A gravitational density-based mass sharing method for imbalanced data classification

Farshad Rahmati¹ · Hossein Nezamabadi-pour¹ · Bahareh Nikpour¹

Received: 9 August 2019 / Accepted: 14 January 2020 / Published online: 23 January 2020
© Springer Nature Switzerland AG 2020

Abstract

Classification is one of the most popular branches of supervised learning algorithms. In the real-world problems, there are some situations in which distributions of the two classes are not the same. This situation is known as a class imbalanced problem. In the past years, several investigations have been done to find a way of handling imbalanced data, which most of them stay in one of two groups, including internal techniques and external techniques. The proposed gravitational density-based mass sharing method (GDMS) is an internal method that is designed based on k-nearest neighbor and fixed radius nearest neighbor (FRNN) rules. GDMS is a new technique that assigns masses to instances based on their local density while considering the global information too. In the labeling phase, GDMS decides based on the sum of gravitational forces coming from the candidates set, which are defined by FRNN rule. The GDMS does not need any parameters to be set in the whole procedure of classification, which is an advantage in comparison with the previous methods. To demonstrate the effectiveness of our proposed method, we use 40 standard datasets from the KEEL repository. The results show the effectiveness and superiority of GDMS compared to the competing methods.

Keywords Classification · Class imbalance · Fixed radius search · Gravitational rule · Instance weighting · Mass sharing

1 Introduction

Today, with the growth of raw data, generation, and accessibility to such data, considerable growth in the advancement of science has accrued. This development shows an excellent opening for the data analysts and computer scientist to play a vital role in a vast variety of states from daily routine life to top security departments, and from information processing for government's budget for the upcoming year to distinguish a patient who has cancer from a healthy one [1]. Supervised learning is one of the essential tasks in machine learning. A supervised learning algorithm analyzes a set of labeled training examples and produces a function to predict the correct output for any other patterns. One of the most popular supervised learning techniques is classification [2, 3] the main goal of

which is to predict the correct label for an unseen sample by studying a group of given training samples. Unlike the usual consideration, distributions of classes are not always the same. There are some situations in which the class distributions are different, i.e., there is a class with much more members than other class or classes. In such cases, a problem appears, which is known as a class imbalanced problem [4–6].

The machine learning society has paid more attention to the imbalanced learning problems' domain in recent years [6–8]. The issue of imbalance learning is expanding, as it is a recurring problem in many real-world applications. Imbalanced datasets are common in medical recorded databases of rare diseases as the number of patients who have rare diseases is much less than those who do not [9]. The same situation occurs

✉ Hossein Nezamabadi-pour, nezam@uk.ac.ir; Farshad Rahmati, rahmatifarshad@eng.uk.ac.ir; Bahareh Nikpour, b.nikpour@eng.uk.ac.ir |
¹Intelligent Data Processing Laboratory (IDPL), Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran.

in fault monitoring issues as the number of non-faulty cases is much more than the faulty ones [10]. Therefore, Imbalanced learning methods could be a proper way of dealing with these kinds of issues. In binary classification problems, the class with fewer instances is defined as the Minor (Positive) class, and the other one is known as Major (Negative) class. In most of the imbalanced data classification problems, the class with fewer instances attracts more interest than others. Thus minor class instances are much more important, and the cost of misclassifying them is more than others [9–11].

Due to the importance of imbalanced data classification, several methods have been suggested that can be categorized into two main groups: internal methods and external methods. Creating a new algorithm or modifying existing ones is the primary purpose of internal methods [12–14], while external methods take advantage of preprocessing to modify the imbalanced distribution [15, 16]. External methods' main advantage is their being independent of the type of classifier, and their drawback is increasing the probability of overfitting and overgeneralization [17].

As the gravitation law principle is simple and its performance is considerable, it has been used in many types of research in recent years. Isaac Newton proposed the universal gravitation law for the first time in 1687. Based on this law, all particles in the universe attract other particles with a force which is proportional to the product of their masses and inversely proportional to the square of the distance between their centers [18]. In recent years, several methods have proposed based on the gravitation law to solve the imbalanced class problems [19, 20]. However, those techniques still have drawbacks that need to be overcome for achieving a satisfying result. Most of the reported algorithms only use global information and need a parameter tuning step, which makes them time-consuming. Also, some drawbacks and defects should be overcome for accessing a satisfying result.

We introduce a new method to overcome the previously methods' drawbacks. With this aim, we assign each pattern a mass based on its importance in classifying procedure by using both local and global information simultaneously. Moreover, there is no need for parameter adjustment in the whole classification process that makes GDMS easier to use. The paper's main purpose is to find a simpler but more efficient instance-based method based on the mass sharing concept to overcome the defects of the earlier classifiers in handling imbalanced data. In summary, the main contributions of this paper are as follow:

- Presenting a new method for imbalanced class problems based on Newton's law of gravity to obtain a better perception of each class's distribution.
- Using both local and global information at the same time to assign a mass to each pattern based on their importance in the classification process.
- Unlike most of the previous methods, GDMS takes advantage of the FRNN rule to gather local information around each sample.
- GDMS does not need any manual parameter selection in its whole procedure that makes it easy to use and adaptive to different kinds of problems.
- The comparison with previous methods on 40 datasets reveals the efficiency and effectiveness of the proposed method.

The rest of the paper is organized as follows: related works are brought in Sect. 2, the proposed method is presented in Sect. 3. The results of Experiments are reported in Sect. 4. Eventually, the conclusion is drawn in Sect. 5.

2 Related works

In recent years, doing research in the imbalanced class domain has been broadly increased. Data Level approaches, which are known as external methods, try to balance the distribution by adding or eliminating some samples from each class. Data level approaches are three types:

- Under-sampling techniques are the first group that seeks to choose a subset of the original dataset by removing samples from the major class.
- Oversampling techniques are the second group that tries to balance the dataset by duplicating minor samples or creating new ones from the existing ones.
- Hybrid techniques are the last ones using a mixture of both under-sampling and oversampling techniques to balance the classes' distribution.

Based on how the algorithm adds or removes samples, different methods of under-sampling and oversampling have been presented [21, 22]. Random oversampling [15], which is the simplest method in this branch, duplicates the minority samples randomly. This method is easy to use, but it may increase the probability of overfitting. Random under-sampling [15], which is another easy method eliminates majority class samples randomly. However, as its removing procedure follows no rule, it may remove useful samples as well. One of the most well-known algorithms in this field is the synthetic minority oversampling technique (SMOTE) [23], which creates synthetic samples for the minor class to make the classes balanced. It selects a group of minority samples in a neighborhood and creates new samples by interpolating between them. In oversampling

techniques, the probability of overgeneralization depends on how synthetic samples are created. Thus SMOTE uses none information for creating samples; overgeneralization is possible.

To enhance the SMOTE algorithm, safe-level-SMOTE [24] has been proposed, which assigns different weights to samples to generate synthetic samples. Some other methods presented to improve SMOTE are hybrid methods such as SMOTE-TL and SMOTE-ENN. SMOTE-TL first generates samples using the SMOTE algorithm and in the second phase, removes redundant samples with Tomek Links (TL) [25]. SMOTE-ENN benefits from the ENN algorithm, which is a prototype selection technique to remove redundant samples produced by SMOTE [25]. Neighborhood Cleaning Method (NCL) [26] is an under-sampling method which first finds three nearest neighbors for each sample. If the sample is a majority class sample and its three nearest neighbors misclassify it, it will be eliminated from the dataset. If the sample belongs to the minority class and its three nearest neighbors classify it wrongly, its neighbors who belong to the majority class will be removed.

The category of algorithmic methods contains other ways of dealing with imbalanced problems. The aim of this is to change baseline learning methods in such a way that they can be used in imbalanced problems [17, 26]. Cost-sensitive methods are included in this category. In imbalanced problems, recognition of minor class instances has a higher priority than major class. Hence, in the cost-sensitive learning methods, the main goal is to minimize both the number of high-cost errors and the overall misclassification cost [27]. It should be noticed that such methods take advantage of the cost matrix to set up a model with the lowest cost [28].

Ensemble learning is another approach that can be used for dealing with imbalanced issues [13, 29, 30]. Ensemble algorithms try to take advantage of a combination of several classifiers to get more accurate results.

The nearest neighbor algorithm (NN) [31] is an instance-based method that is identified as the simplest classification method. It classifies test samples based on the labels of the closest training samples. K-nearest neighbors algorithm (kNN) is a modified version of the NN classifier. It is one of the most powerful data mining techniques and has shown to perform well in many domains [32]. Also, if the size of the training data is big enough, the classification error rate of the nearest neighbor algorithm is only twice the Bayes [33]. However, it does not perform well in imbalanced data problems. Therefore, several methods have been proposed for this purpose that can be categorized into two main groups:

The first group seeks to increase sensitivity to the minor class samples. Exemplar-based nearest neighbor (ENN) [34] and positive biased nearest neighbor (PNN) [35] are

the most popular methods of this branch. As PNN does not have any training phase, it performs faster in the labeling phase and is considered as the modified version of ENN.

The second group tries to take advantage of prior knowledge of the global distribution. The center-based nearest neighbor classifier (CNN) [36] is one of these algorithms. It uses the distance between training samples and centers of their classes as a benchmark to measure how far an unclassified example from training samples is. CNN considers the center-based line passing through training examples and the center of the data class. However, CNN's performance is not acceptable when dealing with datasets that have overlap or comprise small disjuncts. Class confidence weighted k-nearest neighbors (CCW-kNN) [37], class-based weighted k-nearest neighbors (CW-kNN) [19], class conditional nearest neighbors distribution (CCNND) [38] and Localized Informative k-nearest neighbors version (LI-kNN) [39] stay in this branch too. CCW-kNN calculates the mixture model to gain weights and then enforces these weights on different neighbors as a confidence ratio. In addition, CW-kNN calculates the amount of misclassification cost for each class with original kNN to get weights for each sample. Furthermore, informative k-nearest neighbors (IkNN) comprises a localized version of Informative kNN (LI-kNN) and globalized version of informative kNN (GI-kNN). IkNN measures the informativeness between samples by defining a new criterion at the first step. In the second step, it chooses the sample from basic k-Nearest Neighbors as final candidates [39]. GI-kNN is another instance-based method that assumes samples have different amounts of information. It assigns different weights to training samples based on their information and then uses a weighted Euclidean metric to calculate the distances. Most of the kNN based classifiers have defects such as having complex structures, being time-consuming as they need several parameters to be set. Also, they are not achieving satisfying results. As a result, a good classifier is needed to obtain good results on imbalanced datasets. Moreover, dealing with the issues of earlier classifiers seems crucial. An algorithm that performs in this domain and gains acceptable results is the gravitational fixed radius nearest neighbor algorithm (GFRNN) [19]. This method uses Fixed Radius Nearest Neighbor (FRNN) rule to highlight those neighbors that stay in radius R and uses an imbalanced ratio (IR) to weight training samples. The imbalanced ratio is a metric to show the level of inequality of two classes and is computed as follows:

$$IR = \frac{N_{NEG}}{N_{POS}} \quad (1)$$

where N_{NEG} and N_{POS} are the numbers of instances belonging to the negative class and positive class, respectively.

In the decision phase, GFRNN uses the gravitational rule to decide which class is the most suitable one for the unseen sample. Although GFRNN shows considerable progress in imbalanced data classification, it still has defects, as it uses no local information for weighting samples. In other words, global information weakens during the training phase, so it is not enough to achieve a significant result.

3 The proposed method

Classifiers can take advantage of the difference between the distributions of classes for labeling test data. For this purpose, the classifier should gain some information from training samples. It is necessary to choose the type of information that the algorithm wants to use for the classification procedure. There are two types of information: Global and Local.

Most of the previous methods use only one of these two options; thus, their performance is not as great as it could be. Therefore, in this paper, we tend to introduce a new method that takes advantage of both local and global information at the same time. The proposed method procedure consists of three main steps:

- First, the proposed method takes advantage of the FRNN rule to find an appropriate candidate set for a mass sharing algorithm.
- Second, the proposed method assigns each sample a mass using the mass sharing algorithm based on their importance in the classification procedure.
- Finally, the proposed method takes advantage of Newton's universal gravitational law to determine the correct class for test samples.

The main goal of the GDMS algorithm is to introduce a new classifier for imbalanced data classification based on the mass sharing concept to diminish the drawbacks of earlier methods and meanwhile enhance their performance. Moreover, unlike GFRNN which uses the mean squared mutual Euclidean distance between each training sample pairs, the proposed method uses a new strategy to increase the effect of classes' distribution in calculating the radius R . To do so, the proposed method takes advantage of the average distance of each class samples' pair separately which gives a better view of each class distribution.

In classifiers, which are designed to take advantage of local information, selecting a proper neighborhood is a challenge as it plays a critical role in the classification's process, and an inappropriate neighborhood can affect the classifier's generalization. Due to skewed distributions in the imbalanced domain, selecting a large

neighborhood may lead to overgeneralization by misclassifying minority class samples as majority class patterns as there is a significant difference between the number of members of each class in a large neighborhood. It should be noted that the neighborhood should not be too small to be local enough for the samples.

In our method, the training set is called $X_{training}$ which includes both minority (positive) class and majority (negative) class samples, and is defined as follows:

$$X_{Pos} = \{(x_1, c_1), (x_2, c_1), \dots, (x_{N_{pos}}, c_1)\} \quad (2)$$

$$X_{Neg} = \{(x_{N_{pos}+1}, c_2), (x_{N_{pos}+2}, c_2), \dots, (x_{N_{pos}+N_{neg}}, c_2)\} \quad (3)$$

$$X_{training} = X_{Pos} \cup X_{Neg} \quad (4)$$

where c_i is known as the class of training pattern x_i , and is equal to +1 for minority class samples and -1 for majority class samples. Furthermore, the number of training samples $N_{training}$ is equal to the sum of two minority and majority class samples as follow:

$$X_{Pos} = \{(x_1, c_1), (x_2, c_1), \dots, (x_{N_{pos}}, c_1)\} \quad (5)$$

where N_{neg} and N_{pos} are the numbers of majority and minority class samples, respectively.

3.1 Finding candidates set using FRNN rule

In computational geometry, the Fixed Radius Nearest Neighbors algorithm (FRNN) is a variant of the Nearest Neighbors method (NN). In this method, based on the radius R , which is given to the method as an input, a set of patterns is obtained that is known as the candidates set. In the proposed method, training samples and radius R are FRNN rule's inputs. The candidate set for each $x_q \in X_{training}$ is obtained from the FRNN rule as follow:

$$FRNN(X_{training}, R, x_q) = X_{Candidate} = \{x_p \in X_{training}, d(x_q, x_p) < R\} \quad (6)$$

where x_p and x_q are two training samples, $X_{training}$ is the training set, and R is the radius of FRNN rule, which calculated automatically by the algorithm itself. The main advantage of using FRNN instead of kNN goes back to selecting parameter k for kNN, which plays a critical role in reaching the desired results. Moreover, finding the optimal k , which leads to optimal results, is tough and problem-dependent. Furthermore, in our proposed method, FRNN can be more powerful and consistent as we offer a new way of calculating the radius R based on each class's distribution which is defined as follows:

$$R = \frac{1}{2} \times \sum_{j=1}^2 \frac{1}{N_j(N_j - 1)} \sum_{\substack{x_s, x_t \in C_j \\ x_s \neq x_t}} d(x_s, x_t) \quad (7)$$

where N_j is a number of samples in class j and $d(\cdot)$ is the Euclidean distance between two members of the same class which is calculated as follow:

$$d(x_i, x_j) = x_i - x_{j2} \quad (8)$$

In this way, the distribution of each class affects the amounts of radius R , contrary to the way GFRNN employs, which makes the candidates set to be selected more precise. It should be mentioned that we choose Euclidean distance for more simplicity, but it is possible to employ any other appropriate measurements.

3.2 The proposed mass calculation method using mass sharing technique

Most of the recently proposed methods which are based on gravitation only take advantage of either local or global information. That is, while our method tends to combine both local and global information. For this purpose, the proposed method takes advantage of the mass sharing algorithm. The idea comes from the fitness sharing technique, which is originally a niching method in evolutionary computation that permits the evolutionary algorithm to locate several optimal solutions at the same time. Firstly, the proposed method allocates a mass equals to one to all training samples [40]. In the following, based on the mass sharing method, all the training samples are considered as a set while their mass needs to be shared. Mass sharing determines candidates set for each training sample based on the Radius (R), which is obtained in the previous section. After specifying candidates set, the shared mass of each sample is obtained based on the sharing function, which is defined as follows:

$$sh(d(x_i, x_j)) = \begin{cases} \left(\frac{d(x_i, x_j)}{R} \right)^{\left(\frac{1}{IR} \right)}, & \text{if } d(x_i, x_j) < R, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

While $d(x_i, x_j)$ is the Euclidean distance between two samples x_i and x_j of the training set, and IR is the Imbalanced ratio. Afterward, the shared mass of each training sample is calculated as follow:

$$M_{share}(i) = \frac{1}{\sum_{j=1}^{|X_{candidate}|} sh(d_{ij})} \quad (10)$$

where $||$ indicates the cardinality operator, and here is a measure of the number of elements of the set $X_{candidate}$. Eventually, to add global information to the obtained local information, the proposed method takes advantage of IR value. To do so, the proposed method multiply minor class shared mass $M_{share}(\cdot)$ by IR and the final mass of each training sample will be equal to:

$$M(i) = \begin{cases} M_{share}(i) \times IR & \text{if } i \in \text{minority} \\ M_{share}(i) & \text{if } i \in \text{majority} \end{cases} \quad (11)$$

It needs to be mentioned that with this strategy, it is so rare for two samples to obtain the same mass, and each sample is weighted according to its importance in the classification process.

3.3 Classifying test samples using Newton's Universal gravitational law

In the last part, after obtaining each training samples' mass, GDMS uses the gravitation law to classify test samples. Based on Newton's universal gravitational law, every particle attracts all other particles in the universe with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. Therefore, the proposed method considers each training sample as an independent particle with a specific mass, which is obtained in the previous part. Differently, from several previous gravitational based methods, the proposed method does not consider any direction for gravitation. Furthermore, for each test sample y , the sum of gravitational force which is applied to it from the candidates set's members is obtained through the function $F(\cdot)$ as follows:

$$F(y, x_i) = \sum_{x_i \in X_{Candidate}} p_i * G \frac{M(i)}{d(y, x_i)^2} \quad (12)$$

where $M(i)$ is the mass of the training sample x_i and $d(\cdot)$ is the Euclidean distance between the test sample y and training sample x_i . To make this equation simpler, we assume constant G equal to one and p_i , which is a constant equal to:

$$p_i = \begin{cases} -1 & x_i \in \text{majorclass} \\ +1 & x_i \in \text{minorclass} \end{cases} \quad (13)$$

Eventually, the proposed method calculates the sum of forces coming from samples that are located in the candidates set for each class separately and compares them to see which one is stronger. In other words, the

sign of $F(\cdot)$ determines the class of the test sample y , and if $F(y, X_{Candidate}) \geq 0$, then test.

Algorithm 1. The proposed GDMS Algorithm

Input: Training set $X_{training}$ and test set V_{test}

Output: Vector Y_{label} which shows the result of classification for each test sample y_i

1. Calculating Radius R for Mass sharing algorithm based on Eq.7.
2. Assign a mass equal to one to all training samples.
3. Calculate local mass $M_{share}(i)$ after applying mass sharing using Eq.10.
4. Calculate the final mass $M(i)$ for each training sample based on Eq.11.
5. for $i=1$: number of test samples
  - a) Calculate gravitational force between each of the candidates from $X_{Candidate}$ and a test sample y_i , according to Eq.12.
  - b) If $F(y_i, X_{Candidate}) \geq 0$
 $y_{label_i} = +1$
  - c) Else:
 $y_{label_i} = -1$

END

Sample y_i will be classified to minority class, and if $F(y, X_{Candidate}) < 0$, the test sample y_i will be classified to the majority class. The whole procedure of GDMS is summarized as pseudocode given in Algorithm 1.

4 Experiments

In this part, we have applied the proposed GDMS and some representative algorithms to 40 binary-class imbalanced real-world datasets from the KEEL repository (<http://www.keel.es/dataset.php>) to substantiate both the competency and strength of the proposed method. The characteristics of the used datasets are presented in Table 1, including imbalance ratio (IR), which is in the range of 2.46 to 127.42, data sets' size, and data sets' dimension. In addition, 5-fold cross-validation (5FCV) is adopted during the whole experiments to evaluate the performance of the GDMS algorithm in comparison to other representative algorithms. Moreover, three non-parametric statistical tests are used to analyze the results of GDMS and other representative algorithms statistically, which are introduced in Sect. 3.2. The experimentation setting is given in the first part, and the detailed reports on experiments are presented in the rest parts. All the computations are performed on Intel core i7 2670QM processors with 2.20 GHz, 8G RAM DDR3, Microsoft Windows 8.1, and MATLAB environment.

4.1 Performance evaluation

Two common imbalanced indicators are used to measure the results in the imbalanced domain, which are average accuracy (AA) and geometric mean (GM). GM and AA are defined based on the confusion matrix presented in

Table 1 Main characteristics of the selected datasets

Datasets name	Imbalanced ratio (IR)	Datasets size	Datasets dimension
yeast1	2.46	1484	8
new_thyroid1	5.14	215	5
new_thyroid2	5.14	215	5
yeast3	8.13	1484	8
page_blocks0	8.79	5472	10
ecoli-0-3-4_vs_5	9.00	200	7
ecoli-0-2-3-4_vs_5	9.06	202	7
yeast-0-3-5-9_vs_7-8	9.10	506	8
ecoli-0-4-6_vs_5	9.13	203	6
yeast-0-2-5-6_vs_3-7-8-9	9.16	1004	8
yeast-0-2-5-7-9_vs_3-6-8	9.16	1004	8
ecoli-0-3-4-6_vs_5	9.25	205	7
ecoli-0-3-4-7_vs_5-6	9.25	257	7
ecoli-0-1_vs_2-3-5	9.26	244	7
yeast_2_vs_4	9.28	514	8
ecoli-0-6-7_vs_3-5	9.41	222	7
glass-0-4_vs_5	9.43	92	9
ecoli-0-2-6-7_vs_3-5	9.53	224	7
glass-0-1-5_vs_2	9.54	172	9
yeast_0_5_6_7_9_vs_4	9.55	528	8
vowel0	9.97	988	13
ecoli-0-6-7_vs_5	10.00	220	6
glass_0_1_6_vs_2	10.77	192	9
ecoli-0-1_vs_5	11.00	240	6
led7digit-0-2-4-5-6-7-8-9_vs_1	11.21	443	7
glass-0-6_vs_5	11.29	108	9
glass-0-1-4-6_vs_2	11.62	205	9
shuttle_c0_vs_c4	13.78	1829	9
yeast_1_vs_7	14.29	459	8
page_blocks_1_3_vs_4	16.14	472	10
abalone9_18	16.70	731	8
glass_0_1_6_vs_5	20.00	184	9
yeast_1_4_5_8_vs_7	20.08	693	8
yeast_2_vs_8	23.06	482	8
shuttle_c2_vs_c4	24.75	129	9
yeast4	28.68	1484	8
yeast_1_2_8_9_vs_7	30.54	947	8
yeast5	32.91	1484	8
yeast6	41.39	1484	8
abalone19	127.42	4174	8

Table 2 Confusion matrix of prediction and class in the binary-class problems

	Minority class prediction	Majority class prediction
Minority class	True positive (TP)	False negative (FN)
Majority class	False positive (FP)	True negative (TN)

Table 2. Table 2 presents four relationships between the correct class of each instance and the predicted label by the learning method. Based on the confusion matrix, average accuracy (AA) is defined as follows:

$$AA = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) * 100 \quad (14)$$

Moreover, G-mean is calculated as follows:

$$GM = \sqrt[2]{\frac{TP}{TP + FN} * \frac{TN}{TN + FP}} * 100 \quad (15)$$

It should be mentioned that GM measures the accuracy of both positive and negative classes simultaneously. As a result, we can access the maximum accuracy in each class by maximizing this metric.

4.2 Statistical tests

To find out whether there is a significant difference between the results of the used algorithms, three statistical

d for Polynomial kernel is set to 2.

C for RBF kernel is chosen from set $\{10^{-2}, 10^{-1}, 10^0, 10^{+1}, 10^{+2}\}$.

σ for RBF kernel is chosen from set $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^{+1}, 10^{+2}, 10^{+3}\}$.

analysis methods are applied, including the Friedman test, Bonferroni-Dunn test, and Wilcoxon Signed-Rank test. In the following, these tests are briefly elaborated.

- Friedman test [41]: The goal of the Friedman test is to rank the used algorithms for each dataset separately. Its main purpose is to perform a comparison between the average ranks of each algorithm to check whether there is a significant difference between them and the mean rank or not. For each dataset, the Friedman test assigns the rank one to the method with the best performance. The second-best performing algorithm takes the second rank, and so on.
- Bonferroni-Dunn test [41]: To compare the proposed method with other classifiers, we take advantage of the Bonferroni-Dunn test, which controls the family-wise error. To do so, Bonferroni-Dunn seeks to control family-wise error by dividing α by the number of algorithms that are used in the comparison procedure.

4.3 Comparison algorithms and parameters settings

To make a comparison, in the first part, a group of five state-of-the-art methods, including GFRNN [19], CCNND

[38], ENN [34], LI-kNN [39], and PNN [35] is chosen to be compared with the proposed method. Three of this group (GFRNN, CCNND, and LI-kNN) come from distribution-oriented methods, and the remaining two (PNN and ENN) are amongst pattern-based methods. For experiments, the number of neighbors for ENN, PNN, and CCNND are chosen from the set $\{1, 3\}$ while it is chosen for LI-kNN from the set $\{7, 9, 11, 13, 15, 17, 19\}$. Meanwhile, the number of informative points for LI-kNN is chosen from the set $\{1, 3\}$. Also, the confidence level is set at 10% and 20% for ENN and PNN, respectively. Lastly, the parameter z is set to 0.84 for PNN and 1.28 for ENN.

For the second part, nine baseline algorithms including kNN [31], FRkNN [42], C4.5 [43], LR [44], and SVM [45] with three different types of kernel (Linear, Polynomial, and radial basis function (RBF)) are used for comparing with the proposed method. Moreover, the parameter k for kNN and FRkNN is chosen from the set $\{1, 5\}$ since it is mentioned in the literature that these methods perform better with these values in most of the datasets. Eventually, the parameters of SVM are as follow:

4.4 Results of comparing GDMS with other well-known methods

The results of the average accuracy of GDMS and five well-known kNN based methods alongside nine baseline algorithms on 40 datasets are recorded in Fig. 1. Meanwhile, the standard deviations (Std) of 5 folds for all 15 methods are reported besides their results. It is evident that as we compare 15 methods in Fig. 1, the ranks range from 1 to 15. By comparing the final ranks of each algorithm in Fig. 1, it can be shown that GDMS is performing better than other compared methods. By comparing the results of Fig. 1, it is evident that GDMS and GFRNN perform much better than ENN and PNN approaches. Moreover, by comparing GDMS and GFRNN, it can be concluded that taking advantage of both local and global information leads to better results in terms of AA, and Std.

The results of making a comparison of Geometric Mean among GDMS and all other methods on 40 datasets are recorded in Fig. 2. Figure 2 presents the Average results of geometric mean along with the standard deviations (Std) of fivefolds of all compared methods on all datasets. Based on Fig. 2, the methods rank based on their geometric mean from left to right. Based on the results of Fig. 2, it can be concluded that in the GM metric, GDMS achieves the best results as it stays in the best position in Fig. 2. A

Fig. 1 Results of comparing average accuracy (%AA) of competing methods with standard deviations (Std) on all 40 datasets

closer look at the results reveals that for most datasets, GDMS stays among the top three of all algorithms, which demonstrates the stability and constancy of the proposed method. Also. Based on Fig. 2, it is evident that GDMS stays in the best position along with having the lowest value of Std that is another proof of the proposed method's stability in comparison with other methods. Moreover, it should be mentioned that SVM performs better in comparison with other baseline methods in the imbalance domain as it focuses on boundary samples. Although the results of GDMS are more like boundary-sensitive methods, in the main characteristics and the procedure of classifying a query sample, they have a major difference.

Finally, yet importantly, the results of applying non-parametric tests on the results of both Figs. 1 and 2 are reported in Tables 3 and 4. Table 3 reports the average ranks, which are obtained by applying the Friedman test to both Figs. 1 and 2. Based on the results of Table 3,

GDMS stays in the first rank among all other methods in average accuracy and geometric mean. The results of applying the Bonferroni-Dunn test are reported in Table 4. The first column shows the corresponding table's name, while other columns report the p_value of each algorithm. Under each p_value , the comparison between p_value and critical value (CV) is reported. To have a tangible difference between the proposed method and other algorithms, the corresponding p_value should be smaller than the Critical Value (CV).

From the results of both Figs. 1 and 2, it can be concluded that GDMS performs differently from boundary-based methods; meanwhile, their concepts have a tangible difference. Finally, since both GDMS and GFRNN use the same process for classifying test data, there is no difference between the total times that GDMS and GFRNN consume in the test phase. Lastly, from results of both Figs. 1 and 2, it is obvious that GDMS is performing

Fig. 2 Results of comparing geometric mean (%GM) of competing methods with standard deviations (Std) on all 40 datasets

Table 3 Results of ranks obtained by Friedman test on competing algorithms obtained in Figs. 1 and 2

Algorithms	GDMS	GFRNN	5NN	FR1NN	FR5NN	SVM (Linear)	SVM (Poly)	SVM (RBF)	C4.5	LR	1NN	CCNND	ENN	LI-kNN	PNN
Figure 1-AA	3.25	4.25	10.79	9.22	10.93	5.91	8.5	5.58	8.94	14.06	9.22	7.97	5.21	10.5	5.63
Figure 2-GM	3.15	4.03	11.83	9.46	11.06	5.98	8.53	5.22	9.08	14.17	9.46	7.25	5.52	10.3	5.9

Table 4 Results of Bonferroni–Dunn test on competing algorithms obtained in Fig. 2

Algorithms	1NN	5NN	FR1NN	FR5NN	SVM (Linear)	SVM (Poly)	SVM (RBF)	C4.5	LR	GFRNN	CCNND	ENN	LI-kNN	PNN
Figure 2	0	0	0	0	0.004	0	0.037	0	0	0.037	0	0.017	0	0.005
BD	< CV	< CV	< CV	< CV	> CV	< CV	> CV	< CV	< CV	> CV	< CV	> CV	< CV	> CV

(Critical value for Bonferroni – Dunn test = 0.003, $\alpha = 0.05$)

well in both low and highly imbalanced dataset (IR is from range 2.46 to 127.42) and the size of the dataset does not affect the classification procedure (datasets size ranges from 92 to 5472). It should be mentioned that based on all the results of the mentioned figures, it can be seen that GDMS performs more efficient than other compared methods. Moreover, from the speed point of view, GDMS is comparable to other methods as it does not need any parameter selection.

For future works, it could be a good idea to examine other kinds of distance to see if it affects the algorithm's performance. Also, it is possible to extend the algorithm for multiclass problems as well. Also, it is possible to investigate other ways to use global information rather than using IR. It is also possible to search for a better strategy of mixing local and global information to gain better performance in future works.

5 Conclusion

In this paper, we introduce a new method based on the gravitation law called Gravitational Density-Based Mass Sharing method (GDMS) to deal with the problem of imbalanced data classification. GDMS is inspired by the Fixed Radius Nearest Neighbor rule and the concept of Fitness Sharing from evolutionary computing. Then, GDMS takes advantage of Newton's gravitational rule for predicting the correct class of each test sample. GDMS does not need any manual parameter selection in the whole classification procedure. To evaluate the proposed method, an experiment is designed which compares GDMS with 14 other well-known algorithms, including GFRNN, CCNND, ENN, LI-kNN, PNN, kNN, FRkNN, LR, C4.5 and SVM with three different kernels on 40 binary class datasets from KEEL repository. Moreover, the effectiveness of the proposed method is investigated by three nonparametric tests, including the Friedman test, Bonferroni-Dunn test, and Wilcoxon Signed-Rank test to reveal the difference between GDMS and representative methods. The results of the experiments for both Average Accuracy and Geometric Mean demonstrated the efficiency and superiority of the GDMS compared to other representative methods. Ultimately, contributions of this paper can be summarized in designing a new classifier for imbalanced problems based on Newton's gravitation law. The proposed method takes advantage of the Mass Sharing concept to add both local and global information together. This idea leads GDMS to have a more efficient and effective result in dealing with imbalanced class problems than previous methods.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Wang Z, Palade V (2011) Building interpretable fuzzy models for high dimensional data analysis in cancer diagnosis. *BMC Genom* 12:S5
2. García V, Sánchez JS, Mollineda RA (2012) On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowl Based Syst* 25:13–21
3. Yu H, Mu C, Sun C, Yang W, Yang X, Zuo X (2015) Support vector machine-based optimized decision threshold adjustment strategy for classifying imbalanced data. *Knowl Based Syst* 76:67–78
4. Weiss GM (2004) Mining with rarity: a unifying framework. *ACM SIGKDD Explor Newsl* 6(1):7–19
5. Pacheco J, Casado S, Angel-Bello F, Álvarez A (2013) Bi-objective feature selection for discriminant analysis in two-class classification. *Knowl Based Syst* 44:57–64
6. Lin WC, Tsai CF, Hu YH, Jhang JS (2017) Clustering-based under-sampling in class-imbalanced data. *Inf Sci* 409–410:17–26
7. Douzas G, Bacao F (2017) Self-organizing map oversampling (SOMO) for imbalanced data set learning. *Expert Syst Appl* 82:40–52
8. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G (2017) Learning from class-imbalanced data: review of methods and applications. *Expert Syst Appl* 73:220–239
9. Sun Y, Wong AKC, Kamel MS (2009) Classification of imbalanced data: a review. *Int J Pattern Recognit Artif Intell* 23:687–719
10. Nami S, Shajari M (2018) Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors. *Expert Syst Appl* 110:381–392
11. Chan PK, Stolfo SJ (1998) Toward scalable learning with non-uniform class and cost distributions : a case study in credit card fraud detection. In: *Proceedings of the fourth international conference on knowledge discovery data mining*, pp 164–168
12. Barandela R, Sánchez JS, García V, Rangel E (2003) Strategies for learning in class imbalance problems. *Pattern Recognit* 36:849–851
13. Galar M, Fern A, Barrenechea E, Bustince H (2012) A review of ensembles for the class imbalance problem. *IEEE Trans Syst Man Cybern Part C Appl Rev* 42:463–484
14. Ducange P, Lazzerini B, Marcelloni F (2010) Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets. *Soft Comput* 14:713–728
15. Andrew E, Jo T, Japkowicz N (2004) A multiple resampling method for learning from imbalanced data sets. *Comput Intell* 20:18–36
16. Batista GEAPA, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor Newsl* 6:20
17. López V, Fernández A, García S, Palade V, Herrera F (2013) An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. *Inf Sci* 250:113–141
18. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179:2232–2248

19. Zhu Y, Wang Z, Gao D (2015) Gravitational fixed radius nearest neighbor for imbalanced problem. *Knowl Based Syst* 90:224–238
20. Peng L, Yang B, Chen Y, Abraham A (2008) Data gravitation based classification. *Inf Sci* 179:809–819
21. García S, Herrera F (2009) Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. *Evol Comput* 17:275–306
22. Drown DJ, Khoshgoftaar TM, Seliya N (2009) Evolutionary sampling and software quality modeling of high-assurance systems. *IEEE Trans Syst Man Cybern Part A Syst Hum* 39:1097–1107
23. Chawla NV, Bowyer KW, Hall LO (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
24. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2009) Safe-level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: *Proceedings of the 13th Pacific-Asia conference in advanced knowledge discovery data mining*, pp 475–482
25. Verbiest N, Ramentol E, Cornelis C, Herrera F (2014) Preprocessing noisy imbalanced datasets using SMOTE enhanced with fuzzy rough prototype selection. *Appl Soft Comput J* 22:511–517
26. Laurikkala J (2001) Improving identification of difficult small classes by balancing class distribution. In: Quaglini S, Barahona P, Andreassen S (eds) *Artificial intelligence in medicine. AIME 2001. Lecture Notes in Computer Science*, vol 2101. Springer, Berlin, Heidelberg
27. Cheng F, Zhang J, Wen C, Liu Z, Li Z (2017) Large cost-sensitive margin distribution machine for imbalanced data classification. *Neurocomputing* 224:45–57
28. Elkan C (2001) The foundations of cost-sensitive learning. In: *Proceedings of the IJCAI international joint conference on artificial intelligence*, pp 973–978
29. Díez-Pastor JF, Rodríguez JJ, García-Osorio CI, Kuncheva LI (2015) Diversity techniques improve the performance of the best imbalance learning ensembles. *Inf Sci* 325:98–117
30. Błaszczyński J, Deckert M, Stefanowski J, Wilk S (2010) Integrating selective pre-processing of imbalanced data with Ivotes ensemble. *Lecture notes computer science (including subseries in lecture notes in artificial intelligence, lecture notes in bioinformatics)*, vol 6086. LNAI, pp 148–157
31. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. *IEEE Trans Inf Theory* 13:21–27
32. Kononenko I, Kukar M (2007) *Machine learning and data mining*. Woodhead Publishing Limited, Sawston. ISBN 978-1-90427-521-3
33. Bhattacharya G, Ghosh K, Chowdhury AS (2012) An affinity-based new local distance function and similarity measure for kNN algorithm. *Pattern Recognit Lett* 33:356–363
34. Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans Syst Man Cybern* 2:408–421
35. Zhang X, Li Y (2013) A positive-biased nearest neighbour algorithm for imbalanced classification. In: *Proceedings of the lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, vol 7819. LNAI, pp 293–304
36. Gao QB, Wang ZZ (2007) Center-based nearest neighbor classifier. *Pattern Recognit* 40:346–349
37. Liu W, Chawla S (2011) Class confidence weighted kNN algorithms for imbalanced data sets. *Lecture notes in computer science (including Subser. Lecture notes in artificial intelligence, lecture notes in bioinformatics)*, vol 6635. LNAI, pp 345–356
38. Kriminger E, Principe JC, Lakshminarayan C (2012) Nearest neighbor distributions for imbalanced classification. In: *Proceedings of the proceedings of the international joint conference on neural networks*. IEEE, pp 1–5
39. Song Y, Huang J, Zhou D, Zha H, Giles CL (2007) IKNN: informative k-nearest neighbor pattern classification. In: *Proceedings of the European conference principles and practice of knowledge discovery in databases*, pp 248–264
40. Goldberg DE, Richardson J (1987) Genetic algorithms with sharing for multimodal function optimization. In: *Proceedings of the second international conference on genetic algorithms on genetic algorithms and their application*, pp 41–49
41. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
42. Muja M, Lowe DG (2014) Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans Pattern Anal Mach Intell* 36:2227–2240
43. Chawla NV (2003) C4.5 and imbalanced data sets : investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In: *Proceedings of the international conference machine learning work. Learning from imbalanced data sets II*, p 8
44. Menard S (2002) *Applied logistic regression analysis*, 2nd edn. Sage Publications
45. Vapnik VN (1998) *Statistical learning theory*. Wiley-Interscience

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.