A Hybrid Weighted Nearest Neighbor Approach to Mine Imbalanced Data

Harshita Patel[1], G.S. Thakur[2]
[1,2]Department of Computer Applications,
Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India, 462003

Abstract -*Classification of imbalanced data has drawn significant attention from research community in last decade. As the distribution of data into various classes affects the performances of traditional classifiers, the imbalanced data needs special treatment. Modification in learning approaches is one of the solutions to deal with such cases. In this paper a hybrid nearest neighbor learning approach is proposed for mining binary imbalanced datasets. This hybridization is based on different K for different classes as large K for large classes and small K for small classes and with weighted concept as small weights for large classes and large weight for small classes. The merger of dynamic K improves the performance of weighted approach.*

Keywords: Nearest neighbors, imbalanced data, classification, weights.

1 Introduction

The well developed era of science and technology facilitated the world in huge manner, also results in production of immense amount of data. This ever increasing data need proper treatment for use. Data mining, data science and machine learning are some name of research solutions to treat such data. With every new day many challenges are being solved with various new approaches; learning from imbalanced data is one of them and become a part of interest of data researchers from more than a decade. In imbalanced datasets classes are unequally distributes as some classes overwhelmed by others in terms of number of instances [5][9]. Some examples of imbalanced datasets are Credit Card Fraud Detection [12], Oil-spill Detection [7], Medical Diagnosis [13], Cultural modeling [19], Text Categorization, Network Intrusion, Helicopter Gearbox Fault Monitoring [5] etc.

Previous researches has shown that the presence of imbalance in data causes decreased and many times inaccurate performances of traditional classifiers such as Decision Trees, SVM, Naïve-Bayes and Nearest Neighbors which results in biased classification and may be severe in many cases like for medical data. Four common ways to deal with the problems are (i) balance data by using sampling techniques; i.e. under sampling or over sampling, (ii) modification in traditional classification algorithms, (iii) cost sensitive approaches and (iv) ensemble techniques. In this paper second solution of algorithm modification is adopted for nearest neighbor classification in terms of different *K* value for different classes in weighted nearest neighbor algorithms for imbalanced datasets.

K nearest neighbor algorithm comes under top 10 data mining algorithms [20] due to its simplicity, easy programming, implementation, comprehensiveness and robustness. Its error rate is bounded above by twice the Bayes error rate [16][3]. Like other traditional classifiers, K nearest neighbor also suffers from less accurate and biased performance issue in presence of imbalance in datasets. There are many alternative nearest neighbor approaches have been developed for such cases. Some approaches were proposed for text classification or categorization. Average of Similarity-*K*NN (AS-*K*NN) is proposed by Yang et. al [21], in this approach the sum of similarity of each class is divided by the total number of instances in the nearest neighbor. One algorithm Nearest-Weighted *K*NN (NW-*K*NN) is proposed by Tan [6] in which small size classes are assigned with larger weights and large classes assigned with small weights. Adaptive *K*NN (ADPT-*K*NN) is proposed by Baoli et. al [14] performs on different values of *K* for each class on varying number of training data. The value of *K* is large for larger classes and small for smaller classes. Another method is Drag-Pushing KNN (DP-*K*NN) [15] in which weights of features of classes are increased or decreased in dealing with misclassified data.

Also a lot of work has been done on other real world problems and generalized for any dataset. Numerous modifications on *K*NN have been proposed with weight concepts too. A fine background was provided by algorithms like called WDNN (Weighted Distance Nearest Neighbor) [8] and WDkNN [17] for weighted nearest neighbor approach. WDNN preserves important data points by assigning positive weight values. While WDNN works with on k=1, WDkNN is a next improved step which works on values greater than 1 of k. Chawla et. al. have been shown efficient working of CCW (Class Confidence

Weights) [18] as it considering attribute probability of given class. Dubey et al. [4] proposed another weighted *K*-nearest neighbor algorithm. They considered class distribution for neighbors of any test instance. Initial classification taken from traditional *K*-nearest neighbor algorithm that how it classified an instance and used to calculate the weight for each class. Kriminger et. al. [2] proposed a single class algorithm to decrease the effect of imbalance using the local geometric structure in data. This algorithm applies to diverse degrees of inequality and in addition able to work on any number of classes. It also allows adding new examples to training data sets. Chen et. al. [10] have concentrated on the impact of various measures cost ratio, imbalance ratio and sample size on classification results of a French bankruptcy database and found that such measures have severe impact on the classification performance. Tomasev et. al. [11] discussed about the hubness effect related to *K*-nearest neighbors in high-dimensional datasets that result in high misclassification rate due to minority class examples unlike in small and medium dimensional datasets where misclassification occur due to majority class examples. Ryu et. al. [1] proposed an instance hybrid selection using nearest neighbor (HISNN) for cross-project defect prediction (CPDP) where class imbalance exists in distributions of source and target projects. In this algorithm, local information is learned by *K*-nearest neighbor algorithm while naive Bayes is applied to gain global information. This hybridization results in high performance in software defect prediction.

In this paper a hybrid *K* nearest neighbor is proposed for binary classification by combining the ideas of dynamic *K* for different classes with large and small weights for small and large classes. The structure of this paper contains basic learning in preliminaries in section II, followed by proposed algorithm in section III. Section IV contains experiments and result discussions and paper is concluded in section V.

2 Preliminaries

2.1 *K*-Nearest Neighbor

The *K*-nearest neighbor algorithm finds class label for any instance *q* from test dataset, by finding *K* (some integer) nearest neighbors of *q* from training dataset and then assign the class label for which *q* will have maximum neighbors. It could be shown by following equation:

$$C(q) = \arg\max_{C \in \{Cj \mid j=1,2\}} \sum_{y_i \in S(q,K)} T(y_i, C)$$

Here $C(q)$ = *class label of q, to be predicted,*
m = *Number of classes,*
$S(q,K)$ = *Set of* $K-$*nearest neighbors of q and*

$$T(y_i, C) = \begin{cases} 1 & \text{if } y_i \in C \\ 0 & \text{otherwise} \end{cases}$$

2.2 Adaptive *K*-Nearest Neighbor

Baoli et. al. (2004) [6] proposed the Adaptive *K* nearest neighbor for dealing with imbalanced text corpus. The main idea behind this research is to define different *K* for different classes. The *K* is finding out according to sizes of classes. Altering *K* with classes is giving more accurate categories rather than static *K* for all classes; because uneven distribution of data into classes affects the classification performance.

Selection of K_{Cj} (for particular class) for classes is done using following equation

$$K_{Cj} = \min\left(\lambda + \left\lceil \frac{K * I(C_j)}{\max\{I(C_j) \mid j=1,2\}} \right\rceil, K, I(C_j)\right)$$

Here K = *Original input integer to define nearest neighbors,*
K_{Cj} = *Calculated K for each class* C *using above formula,*
$I(C_j)$ = *Number of instances in class* C_j *where* $j = 1$ *and* 2,
λ = *Constant Integer value.*

2.3 Neighbor Weighted *K*-Nearest Neighbor

Neighbor Weighted K nearest Neighbor approach was proposed by Tan [14] for imbalanced text datasets. The basic concept used in this method is to assign large weight to small classes and small weights to large classes to reduce the biasness of the classifier towards majority class and ignorance of minority one. Weights defined through this method help to improve the performance of nearest neighbor classifier in presence of imbalance in datasets.

First we find *K* nearest neighbors for query instance q, find weight from following equation:

$$W_i = \frac{1}{(N(C_j) / Min\{N(C_j) \mid j=1,2\})^{1/p}}$$

p is an exponent and *p*>1,

And then weight is applied to traditional classification algorithm.

$$C(q) = \arg\max_{C \in \{Cj \mid j=1,2\}} W_i \left(\sum_{y_i \in S(q,K)} T(y_i, C) \right)$$

3 Proposed Method

The proposed nearest neighbor approach merges the concepts of weights and specific *K* of each class in any binary imbalance data space where one class is overwhelmed with the other one. The parameter *K* plays an important role in the performance of the nearest neighbor classifier. In presence of imbalance in data space it become more vital as one class dominant the other with more instances. Sometimes minority class contain crucial information as happened in many real world applications, but due to having less or very few representative instances, classical methods unable to classify them correctly. In *K* nearest neighbor common *K* for all classes biases the result towards the majority class. Because in case of common *K* for all classes may implies large number of nearest neighbors consideration for classification for majority and minority both and due to less quantity, minority class instances will be classified as majority class instances. Adaptive approach [6] suggests the dynamic *K*, with the size of classes. It improves the accuracy of classification for minority class too.

Neighbor weighted method is another solution for imbalanced datasets. This method suggests assigning weights according to size of classes, large weights for small classes and small weights for large classes. Both approaches were proposed for imbalanced text corpus to categorize the imbalance text more accurately. These methods perform well with numeric data too. Combining the both methods give better performance by applying varying *K* on neighbor weight approach.

Algorithm

Input: Training Dataset *D*, Query instance *q*, Set of class labels *C* and Parameter *K*.

Output: Class label of *q*.

Step 1. Find K nearest Neighbor for q

Step 2. Obtain weights with

$$W_x = \frac{1}{(N(C_x) / Min\{N(C_j) \mid j = 1,2\})^{1/p}}$$

Step 3. Find K_{Cj} for classes by using equation

$$K_{Cj} = \min\left(\lambda + \left\lceil \frac{K * I(C_j)}{\max\{I(C_j) \mid j = 1,2\}} \right\rceil, K, I(C_j) \right)$$

Step 4. Determine class label of ***q*** with

$$C(q) = \underset{C \in \{Cj \mid j=1,2\}}{\arg\max} \; W_x \Big(\sum_{y_i \in S(q, K_{Cj})} T(y_i, C) \Big)$$

4 Experiments and Results

4.1 Datasets

The experiments with proposed approach are performed for binary classification. Seven imbalanced datasets have been taken with different imbalance ratio. All datasets are processed with their all features and effect of individual feature on classification is not considered so assuming that all features are essential; no feature selection is applied here. Short description of datasets used is given in following table:

TABLE1
DESCRIPTION OF IMBALANCED DATASETS

Datasets	Source	# Instances	Class (1/0)	#Attributes	IR
Phoneme	KEEL	5404	Oral/Nasal Sound	5	2.4
Transfusion	UCI	748	Blood Doneted Yes/No	4	3.2
Vehicle0	KEEL	846	Positive/ Negaive	18	3.3
Yeast-2_vs_4	KEEL	514	Positive/ Negaive	8	9.1
Glass2	KEEL	214	Positive /Negaive <=4/>4	9	11.6
Wine Quality	UCI	4898	Wine Quality >20/<=20	11	25.8
Abalone	UCI	4177	rings	7	66.4

4.2 Evaluation Measures

Evaluating accuracy is not enough for imbalanced datasets. It is possible that a classifier may achieve good accuracy results but accuracy rate would be higher for majority class and very less for minority class due to less number of instances present in dataset. Some known and trendy evaluation measures for classifiers of imbalanced data are F-measure, AUC, G-Mean etc.

The classification results could be seen in terms of confusion metrics:

		Actual Values	
Predicted Values		TP (True Positive)	FP (False Positive)
		FN (False Negative)	TN (True Negative)

Fig 1. Confusion metrics for performance evaluation

TP is the value of actual positives which are correctly classified as positives. FP is the value of actual negatives which are incorrectly classified as positives. Similarly TN is the value of actual negatives that are correctly classified as negatives. FN is the value of actual positives that were incorrectly classified as negatives. Accuracy measures could be shown in terms of these metrics values:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$\Pr esicion = \frac{TP}{TP + FP}$$

$$\mathrm{Re}\,call = \frac{TP}{TP + FN}$$

$$F - Measure = \frac{2 * \Pr ecision * \mathrm{Re}\,call}{\Pr ecision + \mathrm{Re}\,call}$$

F-measure is a popular evaluation measure to evaluate classification performance for imbalanced data and for other machine learning tasks. Table 2 describes the F-measure values of neighbor weighted approach and our hybrid method:

TABLE II
F-MEASURE FOR DIFFERENT VALUES OF *K*

Dataset	K	NWKNN	Hybrid Approach
Phoneme	5	0.4548	0.4548
	10	0.4551	0.4562
	15	0.454	0.4549
	20	0.4497	0.4551
	25	0.4503	0.4568
Transfusion	5	0.3118	0.375
	10	0.2429	0.3523
	15	0.1923	0.3978
	20	0.225	0.4277
	25	0.0606	0.4314
Vehicle0	5	0.3491	0.381
	10	0.3698	0.4096
	15	0.3465	0.4
	20	0.3629	0.4152
	25	0.339	0.4138
Yeast-2_vs_4	5	0.1429	0.1529
	10	0.1606	0.1678
	15	0.1626	0.2029
	20	0.1897	0.1986
	25	0.1622	0.2
Glass2	5	0.1	0.0938
	10	0.0408	0.1449
	15	0.0465	0.127
	20	0.0541	0.1667
	25	0.0606	0.1449
Wine Quality	5	0.0526	0.0636
	10	0.0466	0.0721
	15	0.0432	0.0721
	20	0.0418	0.0721
	25	0.0383	0.0721

The table shows the improvement in performance in terms of F-measure. Table III shows the AUC for neighbor weighted and our method; though it is still very less but significantly improved with the neighbor weighted approach.

TABLE III
AUC FOR DIFFERENT VALUES OF *K*

Dataset	K	NWKNN	Hybrid Approach
Phoneme	5	0.5017	0.5017
	10	0.5031	0.5044
	15	0.5022	0.5027
	20	0.4963	0.5031
	25	0.4988	0.5069
Transfusion	5	0.4677	0.5168
	10	0.4545	0.509
	15	0.4738	0.5667
	20	0.532	0.6049
	25	0.4865	0.6004
Vehicle0	5	0.4696	0.5079
	10	0.5063	0.5541
	15	0.4801	0.5381
	20	0.5064	0.5644
	25	0.4828	0.5619
Yeast-2_vs_4	5	0.4062	0.43235
	10	0.46735	0.48275
	15	0.48085	0.57455
	20	0.5429	0.56375
	25	0.48705	0.56735
Glass2	5	0.3593	0.3254
	10	0.2356	0.5
	15	0.28645	0.44235
	20	0.3373	0.57625
	25	0.3712	0.5
Wine Quality	5	0.392	0.4512
	10	0.3821	0.5
	15	0.3883	0.5
	20	0.4002	0.5
	25	0.4026	0.5

5 Conclusions and Future Work

In this paper adaptive *K* is merged with neighbor weighted approach and it helps to improve the classification performance of neighbor weighted approach. Neighbor weighted approach was in itself classifying well the

imbalanced data. But with the use of small K for small class and large K for large class, classification results get improved. The research was done with two class dataset and it could be further performed with multiclass datasets. AUC is better than the neighbor weighted approach and needs more improvement. Enhancement of the proposed algorithm could be more refined with better performances.

6 References

[1] D. Ryu, J. Jang and J. Baik "A hybrid instance selection using nearest-neighbor for cross-project defect prediction", *Journal of Computer Science and Technology*, vol. 30, no. 5, pp. 969-980, 2015.

[2] E. Kriminger and C. Lakshminarayan, "Nearest neighbor distributions for imbalanced classification", *In Proc. of WCCI 2012 IEEE World Congress on Computational Intelligence*, Brisbane, pp. 10-15, 2012.

[3] G. Loizou and S. J. Maybank. "The nearest neighbor and the bayes error rates", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 9, no. 2, pp. 254-262, 1987.

[4] H. Dubey and V. Pudi, " Class based weighted k nearest neighbor over imbalanced dataset" *PAKDD 2013, Part II, LANI 7819*, pp. 305-316, 2013.

[5] H. He and E. A. Garcia, "Learning from Imbalanced Data", *IEEE Transactions on Knowledge and Data Engineering*, vol.21, no.9, pp.1263-1284, 2009.

[6] L. Baoli, L. Qin and Y. Shiwen, "An adaptive k nearest neighbour text categorization strategy", *ACM Transactions on Asian Language Information Processing*, vol. 3, no. 4, pp. 215-226, 2004.

[7] M. Kubat, R. C. Holte and S. Matwin, "Machine Learning for the Detection of Oil spills n Satellite images", *Machine Learning*, vol. 30, nos. 2/3, pp. 195–215, 1998.

[8] M. Z. Jahromi, E. Parvinnia and R. John, "A method of learning weighted similarity function to improve the performance of nearest neighbor", *Information Sciences*, vol. 179, pp. 2964–2973, 2009.

[9] N. V. Chawla, *Data Mining for Imbalanced Datasets: An overview*, Data Mining and Knowledge Discovery Handbook, pp.853-867.

[10] N. Chen, A. Chen and B. Ribeiro "Influence of class distribution on cost-sensitive learning: A case study of bankruptcy analysis", *Intelligent Data Analysis*, vol. 17, no. 3, pp. 423-437, 2013.

[11] N. Tomašev and D. Mladenic, "Class imbalance and the curse of minority hubs", *Knowledge-Based Systems*, vol. 53 pp. 157–172, 2013.

[12] P. Chan and S. Stolfo, "Toward scalable learning with non-uniformclass and cost distributions: A case study in credit card fraud detection", *In Proc. of Knowledge Discovery and Data Mining*, 164–168,1998.

[13] R. B. Rao, S. Krishanan and R. S. Niculscu., "Data Mining for Improved Cardiac care", *ACM SIGKDD Exploration Newsletter*, vol.8, no.1, pp. 3–10, 2006.

[14] S. Tan, "Neighbor-weighted K-Nearest Neighbor for unbalanced text corpus", Expert *Systems with Applications,* vol. 28, no. 4, pp. 667–671, 2005.

[15] S. Tan, "An effective refinement strategy for K-Nearest Neighbor text classifier", Expert *Systems with Applications,* vol. 30, no. 2, 290–298, 2006.

[16] T. M. Cover P. E. Hart, "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory*, vol. 13, pp. 21-27, 1967.

[17] T. Yang, L. Cao, C. Zhang, " A novel prototype reduction method for the *K*-nearest neighbor algorithm with $K \geq 1$" *In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V.(eds.), PAKDD 2010. LNCS*, Springer, Heidelberg, vol. 6119, pp. 89–100. 2010.

[18] W. Liu, S. Chawla, " Class confidence weighted knn algorithms for imbalanced datasets" *In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS*, Springer, Heidelberg, vol. 6635, pp. 345–356, 2011.

[19] X. C. Li, W. J. Mao, D. Zeng D, P. Su and F. Y. Wang, "Performance evaluation of machine learning methods in cultural modelling", *Journal of Computer Science and Technology*, vol. 24, no. 6, pp. 1010-1017, 2009.

[20] X. Wu et al., "Top 10 algorithms in data mining", *Knowledge Information Systems*, vol. 14, pp. 1-37, 2008.

[21] Y. Yang, T. Ault, T. Peirce, C. W. Lattimer, "Improving text categorization methods for event tracking", *In proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 65–72, 2000.