u_k) \neq IS(u_e, u_1)$ then
        $w_i = \frac{IS(u_e,u_k) - IS(u_e,u_i)}{IS(u_e,u_k) - IS(u_e,u_1)} \times \frac{IS(u_e,u_k) + IS(u_e,u_1)}{IS(u_e,u_k) + IS(u_e,u_i)}$    (8)
    Else
        $w_i = 1$
    End if
End for
```

8: Assign a majority weighted voting class label

$$\bar{y} = \underset{y}{\arg\max} \sum_{(u_i, y_i)\in T} w_i \times \delta(y = y_i) \quad (9)$$

In our proposed work, similarity calculated in step 7 is used in calculating weight of k nearest neighbors .it assigns weight from 1 at the first nearest neighbor to 0 at furthest *k*-th nearest neighbor. Then in last step, it assigns the majority weighted class, where y represents class label.

Flow Of Proposed Work

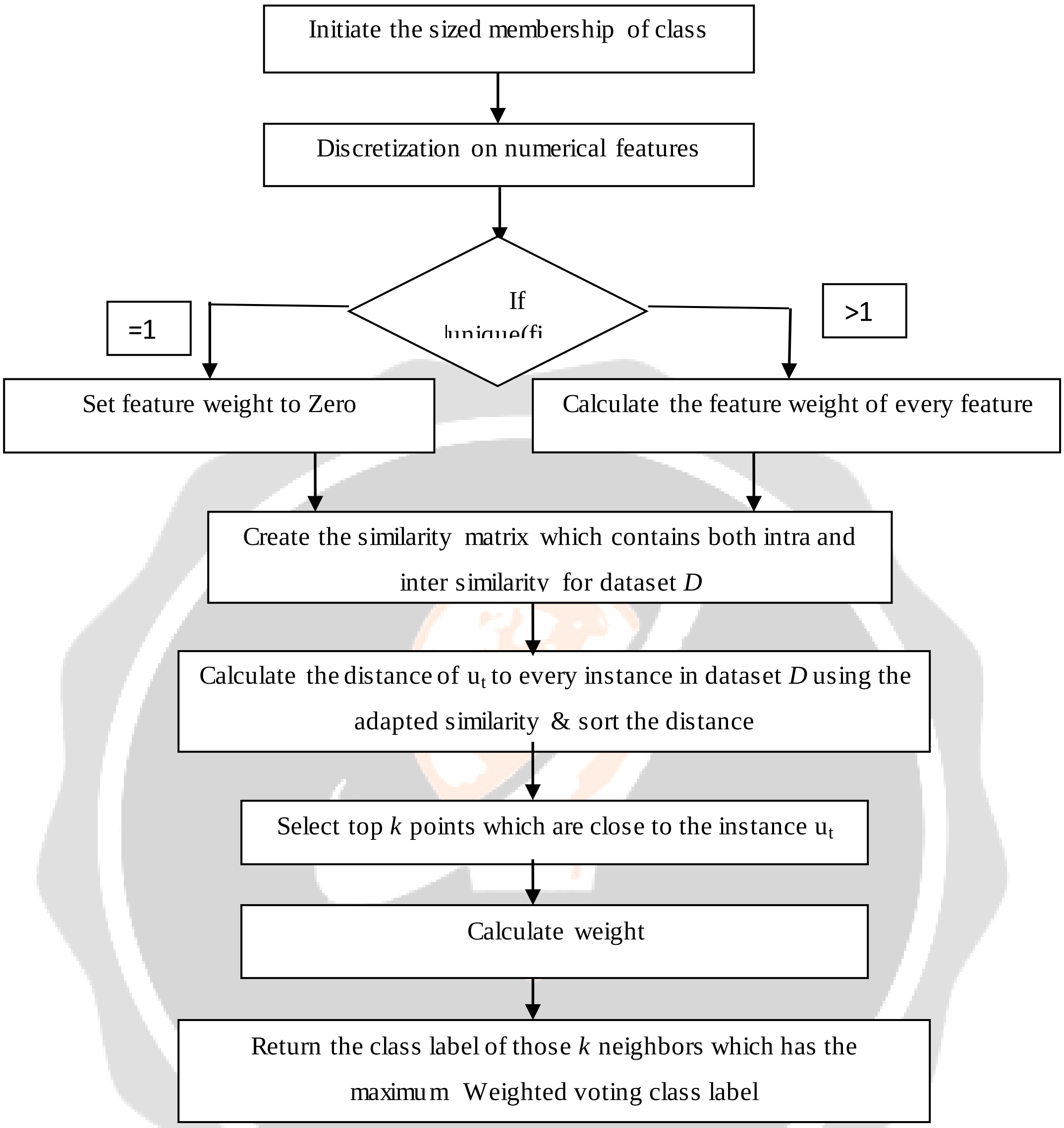
Initiate the sized membership of class
Discretization on numerical features
If
|unique(fi
=1
>1
Set feature weight to Zero
Calculate the feature weight of every feature
Create the similarity matrix which contains both intra and inter similarity for dataset D
Calculate the distance of u_t to every instance in dataset D using the adapted similarity & sort the distance
Select top k points which are close to the instance u_t
Calculate weight
Return the class label of those k neighbors which has the maximum Weighted voting class label

Fig 1. Flow Of Proposed Work

4. Experimental Result

Due to the dominative effect of the majority class, the overall accuracy is not an appropriate evaluation measure for the performance of classifiers on imbalanced datasets, we use Receiver Operating Characteristic (ROC) curve and the Area Under the ROC Curve (AUC)[10,12] to evaluate the performance results. Table IV shows the AUC results for 4 UCI Dataset[11] : Lymphography ,dermatology, annealing and contraceptive .From the table it is clear that our proposed algorithm achieves better AUC.Fig.2 shows AUC result for different minority rate and our proposed algorithm Whc-Knn performs better than Hc-knn. Fig .3 shows that for different Value of k ,our proposed Algorithm Performs Better than HC-knn.

Index	Dataset	Source	#instance	#(N+C)Features	#Class	Minority Name	Minority(%)
D1	Lymphography	UCI	148	(3+15)	4	Normal	1.35%
D2	Annealing	UCI	898	(6+32)	5	U	4.26%
D3	Dermatology	UCI	366	(1+33)	6	P.R.P	5.46%
D4	Contraceptive	UCI	1473	(2+7)	3	Long-term	22.61%

TABLE 1: Dataset Description

Dataset	Minority(%)	AUC	
		Whc_Knn	Hc-Knn
D1	1.35%	0.9932	0.75
D2	4.26%	0.9406	0.9369
D3	5.46%	0.9999	0.9825
D4	22.61%	0.6271	0.5893

TABLE 2: AUC comparison

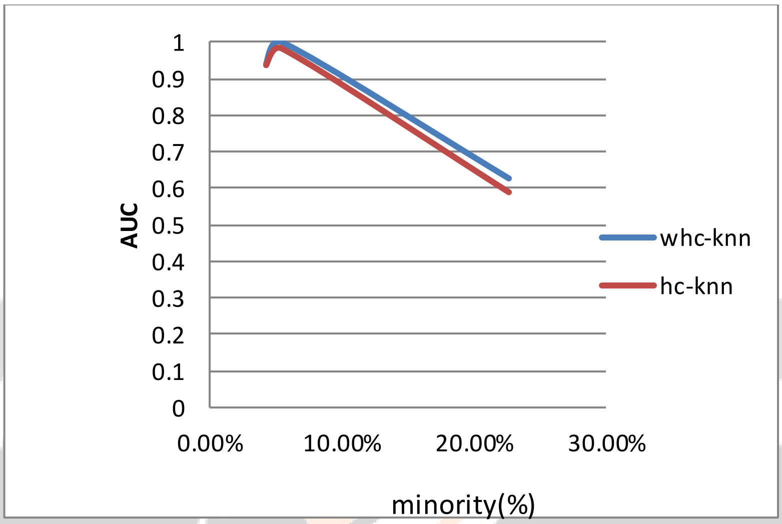

Fig 2. Sensitivity to imbalance Rate

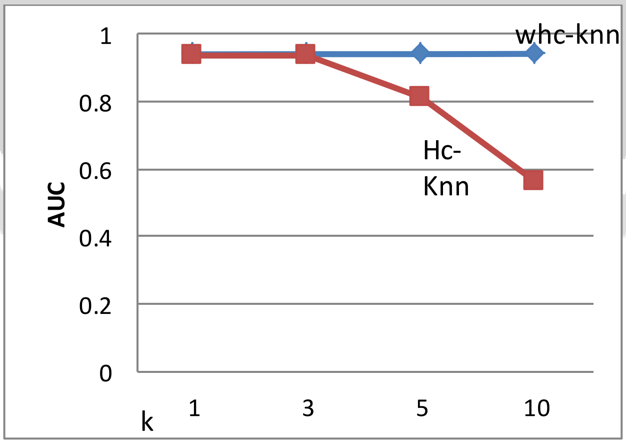

Fig 3. AUC via neighbor size k

5. CONCLUSION AND FUTURE WORK

Class imbalance have wide range of application including medical, fraud detection, network intrusion detection, pollution detection, fault monitoring, biomedical, bioinformatics and remote sensing .we have work on improving accuracy of HC-Knn ,which is algorithm level approach for handling imbalance issue .we have work on sensitivity on different values of k in HC-knn by assigning weights on nearest neighbors. Results shows that our proposed approach gives higher accuracy. Future work will include lowering the time complexity, and applying this idea to other basic classification algorithms based on similarity or distance.

REFERENCES

[1] Shaza M. Abd Elrahman and Ajith Abraham, “A Review of Class Imbalance Problem”, *Journal of Network and Innovative Computing*, Volume 1, 2013, pp. 332-340

[2] Vaishali Ganganwar, “An overview of classification algorithms for imbalanced datasets” ,*International Journal of Emerging Technology and Advanced Engineering*, Volume 2, Issue 4, April 2012

[3]N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[4] W. Liu, S. Chawla, D. Cieslak, and N. Chawla, “A robust decision tree algorithm for imbalanced data sets,” in *SDM 2010*, 2010, pp. 766–777.

[5] Yuxuan Li and Xiuzhen Zhang, “Improving k nearest neighbor with exemplar generalization for imbalanced classification,” Springer-Verlag Berlin Heidelberg 2011

[6] Wei Liu and Sanjay Chawla ,“Class confidence weighted knn algorithms for imbalanced data sets”, Springer-Verlag Berlin Heidelberg 2011

[7]Chunming Liu , Longbing Cao and Philip S Yu ,“A Hybrid Coupled k-Nearest Neighbor Algorithm on Imbalance Data.”, *International Joint Conference on Neural Networks (IJCNN)*, IEEE 2014

[8] Jianping Gou , Lan Du , Yuhong Zhang , Taisong Xiong “A New Distance weighted *k*-nearest Neighbor Classifier” **,** *Journal of Information & Computational Science*, June 2012

[9] L. A. Kurgan and K. J. Cios, “CAIM discretization algorithm,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 2, pp. 145–153, 2004.

[10] T. Fawcett, “An introduction to ROC analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[11] K. Bache and M. Lichman, “UCI machine learning repository,” 2013.

[12] Victoria Lopez a, Alberto Fernandez b, Salvador Garcia b, Vasile Palade c and Francisco Herrera a ”An insight into classification with imbalanced data: Empiricalresults and current trends on using data intrinsic characteristics” ,Elsevier 2013

[13]X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip *et al.*, “Top 10 algorithms in data mining,” *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008