Learning from Imbalanced Data: A Comparative Study

Yu Sui(✉), Mengze Yu, Haifeng Hong, and Xianxian Pan

Program and Research Center, Guangdong Power Grid Corporation, Guangzhou, China
suiyugxzx@163.com

Abstract. Learning from imbalanced data is a great challenge when we use machine learning techniques to solve real-world problems. Imbalanced data can result in a classifier's sub-optimal performance. Moreover, the distribution of the testing data may differ from that of the training data, thus the true mis-classification costs is hard to predict at the time of learning. In this paper, we present a comparative study on various sampling techniques in terms of their effectiveness in improving machine learning performance against class imbalanced data sets. In particular, we evaluate ten sampling techniques such as random sampling, cluster-based sampling, and SMOTE. Two widely used machine learning algorithms are applied to train the base classifiers. For the purpose of evaluation, a number of data sets from different domains are used and the results are analysed based on different metrics.

Keywords: Data mining · Machine learning · Class imbalance · Data sampling

1 Introduction

The class imbalance problem is a challenge to data mining and machine learning, and it is of crucial importance since last decade in many domains, such as network intrusion detection, financial engineering, medical diagnostics, surveillance, and even in-flight helicopter gearbox fault monitoring [13]. In certain cases, this has caused a significant bottleneck in the performance attainable by traditional data mining algorithms, which tend to bias to the majority class. That is, the accuracy for majority class is high while the performance is poor for minority class. For example, if a data sample contains of 95% of majority class and 5% of minority class, thus an accuracy rate of 95% (which is in general a good accuracy) can be achieved by classifying all examples to majority class. However, such a model has no practical value in real-world problems. It is typically the minority class in which the practitioners are more interested. Therefore, a natural question is how to empower the classification performance when the data set is imbalanced?

Supported by Guangdong Power Grid Research Project 030000QQ00180019.

W. Meng and S. Furnell (Eds.): SocialSec 2019, CCIS 1095, pp. 264–274, 2019.
https://doi.org/10.1007/978-981-15-0758-8_20

Class imbalance problem has been studied by many researchers [9,11,16]. So far, a common solution is using data sampling techniques to re-distribute the data across classes. Generally speaking, a pre-sampling method balances the training set, either by oversampling the minority class or undersampling the majority class. A large amount of data sampling techniques have been proposed in the past and some of them have been applied to address the class imbalance problem. The simplest over-sampling method is to randomly duplicate some of the minority instances (ROS), while a more complex version is the synthetic minority class over sampling (SMOTE) [5] technique which artificially creates new minority examples from known examples. Han et al. [8] proposed a Borderline-SMOTE over sampling approach, which improves upon SMOTE by only oversampling minority class samples which are believed to be on the border of the decision regions. Cluster-based oversampling (CBOS) [14] attempts to even out the between-class imbalance as well as the within-class imbalance. Meanwhile, the simplest under-sampling method is random undersampling (RUS), which randomly reduces data of the majority class. One-sided selection (OSS) [15], which removes the majority class samples that are considered either redundant or noisy, is one of the earliest attempts to improve upon the performance of random resampling. In addition, Wilson's editing (WE) [1] uses the kNN (Nearest Neighbor) technique with $k = 3$ to classify each sample in the training set by using the remaining class, and remove those majority class which are misclassified. Various data sampling techniques have been explored. However, there is no universal solution and it is worth to explore which kind of data sampling technique is more effective and efficient in balancing class distribution in terms of the type of data and classifiers.

This paper presents an extensive experimental study on a variety of data sampling techniques, with a focus on their effectiveness in terms of boosting the classification performance of machine learning algorithms on class imbalanced data sets. In particular, we use two popular algorithms, i.e., C4.5 Decision Tree and Support Vector Machines, to train the base classifiers. The study is based on a number of different imbalanced data sets from the PROMISE repository software engineering databases [20]. To our knowledge, this is the first comprehensive empirical investigation in comparing the performance of these data sampling techniques among imbalanced data sets from different application domains.

The rest of the paper is organised as follows. Section 2 introduces the related work, while the details of data sets and methodology are presented in Sect. 3. Section 4 discusses the experimental results on different performance measures. Conclusions and future work are provided in Sect. 5.

2 Related Work

Data-Perspective Approaches. Generally speaking, approaches to classification with imbalanced data issues involve two main categories, i.e., data perspective and algorithm perspective. In this work, we mainly focus on sampling techniques that is related to our study.

Random Over-Sampling (ROS): The minority oversampling randomly selecting a training example from the minority class, and then duplicating it. This may usually cause over-fitting and longer training time during imbalance learning process.

Random Under-Sampling (RUS): Majority under-sampling draws a random subset from the majority class while discarding the rest instances. In doing so, the class distribution can be balanced, however, some important information may be lost when examples are removed from the training data set at random, and especially when the data set is small.

Synthetic Minority Over-Sampling Techniques (SMOTE or SM) [5]: This technique adds new artificial minority attribute examples by extrapolating instances from the k nearest neighbours (kNN) to the minority class instances. In our experiments, the parameter k is set to five.

Border-SMOTE (BSM) [8]: BSM is an attempt to improve upon SMOTE by only oversampling minority class instances which are considered to be on the border of the minority-decision region. It can be described as follows: First, determine kNN for each original sample $x_i \in S_{min}$ and identify the number of nearest neighbours that belong to the majority class, then if $\frac{k}{2} < t < k$ is true, $x_i \in S_{min}$ is considered as borderline instance, finally, SM is applied to create new examples by using borderline samples.

Wilson's Editing (WE) [1]: WE applies the kNN classifier with $k = 3$ to classify each example in the training set by using all the remaining examples, and removes those majority class instances whose class label does not agree with the class associated with the largest number of the k neighbours.

Cluster-Based Oversampling (CBOS) [14]: Before performing random oversampling, CBOS first uses k-means algorithm to cluster the minority and majority classes separately. All clusters in the majority class, except for the largest one, are randomly oversampling as the same number of the training examples as the largest cluster. Then the total number of the majority clusters are even out to each cluster of minority clusters.

Cluster-Based Undersampling (CBUS) [18]: CBUS is not to balance the data ratio of majority class of minority class into 1:1, instead to reduce the gap between the numbers of minority class and minority class. Different from CBOS, this method only clusters the majority class into K clusters and regard each cluster as one subset of the majority class samples. After that CBUS combines each cluster with the whole minority class, and then thecombined data sets will considered as the updated training data sets. Finally, CBUS classifies all the K data sets with a learning algorithm and chose the data set with the highest accuracy for building the training model.

One-Sided Selection (OSS) [15]: Similar to the idea of BSM, OSS aims to create a training set consisting of safe cases by removing the considered either redundant or noisy examples of the majority class examples. When using OSS, Borderline and noisy cases are detected by Tomek links.

Ensemble Oversampling Algorithm (ENOS) [22]: ENOS integrates the information decomposition algorithm, cluster-based oversampling and random oversampling approaches. In specific, first the algorithm assumes that there are missing instances which caused the data set to be imbalanced and the missing instances are recovered by using information decomposition algorithm. After that, the classification models from random oversampling and cluster-based oversampling techniques are combined together, where majority voting is used to obtain the final result.

Algorithm-Perspective Approaches. The goal of algorithm level learning is to optimize the performance of the learning model on unseen data. Various algorithms have been proposed in last decades. For example, cost-sensitive learning is regard as an important approach for the class imbalance problem. Many cost-sensitive methods have been proposed, for instance, cost-sensitive boosting [21], meta cost [7], adjusting misclassification costs algorithm [3], Genetic Programming (GP) [10], and kernel-based one-class classifier via optimizing its parameters [24]. Yang et al. [23] explore the use of cost-based soft-margin maximization method, which is used to penalize certain misclassified examples and treats the positive and negative example differently. Besides, one-class learning methods such as one-class SVMs [19] and neural networks [12] were proposed to combat the over-fitting problem.

3 Methodology

3.1 Data Sets

In this work, we use data sets from the PROMISE repository software engineering databases [20], which are listed in Table 1. Detailed information includes data set name, data set size, the amount of minority class data, and the percentage of minority class data, and the class attribute. As can be seen in the table, the data sets we use in this study cover a variety of sizes and imbalance levels. More specifically, the percentage of minority class data varies from 2.2% (highly imbalanced which can be regarded as imbalance due to rare minority instances) to 12.4%. Besides, the size of data varies from the smallest data set with 253 data points to the largest data set with 17186 data points. Moreover, these data sets represent different application domains.

3.2 Machine Learning Algorithms

In this paper, we employ two classic machine learning algorithms to build classifier models using unbalanced training data and evaluate their performance by

Table 1. Data sets

Dataset	Size	#min	%min	#attr
MW1	253	27	10.7	38
MC1	1988	46	2.3	39
pc1	705	61	8.7	38
pc2	745	16	2.2	37
pc3	1077	134	12.4	38
pc4	1458	178	12.2	38
pc5	17186	516	3.0	39

unseen test data. These two algorithms are popular and widely used in the real world. For the purpose of evaluation, we use the implementation of these algorithms provided in Matlab 8.0.

C4.5 Decision Tree [17]. C4.5 improves upon ID3 by adding support for handling missing values and tree pruning. It builds decision trees using an entropy-based splitting criterion, which is sensitive to class imbalance in the training data. This is because C4.5 works globally while not paying much attention to specific data points.

Support Vector Machine (SVM) [6]. SVM is a classifier that for binary classification, which attempt to find out a linear combination of the variables that best divide the samples into two groups. The ideal separation is that the optimal linear combination of variables can maximize the distance between the classes. However, when the perfect separation is not possible, the optimal linear combination will be determined by a criterion in order to minimize the number of mis-classifications.

3.3 Performance Measure

In this work, we consider the minority class as the positive class and the majority class as the negative class. Overall classification accuracy is not a good metric for measuring the performance of classifiers in the face of imbalanced data. Thus the evaluation of the classification models should be done by other criteria rather than overall accuracy. In this work, we carry out the comparative study using four performance metrics, which are based on the confusion matrix metrics including true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The definition of performance metrics are described as follows.

Precision is the positive predictive value that measures the proportion of positive results in classification that are true positive. The metric is widely used in the evaluation of machine learning results. In particular, we focus on the precision of the minority class in this work, which can be obtained through the following equation.

$$Precision = \frac{TP}{TP + FP}$$

G-Measure (GM) is the geometric mean of the classification accuracy between classes, and each class of poor accuracy will cause low GM value, which in turn indicates that at least one class cannot be identified effectively. G-Measure can be calculated using the following formula.

$$GM = \sqrt{Recall \cdot Precision}$$

Cohen's Kappa rate, which evaluates the merit of the classifier, is an alternative measure to accuracy because it compensates for random hits. Previous studies [4] show that Kappa rate penalizes all-positive or all-negative predictions. The value of Cohen's Kappa ranges from -1 (total disagreement) to 0 (random classification) to 1 (total agreement). It can be derived as follows.

$$Kappa = \frac{N \sum_{i=1}^{k} x_{ii} - \sum_{i=1}^{k} x_i x_i}{N^2 - \sum_{i=1}^{k} x_i x_i}$$

Matthew's Correlation Coefficient is a single performance measure that considers both error rates and mutual accuracy on both the minority class and majority class in terms of confusion matrix. It will be less influenced by imbalanced data sets. The value of MCC ranges from 1 (perfect prediction) to -1 (the worst prediction), while 0 indicates that the model produces random results. According to an earlier study [2], MCC is regarded as a good singular measure for imbalance learning problem.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

3.4 Experimental Design

In this comparative study, we randomly choose 60% of the original data points as training data and the other 40% are used as unseen testing data. We note that all the data sampling techniques are only used to process the training data, while the testing data are left alone. In this way, the testing data can better reflect the real class distribution in real world problems. Therefore, the approach is more practical and applicable.

Regarding the data sampling parameters, undersampling techniques are performed at 20%, 50%, 70%, and 90% of the majority class, while oversampling are performed at 200%, 500%, 700%, 900% of the minority class. For example, running RUS_20 means 20% of majority class will be removed after applying RUS method to the training data set. Similarly, ROS 200 means the data size of minority class will be raised to 200% of the original. Moreover, BSM_even, CBOS_even, CBUS_even, OSS_even, ROS_even, RUS_even, SM_even, WE_even,

and Ensemble_even are also considered in the evaluation. It means that the data size of the minority class and majority class is even. For example, ROS_even means that the minority class and the majority class of each data set has the same number of instances after applying ROS method with minority samples. In addition, original data set are also divided into 60% for training the model and the other 40% for testing in order to provide baseline for our experiments. Hence, the classification results obtained from the original data without data sampling are denoted as "None" in the results.

Table 2. Precision results across data sets

Classifier	Data set	Approach									
		None	BSM	CBOS	CBUS	OSS	ROS	RUS	SM	WE	ENOS
C4.5	MW1	0.44	0.35	0.40	0.38	0.50	0.67	0.71	**0.83**	0.43	**0.83**
	MC1	0.38	0.30	0.43	0.43	0.28	0.71	0.56	0.63	0.38	**0.94**
	pc1	0.44	0.24	0.43	0.44	0.43	0.74	0.61	0.72	0.33	**0.90**
	pc2	0.13	0.14	0.25	0.33	0.17	0.86	0.57	0.57	0.13	**1.00**
	pc3	0.30	0.26	0.26	0.31	0.33	0.70	0.61	0.64	0.30	**0.77**
	pc4	0.58	0.38	0.58	0.54	0.54	0.77	0.74	0.77	0.47	**0.83**
	pc5	0.49	0.53	0.55	0.51	0.51	0.76	0.70	0.72	0.52	**0.78**
SVM	MW1	0.40	0.32	0.43	0.40	0.43	0.54	0.62	0.43	0.38	**0.58**
	MC1	0.08	0.09	0.13	0.50	0.08	**0.17**	0.12	0.09	0.08	**0.17**
	pc1	0.21	0.22	0.43	0.56	0.20	0.38	0.27	0.23	0.21	**0.65**
	pc2	0.07	0.06	0.13	0.25	0.11	**0.40**	0.17	0.17	0.07	**0.40**
	pc3	0.24	0.24	0.35	0.29	0.25	0.37	0.25	0.24	0.24	**0.40**
	pc4	0.33	0.34	0.29	0.30	0.32	**0.74**	0.33	0.39	0.33	**0.66**
	pc5	0.29	0.31	0.57	0.56	0.30	0.30	0.29	0.34	0.31	**0.58**

4 Results

The best Precision, GM, Kappa, and MCC results obtained by each sampling technique are presented in Tables 2, 3, 4 and 5 respectively. We can see that in most situations, the data sampling techniques can improve the performance of the machine learning classifiers. In particular, the ENOS algorithm benefits both the C4.5 and SVM classifiers most by increasing all the metrics in most data sets.

It is worth to notice that not all of the sampling techniques result in better results than the model built directly from the original data set (i.e., the None column in the tables) on Precision measure. Take Precision for C4.5 classifier from Table 2 as an example, we can see that the methods such as BSM, CBOS, CBUS, OSS, and WE perform worse than None. However, we are more interested in the models that can correctly identify more minority class samples that may

Table 3. GM results across data sets

Classifier	Data set	Approach									
		None	BSM	CBOS	CBUS	OSS	ROS	RUS	SM	WE	ENOS
C4.5	MW1	0.4	0.44	0.32	0.42	0.46	**0.78**	0.57	0.64	0.34	**0.78**
	MC1	0.24	0.22	0.26	0.47	0.34	0.75	0.52	0.62	0.24	**0.86**
	pc1	0.39	0.3	0.44	0.54	0.43	0.8	0.61	0.68	0.36	**0.84**
	pc2	0.14	0.15	0.2	0.33	0.33	0.76	0.5	0.5	0.14	**0.8**
	pc3	0.36	0.32	0.3	0.41	0.39	0.78	0.69	0.71	0.37	**0.8**
	pc4	0.53	0.5	0.57	0.54	0.58	0.79	0.71	0.76	0.5	**0.81**
	pc5	0.49	0.51	0.53	0.53	0.57	0.75	0.68	0.7	0.54	**0.76**
SVM	MW1	0.47	0.42	0.48	0.47	0.48	0.62	0.67	0.48	0.45	**0.68**
	MC1	0.20	0.20	0.22	0.22	0.21	0.35	0.30	0.21	0.20	**0.36**
	pc1	0.40	0.39	0.46	0.47	0.40	**0.57**	0.48	0.44	0.40	**0.57**
	pc2	0.18	0.12	0.24	0.20	0.24	0.39	0.34	0.31	0.18	**0.40**
	pc3	0.42	0.43	0.41	0.41	0.44	0.46	**0.46**	0.42	0.42	**0.46**
	pc4	0.54	0.56	0.60	0.66	0.53	0.63	0.55	0.58	0.54	**0.67**
	pc5	0.52	0.54	0.54	0.54	0.53	0.52	0.52	0.56	0.54	**0.57**

Table 4. Kappa results across data sets

Classifier	Data set	Approach									
		None	BSM	CBOS	CBUS	OSS	ROS	RUS	SM	WE	ENOS
C4.5	MW1	0.34	0.34	0.25	0.34	0.36	0.74	0.51	0.59	0.27	**0.75**
	MC1	0.21	0.19	0.22	0.45	0.31	0.74	0.50	0.61	0.21	**0.85**
	pc1	0.32	0.22	0.38	0.48	0.38	0.78	0.57	0.65	0.30	**0.82**
	pc2	0.12	0.14	0.19	0.31	0.18	0.74	0.49	0.49	0.12	**0.77**
	pc3	0.28	0.23	0.21	0.31	0.32	0.74	0.65	0.67	0.28	**0.78**
	pc4	0.46	0.40	0.52	0.47	0.47	0.77	0.68	0.72	0.43	**0.79**
	pc5	0.47	0.50	0.51	0.51	0.55	0.74	0.67	0.69	0.52	**0.75**
SVM	MW1	0.39	0.31	0.41	0.39	0.41	**0.56**	0.62	0.41	0.36	**0.56**
	MC1	0.10	0.11	0.16	0.14	0.10	**0.25**	0.17	0.12	0.10	**0.25**
	pc1	0.23	0.24	0.37	0.42	0.22	0.47	0.32	0.27	0.23	**0.52**
	pc2	0.08	0.06	0.16	0.14	0.14	0.31	0.24	0.23	0.08	**0.32**
	pc3	0.26	0.27	0.29	0.29	0.28	**0.32**	0.27	0.26	0.25	**0.32**
	pc4	0.37	0.39	0.48	0.59	0.35	0.57	0.38	0.44	0.38	**0.60**
	pc5	0.42	0.44	0.45	0.51	0.42	0.42	0.42	0.47	0.44	**0.53**

have a lower overall accuracy. To be more specific, Table 3 shows that CBOS, CBUS, OSS, ROS, RUS, SM, and ENOS perform better than None for most times. This is because GM measures the classification accuracy from both posi-

Table 5. MCC results across data sets

Classifier	Data set	Approach									
		None	BSM	CBOS	CBUS	OSS	ROS	RUS	SM	WE	ENOS
C4.5	MW1	0.34	0.35	0.25	0.34	0.37	0.75	0.53	0.60	0.28	**0.76**
	MC1	0.23	0.20	0.25	0.45	0.32	0.74	0.51	0.61	0.23	**0.86**
	pc1	0.33	0.22	0.38	0.49	0.38	0.78	0.57	0.65	0.30	**0.83**
	pc2	0.12	0.14	0.19	0.31	0.29	0.75	0.49	0.49	0.12	**0.74**
	pc3	0.29	0.23	0.21	0.32	0.32	0.75	0.65	0.68	0.28	**0.78**
	pc4	0.46	0.42	0.52	0.47	0.50	0.77	0.68	0.72	0.43	**0.79**
	pc5	0.47	0.50	0.51	0.52	0.56	0.74	0.67	0.69	0.52	**0.75**
SVM	MW1	0.39	0.32	0.41	0.39	0.41	0.57	0.62	0.41	0.37	**0.63**
	MC1	0.15	0.16	0.18	0.18	0.16	**0.33**	0.27	0.17	0.15	**0.33**
	pc1	0.31	0.30	0.39	0.43	0.30	**0.52**	0.41	0.35	0.30	**0.52**
	pc2	0.12	0.07	0.20	0.15	0.18	0.36	0.30	0.27	0.12	**0.37**
	pc3	0.32	0.33	0.30	0.31	0.35	**0.37**	0.36	0.32	0.31	**0.37**
	pc4	0.45	0.46	0.52	0.61	0.43	0.57	0.46	0.50	0.45	**0.61**
	pc5	0.50	0.52	0.52	0.51	0.50	0.50	0.50	0.54	0.52	**0.55**

tive and negative perspectives, and low accuracy on either class will lead to low GM value. While Table 4 demonstrates that almost all the sampling techniques (except BSM and WE) can achieve higher Kappa value than None. Because the Kappa rate penalises all-positive or all-negative predictions, and its value 1 means total agreement. Obviously, we can see that sampling methods such as CBOS, ROS, RUS and ENOS can improve the imbalanced learning performance in terms of Kappa performance measure while the rest are close with each other, which are not much better than None. Finally, as we discussed before, MCC is less influenced by imbalanced unseen test sets. Table 5 shows that ROS, RUS and ENOS can result in higher MCC values in terms of average results.

Tables 2, 3 and 4 also show that when the data set is slightly imbalanced, most of the sampling techniques do not perform much improvement in imbalanced learning. However, when the datasets becomes more imbalanced (e.g., pc2, pc5, and MC1), almost all the sampling techniques perform better, which means such sampling techniques can improve the performance of imbalanced learning. Practically speaking, ROS, RUS and ENOS are the top three sampling techniques when facing different kinds of imbalance ratio on different performance measures.

In summary, we find that the ENOS algorithm boosts the performance of C4.5 classifier in all metrics including Precision, GM, Kappa, and MCC in most data sets, while ROS achieves the second best results. The case for the SVM classifier is similar, ENOS performs the best for most of the times. Next comes the ROS algorithm.

5 Conclusions

In this paper, we present a comparative study for machine learning with imbalanced data. A variety of imbalanced data sets are used in the evaluation. The main goal is to examine the performance of various data sampling approaches, in terms of the boosting of the classification performance of C4.5 Decision Tree and Support Vector Machines on class imbalance data, so as to provide practical guidance to machine learning practitioners when facing imbalanced learning problem. Based on our extensive experiments, we find that data sampling techniques can improve the performance of machine learning when data sets are severely imbalanced. Besides, we find that the ensemble oversampling algorithm and random oversampling achieve the top performance in most data sets and both classifiers.

References

1. Barandela, R., Valdovinos, R.M., Sánchez, J.S., Ferri, F.J.: The imbalanced training sample problem: under or over sampling? In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR /SPR 2004. LNCS, vol. 3138, pp. 806–814. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27868-9_88
2. Bekkar, M., Djemaa, H.K., Alitouche, T.A.: Evaluation measures for models assessment over imbalanced data sets. J. Inf. Eng. Appl. **3**(10) (2013)
3. Bhowan, U., Johnston, M., Zhang, M.: Developing new fitness functions in genetic programming for classification with unbalanced data. IEEE Trans. Syst. Man Cybern. Part B **42**(2), 406–421 (2012). https://doi.org/10.1109/TSMCB.2011.2167144
4. Cano, A., Zafra, A., Ventura, S.: Weighted data gravitation classification for standard and imbalanced data. IEEE Trans. Cybern. **43**(6), 1672–1687 (2013). https://doi.org/10.1109/TSMCB.2012.2227470
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002). https://doi.org/10.1613/jair.953
6. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995). https://doi.org/10.1007/BF00994018
7. Domingos, P.: Metacost: a general method for making classifiers cost-sensitive. KDD **99**, 155–164 (1999)
8. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
9. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009). https://doi.org/10.1109/TKDE.2008.239
10. He, H., Shen, X.: A ranked subspace learning method for gene expression data classification. In: Arabnia, H.R., Yang, M.Q., Yang, J.Y. (eds.) Proceedings of the 2007 International Conference on Artificial Intelligence, ICAI 2007, Las Vegas, Nevada, USA, 25–28 June 2007, vol. I, pp. 358–364. CSREA Press (2007)

11. Hulse, J.V., Khoshgoftaar, T.M., Napolitano, A.: Experimental perspectives on learning from imbalanced data. In: Ghahramani, Z. (ed.) Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, 20–24 June 2007, vol. 227, pp. 935–942. ACM International Conference Proceeding Series, ACM (2007). https://doi.org/10.1145/1273496.1273614
12. Japkowicz, N.: Supervised versus unsupervised binary-learning by feedforward neural networks. Mach. Learn. **42**(1/2), 97–122 (2001). https://doi.org/10.1023/A:1007660820062
13. Japkowicz, N., Myers, C., Gluck, M.A.: A novelty detection approach to classification. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 1995, Montréal Québec, Canada, 20–25 August 1995, vol. 2, pp. 518–523. Morgan Kaufmann (1995). http://ijcai.org/Proceedings/95-1/Papers/068.pdf
14. Jo, T., Japkowicz, N.: Class imbalances versus small disjuncts. SIGKDD Explorations **6**(1), 40–49 (2004). https://doi.org/10.1145/1007730.1007737
15. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. In: Fisher, D.H. (ed.) Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, 8–12 July 1997, pp. 179–186. Morgan Kaufmann (1997)
16. Lin, M., Tang, K., Yao, X.: Dynamic sampling approach to training neural networks for multiclass imbalance classification. IEEE Trans. Neural Netw. Learn. Syst. **24**(4), 647–660 (2013). https://doi.org/10.1109/TNNLS.2012.2228231
17. Quinlan, J.R.: C4.5: programs for machine learning. Elsevier (2014)
18. Rahman, M.M., Davis, D.: Cluster based under-sampling for unbalanced cardiovascular data. Proc. World Congr. Eng. **3**, 3–5 (2013)
19. Raskutti, B., Kowalczyk, A.: Extreme re-balancing for svms: a case study. SIGKDD Explorations **6**(1), 60–69 (2004). https://doi.org/10.1145/1007730.1007739
20. Shirabad, J.S., Menzies, T.J.: The promise repository of software engineering databases. School of Information Technology and Engineering, University of Ottawa, Canada 24 (2005)
21. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn. **40**(12), 3358–3378 (2007). https://doi.org/10.1016/j.patcog.2007.04.009
22. Wang, C., Hu, L., Guo, M., Liu, X., Zou, Q.: imdc: an ensemble learning method for imbalanced classification with mirna data. Genet. Mol. Res. **14**(1), 123–133 (2015)
23. Yang, S., Khot, T., Kersting, K., Kunapuli, G., Hauser, K., Natarajan, S.: Learning from imbalanced data in relational domains: a soft margin approach. In: Kumar, R., Toivonen, H., Pei, J., Huang, J.Z., Wu, X. (eds.) 2014 IEEE International Conference on Data Mining, ICDM 2014, Shenzhen, China, 14–17 December 2014, pp. 1085–1090. IEEE Computer Society (2014). DOI: https://doi.org/10.1109/ICDM.2014.152
24. Zhuang, L., Dai, H.: Parameter optimization of kernel-based one-class classifier on imbalance learning. JCP **1**(7), 32–40 (2006). https://doi.org/10.4304/jcp.1.7.32-40