IOP Conf. Series: Materials Science and Engineering **719** (2020) 012072 doi:10.1088/1757-899X/719/1/012072

Improving k-Nearest Neighbors Algorithm for Imbalanced Data Classification

Zhan Shi

Power Dispatch and Control Center, Guangdong Power Grid Co., Ltd., China
Email: shiz@foxmail.com

Abstract. The k-Nearest Neighbors (k-NN) algorithm is a classic non-parametric method that has wide applications in data classification and prediction. Like many other machine learning schemes, the performance of k-NN classifiers will be significantly impacted by the imbalanced class distributions of data. That is, the data instances in the majority class tend to dominate the prediction of the test instances. In this paper, we look into the data pre-processing techniques that can be used to rebalance the training data and enhance the performance of k-NN classifiers in imbalanced data sets. We conduct extensive experiments on 14 real-world data sets collected from different application domains. We also perform statistical tests to verify the significance of different data pre-processing techniques in terms of boosting k-NN classification precision.

1. Introduction

In the big data era, data mining and machine learning techniques become a fundamental building block in the application systems across a variety of domains. Researchers and practitioners take advantage of the ever-growing and diverse data from different sources for knowledge extraction, in order to obtain meaningful predications.

The quality of data is a critical challenge in real-world data mining applications. In many domains, the nature of data is imbalanced. That is, there is a majority class that dominates the data set, and data from the minority class is rare. Usually, the minority class is of great concern in these applications. For example, for social network spam detection in cybersecurity domain, most of the messages are benign and only around 5% are spam [1]. Lots of traditional machine learning algorithms have difficulties to handle imbalanced data [2-4].

The k-Nearest Neighbors (k-NN) algorithm [5] is a classic non-parametric method that is widely used for classification. For training, it only needs to store the feature vectors and class labels of the training data. In the testing phase, a test instance is given the label that is most frequent among the k training instances nearest to itself. The two-class k-NN algorithm has a nice statistical property. That is, it is guaranteed to yield an error rate no worse than twice the Bayes error rate, as the data size approaches infinity. k-NN suffers from the problem of class imbalance. This is because during the majority voting in the classification phase, instances from the majority class tend to dominate the prediction of the test instance. One way to overcome this drawback is to give different weights to the training instances. The other way is to perform data resampling, including oversampling the minority class and undersampling the majority class.

In this paper, we investigate a variety of data resampling methods for improving the classification performance of k-NN in imbalanced data sets. Specifically, we consider nine sampling schemes, such as random sampling, synthetic minority class oversampling (SMOTE), Wilsons editing, cluster-based sampling, and ensemble data sampling. The goal is to identify which schemes are more suitable for k-NN in real-world applications. For the purpose of evaluation, we conduct extensive experiments based on 14 publicly available real-world data sets that are collected from different areas, including web-usage records and medical data. The precision results of the minority class are presented, along with a hypothesis testing using one-factor analysis of variance (ANOVA) models.

The remainder of this paper is organized as follows. Section 2 presents a brief literature review of related work. Section 3 discusses the methodology of this work. Section 4 introduces the experimental design and results. Section 5 concludes this work.

2. Related Work

Class imbalance occurs in a classification problem when the classes exhibit a skewed distribution. It is ubiquitous and prevalent in many real-world applications. Generally, there are two types of methods to solve the problem. The first type is data pre-processing, which attempts to re-balance the training data by generating artificial data instances and using data sampling techniques. The second type is to adjust the weight of instances or the cost of objective functions in the classification models, which is usually model specific. In this work, we focus on the data pre-processing approaches, which are more general and applicable to different classification schemes.

In order to re-balance the training data, one can choose to remove some data in the majority class or generate some data (e.g., by duplicating instances and generating artificial instances) to compliment the minority class. For example, random oversampling selects instances in the minority class randomly and then duplicates them, and random undersampling selects data instances from the majority class in random and then removes them [6]. The drawbacks of random oversampling and undersampling are the potential overfitting of classifiers and the possibility of losing useful information. Chawla et al. [7] propose the Synthetic Minority Over-sampling Technique (SMOTE), which is extended by Han et al. [8] to the Borderline-SMOTE algorithm. Barandela et al. [9] introduce the Wilsons editing algorithm that downsizes the majority class using k-NN classifiers. The cluster-based oversampling [10] and cluster-based undersampling [11] algorithms use clustering techniques to generate balanced training sets. Kubat et al. [12] propose the one-sided selection method that creates a training set consisting of safe data by removing the instances that are considered either redundant or noisy in the majority class. An ensemble method is proposed by Wang et al. [13], which combines the outputs of various sampling methods by using the majority voting scheme.

A series of researches have investigated the approaches to perform imbalanced data mining in big data environments. Fernández et al. [2] provided an insight into the existing approaches and challenges of imbalanced classification in big data applications. The key questions towards solving the problem are how to generate artificial data instances and how to balance the trade-offs for the ratio between classes. Abdel-Hamid et al. [14] propose an imbalanced data mining framework based on Spark that consists of two key modules for border handling and selective border instance sampling. Triguero et al. [15] propose a parallel model of evolutionary undersampling for imbalanced big data. Rastogi et al. [16] take advantage of locality sensitivity hashing to implement a distributed version of SMOTE based on Spark. In [17], Jedrzejowicz et al. introduce a novel approach to parallelize computations in imbalanced data classification using MapReduce.

3. Methodology

3.1. k-Nearest Neighbors Algorithm

Machine learning algorithms can be generally divided into two categories, that is, parametric and non-parametric. The k-Nearest Neighbors (k-NN) algorithm [5] is the most popular representative of the non-parametric machine learning algorithms.

The k-NN algorithm basically requires no training, just simply stores the feature vectors and class labels of the training data. Given a new instance for testing, its class is predicted as majority class label from its k nearest neighbors.

In particular, we use Euclidean distance to define the distance between instances. Besides, we set $k = 1$ in the experiment. This is the most intuitive nearest neighbor classifier, which simply assigns an instance to the class of its closest neighbor in the feature space. As the size of training set approaches infinity, the 1-NN classifier also guarantees an error rate of no worse than twice the Bayes error rate.

3.2. Data Pre-processing Techniques

In this work, we focus on improving the imbalanced data classification performance of k-NN by using data pre-processing techniques. In particular, we explore a variety of data resampling methods, which are introduced as follows.

Oversampling techniques are used to increase the number of data instances in the minority class, by either duplicating some of the minority data instances or generating some artificial data instances. We adopt five oversampling techniques in this study. In specific, we use the classic random oversampling, SMOTE [7] and its variant Borderline-SMOTE [8], as well as the cluster-based oversampling [10]. In addition, we adopt the ensemble oversampling algorithm [13] based on information decomposition, cluster-based oversampling and random oversampling.

Undersampling techniques are used to decrease the number of data instances in the majority class. Specifically, we implement four undersampling techniques in this study. These algorithms are random undersampling, Wilsons editing undersampling [9], cluster-based undersampling [11] and one-sided selection undersampling [12].

The data pre-processing techniques are used to re-balance the classes in the training data, in order to provide the k-NN model with a relatively balanced training set. The testing data set are left intact in our study. In other words, the testing set is imbalanced as in the state they are in the real world. This ensures that our results reflect the actual performance in real-world applications.

3.3. Evaluation Methods

In the experiments, the minority class is treated as the positive class and the majority class is treated as the negative class. For imbalanced data sets, the overall accuracy is not a good indicator of the actual classification performance, especially for the positive (minority) class. Therefore, we use the precision metric in our study, which is widely used in data mining and statistical test. In particular, we count the number of true positives, false positives, true negatives and false negatives. Precision is the proportion of positive results in classification that are true positive, as showed in the following equation.

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{1}$$

To provide some insight into the experimental results, we conduct a statistical test using one-factor analysis of variance (ANOVA) models [18] for different data sampling methods. We consider the null hypothesis is that there is no significant difference among different sampling techniques, while against it the alternative hypothesis is that at least one is significantly different. Additionally, we perform the Tukey’s HSD (honestly significant difference) test, which indicates the performance levels of different sampling techniques.

4. Experiment and Results

4.1. Experimental Design

Table 1 shows the data sets used in this work. In specific, we make use of 14 public data sets that are collected from different areas [19-21]. As can been seen in the table, the size of data sets (i.e., the number of instances) varies a lot, ranging from the smallest of 253 (i.e., mw1) to the largest of 17186 (i.e., pc5). The number of attributes is between 9 and 74. In addition, Table 1 presents the size of minority class and the imbalance rate in each data set. The imbalance rate is defined as the number of instances in the majority class divided by the number of instances in the minority class. The most imbalanced data set is mv1 that has the highest imbalance rate of 4556%, while the diabetic data set with 113% imbalance rate is roughly balanced. In short, the data sets we use in the experiments cover a variety of sizes and imbalance rates.

Table 1. Data Sets.

Data Sets	Size	Size of Minority Class	Imbalance Rate	Number of Attributes
diabetic	1151	540	113%	20
pima	768	268	187%	9
ilpd	583	167	249%	11
etoys	270	28	864%	41
onehr	1832	57	3114%	74
bmg	2417	547	342%	41
qvc	2152	386	458%	41
mw1	253	27	837%	38
mc1	1988	46	4222%	39
pc1	705	61	1056%	38
pc2	745	16	4556%	37
pc3	1077	134	704%	38
pc4	1458	178	719%	38
pc5	17186	516	3231%	39

For the purpose of evaluation, we split each of the data sets into two parts in random. The first part consists of 60% of instances, which are used as training data. The other part consists the rest instances that are used to test the classification performance.

In order to find the best parameters in the data sampling step, we apply each sampling algorithm to the training data with a range of parameters. For undersampling on the majority class, we use sampling rates of 20%, 50%, 70%, 90% and EVEN (i.e., the number of instances in both classes are even). For oversampling on the minority class, we test sampling rates of 200%, 500%, 700%, 900%, and EVEN. We also build the baseline classifier from the original imbalanced data set.

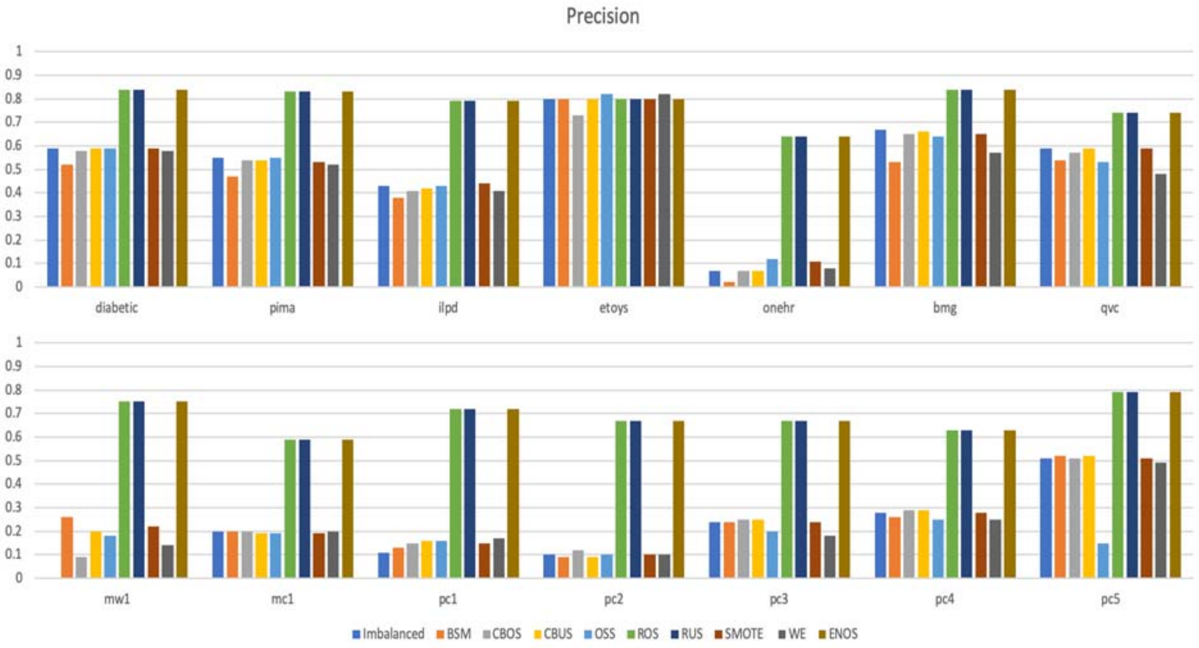

Figure 1. Precision Results.

4.2. Results

The precision results are illustrated in Figure 1. First of all, among the data sampling schemes, random oversampling (ROS), random undersampling (RUS) and ensemble oversampling (ENOS) are able to improve the precision of k-NN classifiers in most data sets. In mw1, mc1, pc1, pc2, and pc3 data sets, the improvements of precision are over 40%, while in diabetic, pima, ilpd, onehr, bmg, qvc, pc4, pc5 data sets, the improvements are between 15% and 39%. The only exception is the etoys data set, where there is no obvious improvement and the precision level is around 80% for all algorithms.

Table 2. ANOVA Models.

Data Sets	Degrees of Freedom	Sum of Squares	Mean Squares	F-statistics	P-value
diabetic	9	40.28	4.48	18.51	<<0.05
pima	9	20.44	2.27	9.81	<<0.05
ilpd	9	5	0.56	2.52	<0.05
etoys	9	7.23	0.8	7.61	<<0.05
onehr	9	67.69	7.52	162.07	<<0.05
bmg	9	4.31	0.48	2.43	<0.05
qvc	9	7.25	0.81	4.26	<<0.05
mw1	9	3.87	0.43	5.79	<<0.05
mc1	9	5.81	0.65	14.79	<<0.05
pc1	9	2.47	0.27	2.68	<0.05
pc2	9	0.35	0.04	0.98	<0.05
pc3	9	5.35	0.59	5.11	<<0.05
pc4	9	3.59	0.4	3.31	<<0.05
pc5	9	64.79	7.2	190.78	<<0.05

Table 2 and Table 3 depict ANOVA models and the HSD test results of the classification precision for various methods across data sets. We can see that the selection of data sampling techniques has a significant impact on prediction label at the 5% level (i.e., $\mathrm{P-value} < 0.05$). In other words, at least one sampling technique is statistical significantly different from others. The HSD test shows that ROS, RUS and ENOS have outstanding performance (HSD marked as "A").

Table 3. HSD Test Results.

Methods	Mean	HSD
Imbalanced	0.37	B
Borderline-SMOTE (BSM)	0.35	B
Cluster-based Oversampling (CBOS)	0.37	B
Cluster-based Undersampling (CBUS)	0.38	B
One-sided Selection (OSS)	0.35	B
Random Oversampling (ROS)	0.74	A
Random Undersampling (RUS)	0.74	A
SMOTE (SM)	0.39	B
Wilsons Editing (WE)	0.36	B
Ensemble Oversampling (ENOS)	0.74	A

5. Conclusion

The class imbalance problem is a great challenge for most machine learning algorithms. In this work, we focus on k-NN, a popular traditional algorithm, and explore the impact of imbalanced data set. To address this problem, we use a variety of data sampling algorithms to rebalance the training set for k-NN. Experiment results obtained based on 14 real-world data sets from different areas indicate that the precision performance of k-NN classifiers in imbalanced data could be greatly improved by applying random oversampling, random undersampling and ensemble oversampling to the training data.

6. Acknowledgments

This work is supported by the Guangdong Power Grid Project (GDKJXM20162186).

7. References

[1] Liu S, Wang Y, Zhang J, Chen C and Xiang Y. Addressing the Class Imbalance Problem in Twitter Spam Detection Using Ensemble Learning. Computers & Security, 69:35-49, 2017.
[2] Fernández A, del Río S, Chawla NV and Herrera F. An insight into imbalanced Big Data classification: outcomes and challenges. Complex Intell. Syst. (2017) 3: 105.
[3] Chen C, Wang Y, Zhang J, Xiang Y, Zhou W, and Min G. Statistical Features Based Real-time Detection of Drifted Twitter Spam. IEEE TIFS, vol.12, no. 4, pp. 914-925, Apr 2017.
[4] Wang Y, Zhang J, Xiang Y, and Zhou W. Internet Traffic Clustering with Side Information. Journal of Computer and System Sciences, vol. 80, no. 5, pp. 1021-1036, Aug 2014.
[5] Fix E and Hodges JL. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. International Statistical Review, (1989)57(3), 238-247.
[6] Sui Y, Zhang X, Huan J, and Hong H. Exploring data sampling techniques for imbalanced classification problems. Proc. SPIE 11198, Fourth International Workshop on Pattern Recognition, 1119813 (31 July 2019)
[7] Chawla NV, Bowyer KW, Hall LO and Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, vol. 16, pp. 321-357, 2002.
[8] [8] Han H, Wang WY, and Mao BH. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Advances in intelligent computing, 2005, pp. 878-887.
[9] Barandela R, Valdovinos RM, Sánchez JS, and Ferri FJ. The imbalanced training sample problem: Under or over sampling? in SSPR /SPR 2004, pp. 806-814.
[10] Jo T and Japkowicz N. Class imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter, vol. 6, pp. 40-49, 2004.
[11] Rahman MM and DavisD. Cluster based under-sampling for unbalanced cardiovascular data. in Proceedings of the World Congress on Engineering, 2013.
[12] Kubat M and Matwin S. Addressing the curse of imbalanced training sets: one-sided selection. ICML, 1997, pp. 179-186.
[13] Wang C, Hu L, Guo M, Liu X, and Zou Q. imDC: an ensemble learning method for imbalanced classification with miRNA data. Genetics and molecular research, vol. 14, pp.123, 2015.
[14] Abdel-Hamid NB, ElGhamrawy S, Desouky AE, Arafat H. A Dynamic Spark-based Classification Framework for Imbalanced Big Data. J Grid Computing (2018) 16: 607.
[15] Triguero I, Galar M, Vluymans S, Cornelis C, Bustince H, Herrera F, Saeys Y. Evolutionary undersampling for imbalanced big data classification. in CEC 2015, Sendai, pp.715-722.
[16] Rastogi AK, Narang N, and Siddiqui ZA. Imbalanced big data classification: a distributed implementation of SMOTE. In Proceedings of the ICDCN '18 Workshops. ACM, NY, USA.
[17] Jedrzejowicz J, Kostrzewski R, Neumann J and Zakrzewska M. (2018) Imbalanced data classification using MapReduce and relief. J. of Info. and Tele., 2:2, 217-230.
[18] Berenson ML, Levine DM, and Goldstein M. Intermediate statiscal methods and applications: a computer package approach. 1983.
[19] Asuncion A and Newman D. UCI machine learning repository.
[20] Saar-Tsechansky M and Provost F. Handling missing values when applying classification models. 2007.
[21] Shirabad JS and Menzies TJ. The PROMISE repository of software engineering databases.