An Improved KNN Algorithm Based on Minority Class Distribution for Imbalanced Dataset

Bo Zang, Ruochen Huang, Lei Wang, Jianxin Chen, Feng Tian, Xin Wei
College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China, 210003
Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, China, 210003
Email: {1015010627, 2014010102, wanglei, chenjx, tianf, xwei}@njupt.edu.cn

***Abstract*—K-nearest neighbor (KNN) is a popular classification algorithm with good scalability, which has been widely used in many fields. When dealing with imbalanced data, minority examples are given the same weight as majority examples in the existing KNN algorithm. In this paper, we pay more attention to the minority class than the majority class, and we increase the weight of minority class according to the local characteristic of minority class distribution. In addition, we compare the proposed algorithm with the existing Weighted Distance K-nearest neighbor (WDKNN). Experimental results show that our algorithm performs better than WDKNN in imbalanced data sets.**

Keywords—imbalanced data; KNN; WDKNN.

I. INTRODUCTION

In real life, most datasets to be classified are imbalanced, such as medical diagnosis, failure detection, text classification and etc. Imbalanced datasets refer to those the amount of data in one class is far more than those in the other class. The class containing more data is called majority class while the other is called minority class. In conventional machine learning researches, the training set is balanced and the number of examples is equal in both classes, so it appears that the generalization ability is unsatisfactory when the traditional algorithms are applied to the imbalanced datasets. Therefore, how to improve the classification accuracy and the overall balance of the classifiers has become a hot but difficult issue in the field of machine learning and data mining for imbalanced datasets [1].

At present, the classification problem of imbalanced data sets has been widely concerned by scholars at home and abroad. Research on this problem mainly includes two aspects: 1) processing method of data level, and 2) processing method of algorithm level [2]. Data level is one of the most important ways in dealing with imbalanced data classification problem. The method is mainly divided into two kinds: the under-sampling method for majority class examples and over-sampling for minority class examples. The core idea is to achieve the balanced data by cutting or adding reasonable objective data, which would alleviate the negative effect for imbalanced data classifier. The research on algorithm level improves the classification algorithm mainly according to the characteristics of the imbalanced datasets. The primary means include setting different weights for different classes, changing the probability density distribution, adjusting the classification boundary. Examples are given as cost sensitive learning, support vector machine (SVM), single class learning and integrated methods.

KNN is a common and powerful classification algorithm in the field of data mining, it has been widely employed in many fields due to its simplicity [3]. The nearest neighbor algorithm was first proposed by Cover and Hart in 1967, in which the unclassified examples are assigned to a nearest classified example by the rule [4]. KNN is a simple and intuitive algorithm, which is suitable for almost all kinds of data structures. Example modeling is not required in this algorithm, which makes it easy to access and maintain. However, there are some disadvantages in KNN. On one hand, it has high computational complexity and slow speed, because all the data need to be calculated for global search when a new example requires classification, bringing about slow speed. On the other hand, the accuracy is unstable because the Euclidean distance metric is usually used in this algorithm to calculate the distance between examples. Euclidean distance has high computational complexity especially in high-dimensional data and it cannot describe the distance well when there is some correlation between attributes of examples, which leads to the low classification accuracy.

In view of the above shortcomings, some improved KNN algorithms have been proposed in recent years. Weinberger *et al.* put forward a novel Mahalanobis distance metric instead of Euclidean distance metric, which is called large margin nearest neighbor (LMNN) [5]. Min *et al.* proposed a scalable non-linear feature mapping method to improve KNN algorithm, which is based on deep neural network and called DNet-kNN [6]. Then, Yang *et al.* presented a novel WDkNN algorithm, which is based on the weighted similarity function and maximizes the leave-one-out cross-validation accuracy [7]. These improved algorithms have greatly ameliorated the classification performance of KNN. However, these algorithm do not increase the weight of minority examples in imbalanced datasets and we improve the algorithm at this point.

The rest of this paper is organized as follows: Existing KNN algorithms are introduced in section II. Our proposed

978-1-5090-3438-3/16 $31.00 © 2016 IEEE
DOI 10.1109/ICS.2016.142

method is presented in section III. Section IV reports the experimental procedure and analyzes the corresponding results. Finally, in Section 5 we summarize the study and conclude the paper.

II. Existing KNN algorithms

KNN is a popular classification algorithm with good scalability, so it has been extensively employed in many fields. The core idea of the KNN classification algorithms is to calculate the distance or similarity between the unclassified examples and the training examples with the known class label, then we can find the k nearest neighbors for the examples to be classified according to the distance or similarity [8]. Then the classification of examples to be classified depends on the categories of these nearest neighbors. If all the k nearest neighbors belong to the same class, then the example to be classified also belongs to this class. Otherwise, we can determine the class of examples according to the majority voting mechanism, which is usually expressed as:

$$\begin{aligned} y_t' &= \arg\max_{c\in\{c_1,c_2\}} \sum_{x_i\in\phi(x_t)} I(y_i=c) \\ &= \max\left\{ \sum_{x_i\in(x_t)} I(y_i=c_1),\ \sum_{x_i\in(x_t)} I(y_i=c_2) \right\} \end{aligned} \qquad (1)$$

where c_1 and c_2 are classification labels, y_t' is prediction label, $I(\ )$ is a function that returns 0 when the condition is false and returns 1 in the opposite condition, $\Phi(x_t)$ denotes the nearest neighbors. And in this algorithm, we use Euclidean distance to determine the similarity between examples. The Euclidean distance formula is expressed as:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (2)$$

where $x = (x_1, x_2, \cdots, x_n)$ and $y = (y_1, y_2, \cdots, y_n)$ represent two examples, n is the number of example feature attributes. Because the distance between the unclassified example and the k nearest neighbors are different, they should have different weights, leading to the idea of WDKNN. In this algorithm, nearer neighbors are more reliable, the weight is defined as the inverse distance [7], which is usually expressed as:

$$y_t' = \arg\max_{c\in\{c_1,c_2\}} \sum_{x_i\in\phi(x_t)} I(y_i=c)\frac{1}{d(x_t,x_i)} \qquad (3)$$

where $d(x_t, x_i)$ is the distance between the test example and training example. WDKNN algorithm can improve the classification performance in imbalanced data.

III. The proposed algorithm

A. Motivation

Although many researchers have experimentally shown that standard classifiers have difficulties in identifying the minority class, we can also note that there are still some accurate classifiers in extreme imbalanced situation. In [9,10], the data problem of imbalanced datasets plays the most crucial role in the imbalanced problem. Some experts speculated that the class imbalance rate (the ratio between the majority and minority class sample) is not the only factor that affects the classification performance. In addition to the imbalanced ratio, there may be other factors leading to the deterioration of the classification performance. The minority class does not form a uniform and compact distribution through the study of imbalanced datasets, but they are many small sub-clusters surrounded by a lot of majority class examples. So, these small sub-clusters are difficult to learn and they may cause a classification error. Furthermore, another factor that affects the classification performance of imbalanced data may be the noise example [11]. Experiments show that the single minority examples in majority class cannot be regarded as noise points, because they are too rare to be ignored and the appropriate preprocessing of these minority data can improve the performance of the classifier.

In most experiments, researchers only focus on a single factor, but usually these factors occur together in imbalanced data, and the main problem is that it not easy to determine the influence factor in the real dataset. In this paper, we will pay more attention to the local characteristic of the minority class distribution. We propose a method based on local characteristic of the minority class distribution to increase the weight of minority class, which is beneficial to promote the performance of the classifier.

B. Improved KNN algorithm based on the minority class

As mentioned above, most imbalanced datasets contain many safe majority class examples and unsafe minority class examples, so we propose a method based on the local characteristics of the minority class. Our intention is to change the weight of the minority class examples, which can be obtained by analyzing the local characteristics of a minority example. In this paper, a relatively simple method is proposed, we only calculate the number of majority classes in the neighbor of minority class [12,13].

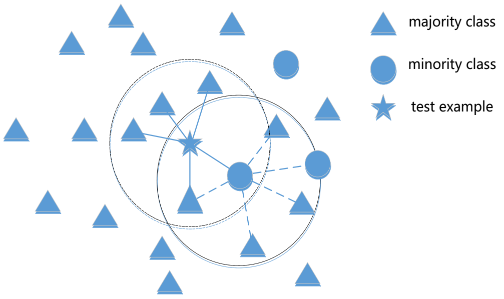

Fig.1. the local characteristics of a minority example

In Fig.1, we can see that if there is minority example in the k nearest neighbors of the test example, we will study the local characteristics of the minority example and find the number of majority class in its nearest neighbors. And the more majority examples, the more weight should be given to the minority example, so we make the definition of minority example weight as follow:

$$L_{\min} = \frac{\left(N_{maj}\right)^{\alpha}}{K} \tag{4}$$

where N_{maj} represents the number of majority class examples, α is a parameter factor, this function is a linear function when $\alpha = 1$, otherwise the function appears as an exponential function. The value of α can be determined by the characteristic of the imbalanced data.

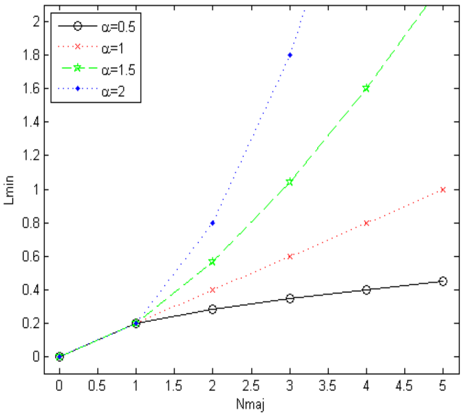

Fig.2. $L_{\min}$ weights depending on α

Fig.2 shows that the value of $L_{\min}$ changes with the value of N_{maj} depending on the different value of α. If we increase the value of α, the weight of minority example will be exponentially amplified. Of course, this amplification may be beneficial for the extremely imbalanced dataset. In this paper, we set the value of α to 1. As the range of N_{maj} is $0 \le N_{maj} \le K$, which may result in the value of $L_{\min} = 0$, we re-formulate it as:

$$w\left(L\right) = (L_{\min} + 1)\lambda \tag{5}$$

where λ is also a technical parameter, which can make the weight normalized. In this paper, we set λ to 1. As a result, the weight of the majority class remains the same, while the weight of minority class is increased. We define the improved classification rule as:

$$\mathrm{y}_t^{'} = \underset{c \in \{c_1, c_2\}}{\arg\max} \sum_{x_i \in \phi(x_t)} I\left(y_i = c\right) \frac{1}{d\left(x_t, x_i\right)} w\left(L\right) \tag{6}$$

Here, we give the concrete steps of improved KNN algorithm based on local characteristic of the minority class distribution as:

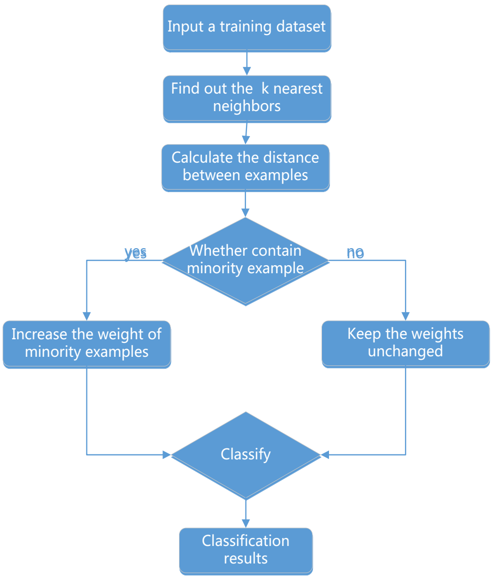
Input a training dataset
Find out the k nearest neighbors
Calculate the distance between examples
Whether contain minority example
yes
no
Increase the weight of minority examples
Keep the weights unchanged
Classify
Classification results

Fig.3. The flow chart of KNN algorithm based on the minority class

As we can see from Fig.3, the first step is to find out the k nearest neighbors of test example in the imbalanced datasets. Then we will calculate the distance between test example and the k neighbors. The main improvement is to judge whether there is minority example in the k nearest neighbors of the test example. If minority example exists in the neighbors, the local characteristic of the minority example will be analyzed to increase its weight. In the same way, we find out the distribution characteristics of the k nearest neighbors of the minority example, and calculate the number of the majority examples. The more numbers of the majority examples in its neighbors represent the more unsafe the minority example is. We will increase the weight of minority example according to the number of majority examples and keep the weight of majority examples the same. Finally, we classify the example according to the principle of the minority obeying the majority.

IV. Experiments and Analysis

A. Data sets

In this paper, we use 14 imbalanced binary datasets from Keel data set repository. The characters of these datasets are shown in Table I, including imbalance radio, total attributes, total examples, the number of majority class examples and the number of minority class examples. For more details about these datasets, you can refer to url: http:// sci2s.ugr.es/keel/imbalanced.php.

TABLE I. THE CHARACTER OF THE IMBALANCED DATA SETS

ID	Dataset Name	Attributes	Examples	Minority	Majority	IR
1	yeast3	9	1484	163	1321	8.10
2	ecoli3	8	336	35	301	8.60
3	yeast2vs4	9	514	51	463	9.08
4	yeast0359vs78	9	506	50	456	9.12
5	yeast0256vs3789	9	1004	99	905	9.14
6	yeast02579vs368	9	1004	99	905	9.14
7	ecoli0267vs35	8	224	22	202	9.18
8	yeast05679vs4	9	528	51	447	9.35
9	vowel0	14	988	90	898	9.98
10	ecoli0147vs56	7	332	25	307	12.28
11	ecoli0146vs5	7	280	20	260	13.00
12	yeast4	9	1484	51	1433	28.10
13	yeast5	9	1484	44	1440	32.73
14	yeast6	9	1484	35	1449	41.40

B. *Experimental setup*

The experiment was conducted using the 5-fold cross validation strategy and we set the value of k to 5. We implement the algorithm with Python. In this paper, evaluation parameters of classification models are defined based on confusion matrix shown in Table II.

TABLE II. THE CONFUSION MATRIX

	True class (p)	True class(n)
Hypothesis output(Y)	TP	FP
Hypothesis output(N)	FN	TN

Consider a basic two-class classification problem, let {p, n} be the true positive and negative class label and {Y, N} be the predicted positive and negative class labels, respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \tag{7}$$

$$prcision = \frac{TP}{TP+FP}, recall = \frac{TP}{TP+FN} \tag{8}$$

$$F-Measure = \frac{\left(1+\beta\right)^2 \cdot recall \cdot precision}{\beta^2 \cdot recall + precision} \tag{9}$$

$$G-mean = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \tag{10}$$

Where β is a coefficient to adjust the relative importance of precision versus recall and usually we set β to 1.Accuracy, precision and recall can provide a simple way for describing the performance of classifier, but they are sensitive to environmental changes. And other evaluation metrics are frequently adopted to provide comprehensive assessments of imbalanced data, namely F-Measure (F-Meas) and G-mean (G-mean). In the following experiment, we compare the improved algorithm with WDKNN based on 14 imbalanced data sets, and the result is presented in Table III.

TABLE III. THE COMPARISON BETWEEN THE TWO ALGORITHM

	WDKNN		Proposed algorithm	
DataName	F-Meas	G-mean	F-Meas	G-mean
yeast3	0.7362	0.8133	0.7477	0.8262
ecoli3	0.6065	0.7337	0.6638	0.8043
yeast2vs4	0.7280	0.7832	0.7374	0.8039
yeast0359vs78	0.3698	0.5151	0.4267	0.6243
yeast0256vs3789	0.6167	0.7149	0.6270	0.7349
yeast02579vs368	0.8265	0.8867	0.8269	0.8949
ecoli0267vs35	0.7099	0.7558	0.7318	0.7767
yeast05679vs4	0.5688	0.6875	0.6492	0.7707
vowel0	0.9828	0.9880	0.9946	0.9994
ecoli0147vs56	0.7822	0.8247	0.8076	0.8045
ecoli0146vs5	0.8159	0.8541	0.8686	0.9314
yeast4	0.2814	0.4345	0.3955	0.5671
yeast5	0.7499	0.8476	0.7546	0.8735
yeast6	0.5444	0.7038	0.5969	0.7807
average	0.6656	0.7531	0.7020	0.7995

From the Table III, we can see that the values of the F-Measure and G-mean are obviously improved for all the imbalanced data sets in our experiment. For the dataset yeast4, both algorithms do not perform well, but they have very good classification performance in vowel0, the performance of the improved algorithm has been greatly ameliorated.

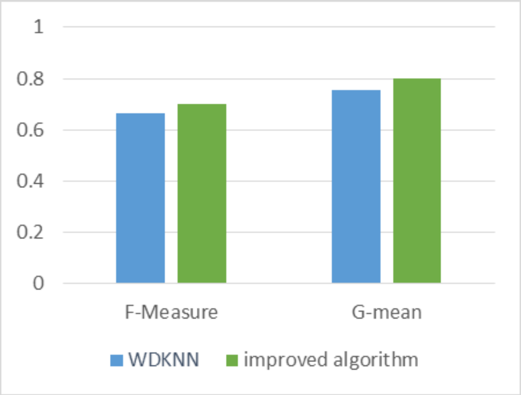

Fig.4. The average value of F-Measure and G-mean

From Fig.4, we know that the average value of F-Measure increases by 9% and G-mean increases by 9%. Experimental results show that the proposed algorithm improves both accuracy and efficiency for classification.

V. Conclusion

This paper provided an improved method based on the existing KNN classification algorithm. We pay more attention to the minority class than the majority class in the imbalanced data. The proposed method increases the weight of minority class based on local characteristic of the minority class distribution. In addition, we compare the parameter of F-Measure and G-mean of the proposed algorithm with the existing WDKNN. Experimental results show that our improved algorithm performs better than WDKNN in imbalanced data sets.

Acknowledgement

This work is partly supported by the State Key Development Program of Basic Research of China (2013CB329005), the National Natural Science Foundation of China (Grant No. 61322104, 61571240), the Priority Academic Program Development of Jiangsu Higher Education Institutions, the Natural Science Foundation of Jiangsu Province (Grant No. BK20161517), the Qing Lan Project.

References

[1] Q. Yang, X. Wu, "Ten challenge problems in data mining research, " *International Journal of Information Technology & Decision Making*, vol.05, no.4, pp.597-604, 2006.

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

[3] V. Athitsos, J. Alon, S. Sclaroff, "Efficient nearest neighbor classification using a cascade of approximate similarity measures," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 486-493, 2005.

[4] T. Cover, P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, January 1967.

[5] K. Weinberger, L. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Journal of Machine Learning Research*, vol. 10, no.1, pp. 207-244, 2006.

[6] R. Min, D. A. Stanley, Z. Yuan, et al, "A Deep Non-linear Feature Mapping for Large-Margin kNN Classification," *2009 Ninth IEEE International Conference on Data Mining*, Miami, FL, pp. 357-366, 2009.

[7] T. Yang, L. Cao, C. Zhang, "A Novel Prototype Reduction Method for the K-Nearest Neighbor Algorithm with $K\geq 1$," *Lecture Notes in Computer Science*, pp. 89-100, 2010.

[8] W. Liu, S. Chawla, "Class Confidence Weighted kNN Algorithms for Imbalanced Data Sets," *Advances in Knowledge Discovery and Data Mining*, pp. 345-356, 2011.

[9] V. López, A. Fernández, S. García, et al, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, " *Information Sciences*, vol. 250, no. 11, pp. 113-141, 2013.

[10] J. Stefanowski, "Overlapping, Rare Examples and Class Decomposition in Learning Classifiers from Imbalanced Data, " Emerging Paradigms in Machine Learning, Berlin Heidelberg, pp. 277-306, 2013.

[11] D. Anyfantis, M. Karagiannopoulos, S. Kotsiantis, et al, "Robustness of learning techniques in handling class noise in imbalanced datasets," Artificial Intelligence and Innovations 2007: from Theory to Applications, Springer US, pp. 21-28, 2007.

[12] J. Błaszczyński, J. Stefanowski, "Neighbourhood sampling in bagging for imbalanced data, " *Neurocomputing*, pp. 529-542, 2015.

[13] Y. Song, J. Huang, D. Zhou, et al, "IKNN: Informative K-Nearest Neighbor Pattern Classification, " Knowledge Discovery in Databases: Pkdd 2007, European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, pp. 248-264, September , 2007.