International Journal of Machine Learning and Cybernetics (2019) 10:3687–3699
https://doi.org/10.1007/s13042-019-00953-2

ORIGINAL ARTICLE

A Gaussian mixture model based combined resampling algorithm for classification of imbalanced credit data sets

Xu Han[1] · Runbang Cui[2] · Yanfei Lan[1] · Yanzhe Kang[1] · Jiang Deng[2] · Ning Jia[1]

Received: 19 April 2018 / Accepted: 23 April 2019 / Published online: 8 May 2019

Abstract
Credit scoring represents a two-classification problem. Moreover, the data imbalance of the credit data sets, where one class contains a small number of data samples and the other contains a large number of data samples, is an often problem. Therefore, if only a traditional classifier is used to classify the data, the final classification effect will be affected. To improve the classification of the credit data sets, a Gaussian mixture model based combined resampling algorithm is proposed. This resampling approach first determines the number of samples of the majority class and the minority class using a sampling factor. Then, the Gaussian mixture clustering is used for undersampling of the majority of samples, and the synthetic minority oversampling technique is used for the rest of the samples, so an eventual imbalance problem is eliminated. Here we compare several resampling methods commonly used in the analysis of imbalanced credit data sets. The obtained experimental results demonstrate that the proposed method consistently improves classification performances such as F-measure, AUC, G-mean, and so on. In addition, the method has strong robustness for credit data sets.

Keywords Credit scoring · Imbalanced data · Combined resampling · Gaussian mixture model

1 Introduction

With the rapid development of world economy, the loan has become an indispensable part of modern society, but high profit is often accompanied by high risk. One of the major risks comes from the difficulty to distinguish the creditworthy applicants from those who will probably default on repayments [49]. In this context, credit scoring has been identified as a crucial tool to reduce the possible risks and make managerial decisions [60], and one of the most popular application fields for both data mining and operational research [5].

Many techniques have been proposed for credit scoring, from the statistical models to the artificial intelligence methods [23]. Presently, the calculation of early default loan is mainly based on a subjective judgment method. Some of the qualitative criteria of borrowers such as the classic 5C standard have the disadvantages of large subjectivity and randomness [60]. With the wide application of computer and network technology in the banking industry, the artificial approval loans have become unable to meet the needs of the society. The statistical models, data mining, and other methods have been applied to credit scoring [27]. Most classical credit scoring methods are based on the parametric statistical models, such as discriminant analysis [4, 17, 52, 64, 67] and logistic regression [3, 4, 17, 57, 64, 65]. Moreover, recent researches have also implemented the non-parametric methods and computational intelligence technologies such as decision tree [3, 4, 64, 67], neural network [2–4, 17, 64, 67], support vector machine [4, 66] and others.

Most of the traditional statistical models have definite mathematical forms and uncomplicated characteristics. Besides, it is hard to imagine that the complex real world can be described by a limited mathematical formula. An intelligent method represents a kind of data learning algorithm which does not rely on the rule design, its prediction effect is quite good, and the cross-validation results are easily understood by the practical workers.

According to many comparative studies [4, 30, 62], it is not possible to claim the superiority of any method over the other competing algorithms regardless considering the

✉ Ning Jia
jia_ning@tju.edu.cn

[1] College of Management and Economics, Tianjin University, Tianjin 300072, China

[2] QingDao Fantaike Technology Co., Ltd, Qingdao, China

data characteristics. For instance, the noisy samples, missing values, and skewed class distribution may significantly affect the success of most prediction models.

This paper focuses on one of the data characteristics that may have the most influence on the performance of classification techniques: the imbalance in class distribution [12, 25, 33]. While some complexities have been widely studied in the credit scoring literature (e.g., attribute relevance), the class imbalance problem has received relatively little attention so far. Nevertheless, an imbalanced class distribution naturally happens in the credit scoring where, in general, the number of observations in the class of defaulters is much smaller than the number of cases belonging to the class of non-defaulters [54].

In real life, the credit scoring denotes an imbalanced classification problem due to the relatively scarce information on overdue users. According to the statistics, in 2014, the default ratio of China's banking financial institutions reached 1.64%, while in 2013, the default ratio of banking financial institutions was 1.49%. For the commercial banks, at the end of 2013, the default rate was 0.97%. Besides, the default rate of the commercial bank increased by 0.16 percentage reaching 1.13% in 2014, indicating a potential credit risk. Therefore, credit scoring is essential to classify loan applicants into two classes, i.e., normal users (i.e., those who are likely to keep up with their repayments), and overdue users (i.e., those who are likely to default on their loans) [8]. Because of the imbalanced data distribution, it is often difficult to obtain a good performance at most cases by using only the traditional classifiers wherein a balanced distribution of classes is assumed, and an equal misclassification cost is assigned to each class. As a result, the traditional classifiers tend to be overwhelmed by the majority classes ignoring the minority ones, which is not acceptable in many real applications [22].

Therefore, to improve the accuracy of the minority class is an important and meaningful issue. Nowadays, learning the imbalanced data is an important research direction of machine learning because in the real world, the imbalanced data exist in many applications, such as fault diagnosis [44], medical diagnosis [50], intrusion detection [14, 59], text classification [42, 68], financial fraud detection [53], data stream classification [24], natural disasters [48], and so on. In these applications, there are often one or more minority classes possessing very few samples compared with the other classes. Most of the time, the minority classes are more important than the majority classes.

Recently, a variety of methods have been proposed to solve this problem, and they can be divided into four categories: algorithmic-level methods, data-level methods, cost-sensitive methods, and ensembles of classifiers [6, 7, 9, 10, 13, 15, 19, 25]. The cost-sensitive learning methods are mainly considered in the classification. These methods assign different costs to different types of errors, minimizing the number of high-cost errors in the classification and the cost of the error classification [11, 18, 20, 31, 39, 47, 51, 58]. The cost matrix is usually determined by expert opinions. However, this method has not been widely used by scholars because it is very difficult to set up the cost matrix. The integrated learning solves the same machine learning problem by combining multiple learners. Compared with the traditional single learning, the integrated learning has better learning effect and stronger generalization ability. According to the generation method of an individual learner, the current integrated learning methods can be roughly divided into two categories: serially generated serialization methods (such as Boosting) and parallel generated serialization methods (such as Bagging) [22]. The selection of a proper type of combination method and a base learner is still a challenge. The other category is the algorithmic level method which adapts a supervised classifier to strengthen the accuracy towards the minority class. Therefore, this approach creates new classifiers or modify existing ones to tackle the class imbalance problem. Also, this method greatly relies on the classifier nature and most of the works on this method are focused on solving a specific issue. Moreover, it is difficult to develop new algorithms or modify the existing ones [41, 45, 46, 55]. Based on that, the data-level methods that focus on the preprocessing of imbalanced datasets before constructing the classifiers are widely considered in the literature. This is because the data-level approach is more flexible, and data preprocessing and classifier training can be performed independently. In addition, according to Albisua et al. [1] and Galar et al. [22], where a comparative study of numerous well-known method was presented, the combinations of data preprocessing methods with ensembles of classifiers perform better than other methods; besides, focus on data angle is easier to understand and implement.

Data preprocessing methods are based on the resampling of imbalanced training data set before model training. To create the balance, the original imbalanced data set can be resampled by oversampling the minority class [10, 13, 15, 19, 21, 22, 28, 29] or undersampling the majority class [32, 34–38, 40]. Especially, among these two resampling strategies, the undersampling has been shown to be a better choice [22]. However, both of them have drawbacks; namely, oversampling increases the amount of unnecessary information, and undersampling causes the deletion of some information. As a result, more and more scholars study the combination of these two methods. Lin et al. [43] put forward a resampling algorithm combining random undersampling (RUS) and synthetic minority oversampling technique (SMOTE), and good results in the extreme risk early warning in the financial market field were achieved. In addition, Tomek's modification of a condensed nearest neighbor [61] has often been combined with the SMOTE and used as a sampling strategy in experiments. It is worth noting that in order to

achieve a relative balance of two kinds of data, it is inevitable to delete a large number of majority class samples. However, the undersampling part of these two combination methods does not take into account the distribution characteristics of data, which affects the results.

To overcome this shortcoming, we propose a new algorithm for dealing with the credit data sets, which combines oversampling and undersampling. The main contributions are two-fold.

First, we improve the algorithm by replacing the technique whose core is a clustering algorithm with a Gauss mixed model (GMM). The aim of clustering analysis is to group similar objects (i.e., data samples) into the same cluster; thus, the objects in different clusters are different regarding their feature representations. Therefore, the original data in the same groups are replaced by the cluster centers, thereby reducing the size of the majority class. The GMM determines the probability that each data point is assigned to each cluster. Usage of such a probability has many advantages because the amount of information is more than the direct result of clustering.

Second, there are a lot of imbalanced data in the financial field, but some of them are applied to the credit scoring. We propose a new algorithm for dealing with the credit datasets, which combines oversampling and undersampling. The proposed approach is applied to three different credit datasets containing the real business data to tests the performance of the method from three viewpoints.

The numerical results show that the algorithm we propose here is more effective than the existing algorithms. In this paper, we demonstrate that this type of resampling strategy can reduce the risk of removing useful data from the majority class and overfitting risk of the oversampling enabling the constructed classifiers (including both single classifiers and classifier ensembles) to outperform classifiers developed using some other resampling strategy.

The rest of the paper is organized as follows. A brief explanation of resampling techniques to be used in the analysis of the data sets is given in Sect. 2. The experimental data and the criteria used for comparing the classification performance are described in Sect. 3. Experimental results are presented and discussed in Sect. 4. Lastly, conclusions and recommendations for further research work are outlined in Sect. 5.

2 Resampling ensemble algorithm for classification of imbalanced data

At the data level, most popular strategies apply different resampling forms to change the class distribution of the data. This can be done either by oversampling the minority class or undersampling the majority class until both classes become approximately equally represented. Both of these data-level solutions have certain drawbacks because they change the original class distribution artificially. Namely, undersampling may result in discarding potentially useful information on most categories, and oversampling may increase the computational burden of some learning algorithms and produce noise that may result in performance degradation. Hence, this study focuses on the use of the resampling strategies to solve this problem.

2.1 Gaussian mixtures model

In order to make the samples generated by a sampling algorithm more consistent with the true data distribution, the proposed sampling algorithm is based on the Gaussian mixture model (GMM) probability distribution.

The Gaussian mixed model refers to the linear combination of multiple Gaussian functions. The GMM can be considered as a mixture of L Gaussian distributions in a certain proportion. Each Gauss component is determined by mean μ and covariance matrices δ:

$$p(x) = \sum_{i=1}^{L} p(l)p(x|l) = \sum_{l=1}^{L} \pi_l N(\mu_l, \sigma_l), \ \sum_{l=1}^{L} \pi_l = 1. \quad (1)$$

Since the GMM represents a real distribution of simulation data and a semiparametric approximation expression model can approximate it to arbitrary data distribution, the Gaussian distribution assumption of two kinds of samples mixed with some data conforms to parameters obtained by the GMM according to the distributions of two types of parameter estimation. The common method for parameter estimation using the GMM is the Expectation Maximization (EM) algorithm [16].

2.2 Silhouette coefficient

The Silhouette coefficient is a measure of cluster validity. Namely, it is a kind of evaluation measure for clustering effect, and it was originally proposed by Rousseeuw [56]. The Silhouette coefficient combines cohesion and resolution of two factors. Therefore, it can be used to evaluate different algorithms based on the same original data or the effect of different operation modes on the clustering results. The Silhouette coefficient is defined by:

$$Sil = (b(i) - a(i))/\max(b(i), a(i)) \quad (2)$$

where $a(i)$ is an average dissimilarity between object i and any other object of the cluster to which i belongs, and $b(i)$ is the lowest average distance from i to any point in any other cluster that i does not belong to. The cluster with this lowest average dissimilarity is labeled as the "neighbouring cluster" of i because it is the next best fit cluster for point

i. The Silhouette coefficient is in the range $(-1, +1)$ where higher values indicate that the object matches well with its own cluster but is not well matched with the adjacent cluster. If most objects have a high-value Silhouette coefficient, the clustering configuration is appropriate. On the other hand, if many points have a low or negative Silhouette coefficient, the clustering configuration has too many or too few clusters, respectively. The Silhouette coefficient can be calculated using any distance metric, such as Euclidean distance or Manhattan distance [56].

2.3 SMOTE algorithm

A common practice in the classification of an imbalanced data source is to oversample the minority classes. The synthetic minority oversampling technique is one of the most commonly used approaches to address data imbalance problem. The SMOTE is an oversampling approach based on creating the synthetic training examples for interpolation with the minority classes.

The basic assumption of SMOTE is that there exists a virtual positive sample between two real positive samples that are near to each other. Therefore, the SMOTE algorithm tries to artificially create a new positive sample between two real positive samples that are near to each other. Suppose the number of positive samples after oversampling is $(m+1)$ times greater than the original number of positive samples. For each positive sample x_i^{Pos} $(i \in [1, 2, \ldots, S_{Pos}])$, the SMOTE algorithm needs to find m nearest positive samples, x_{ik}^{Pos} $(k = 1, 2, \ldots, m)$. Then, m new positive samples can be artificially created around the original positive sample x_i^{Pos} according to (3). Finally, the number of artificially-created positive samples is $m \times S_{Pos}$.

$$x_{ik}^{Pos-new} = x_i^{Pos} + rand(0,1) \times (x_{ik}^{Pos} - x_{\mathrm{i}}^{Pos}) \quad (i \in [1, 2, \ldots, S_{Pos}], k \in [1, 2, \ldots, m]). \tag{3}$$

In (3), *rand*(0,1) is the function that produces a random value between zero and one. Both the newly created positive samples and the original positive samples are used in training. The number of artificially created positive samples varies with m, which leads to different degrees of balance between positive class and negative class in the final training data set. Besides a successful application in the handwritten character recognition problems, the SMOTE has received considerable interest in the pattern recognition field. In recent years, many scholars have improved the SMOTE. This paper chooses the Borderline-SMOTE algorithm [26].

2.4 Gaussian mixture model based combined resampling algorithm

In this work, we solve the problem of an imbalanced dataset by using the resampling ensemble method.

In the proposed framework, the undersampling is used for majority classes, and the SMOTE oversampling is used for minority classes. Moreover, several different machine learning methods are employed to construct the ensemble. Both undersampling and oversampling can improve the imbalance of the data set. On the other hand, undersampling of big classes could enhance the diversity of the base learners, which is a crucial factor affecting the classification performance [63]. Besides, we try to avoid losing too much information, and we pay attention to the balance between different classes. Therefore, determination of the final size of the processed classes is of great importance. The detailed empirical analysis is given in Sect. 4. The framework of the proposed resampling ensemble algorithm is shown in Fig. 1.

The flowchart of the Gaussian mixture model based combined resampling algorithm (GSRA) is presented in Fig. 1. The processes are as follows. In a given two-class imbalanced dataset D composed of a majority class and a minority class, the majority and minority classes contain M and N data samples, respectively. The first step of the GSRA is to divide the imbalanced dataset into training and testing sets based on the k-fold cross-validation method. The second step is to divide the training set into a majority class subset and a minority class subset. Next, the GMM undersampling method is employed to reduce the number of data samples in the majority class. The minority class uses the SMOTE algorithm to perform the oversampling. The reduced majority class subset is then combined with the increased minority class subset resulting in a balanced training set. Finally, the classifier is trained and tested by using the balanced training and testing sets, respectively.

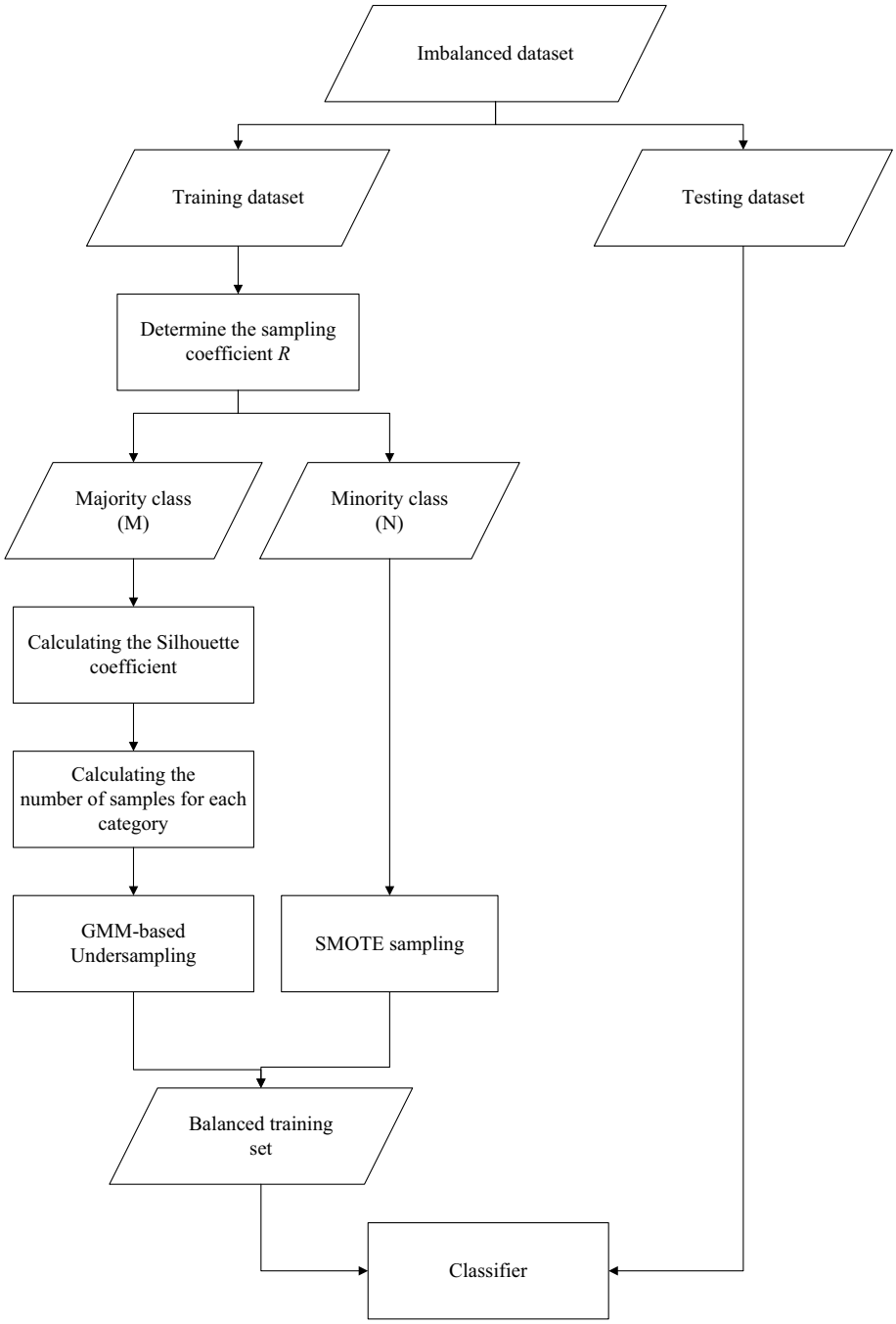

Fig. 1 The flowchart of the Gaussian mixture model based combined resampling algorithm

Algorithm: A Gaussian Mixture Model based combined resampling algorithm

Input: Training Data Set ={(x1,y1),(x2,y2),...,(xm,ym)

Process: Calculate the quantity difference between two classes

Get the quantity difference D between two classes

Determine the sampling coefficient R (R is an arbitrary value between 0 and 1),

The majority class of samples that should be changed $M\_under = D * (1 - R)$

The minority class of samples that should be changed $N\_over = D * R$.

For minority classes: perform oversampling (SMOTE algorithm)

For i=1,2,...,N_over

Randomly select a sample in class N,(xj, yj)

Compute k nearest neighbors of the sample

Generate a new sample of class l, by averaging k nearest neighbors with random weighs

Add a new sample in population

End For

For majority classes: undersampling (GMM)

Use the Silhouette coefficient to determine the best clustering parameters

According to the sample size of each category after the Gaussian mixture clustering and the total amount of needed undersampling, the number of undersampling cycles for each subclass is determined proportionally.

For each subclass of most classes, a sample close to the cluster center is deleted (it is necessary to reduce the number of redundant samples on the premise that the spatial structure information of a subclass is not destroyed, so that each subclass is denser than the other regions in the central area of each subclass, so it should have a higher probability of falling sampling, preserving the representative samples at the same time during the compression of most classes.)

End For

Output: Generate resampling data set

The performance of the GSRA can be explained better using the distributions presented in Fig. 2. The original data distribution D is shown in Fig. 2a, where triangles are a few classes and circles are most classes. The data distribution after the SMOTE sampling is presented in Fig. 2b, where a rectangle denotes a newly synthesized few samples. As it can be seen in Fig. 2, the SMOTE algorithm based on the sample connection interpolation does not consider the distribution of the majority of the data and generates a lot of noise data. In Fig. 2c, for most classes of the GMM modeling and decomposition results, some of the redundant data of the majority of the class will be deleted. In Fig. 2, the result of the GSRA sampling is presented, where it can be seen that the GSRA combines the advantages of the above two kinds of sampling.

3 Data set and evaluation metrics

3.1 Data sets

In this work, three experimental data sets were used (Table 1). The first data set was based on two small-scale data sets, the Australian and German Credit data sets that are publicly available at the UCI repository. The imbalance ratios of these data sets are between 1.24 and 2.3, respectively, with the numbers of collected data samples ranging from 690 to 1000. Both of them are two-class classification data sets. In addition, we adjusted the proportion of these two data sets by reducing the number of samples, and generated several new datasets, as shown in Table 2. Then, we divided the German data set into a certain number of data according to the noise ratio, as shown in Table 3, which will be described further in Sect. 4.

In the second data set, the real data sets from the financial company (It's a consumer financial service provider in China. Its main business is a car loan service for individuals. The average monthly application of customers is about three thousand, and the default rate is about 1.6%.) were used. The enterprise data were obtained from one of the major financial institutions from July 2015 to January 2016, and they are shown in Table 4. In mentioned data sets, a bad customer was defined as someone who had missed three consecutive monthly payments.

To perform classifier training and testing, all of the data sets were divided such that 80% of data were used for training and the rest 20% for testing through the fivefold cross-validation approach.

3.2 Evaluation metrics

We used six metrics to evaluate our model: accuracy, F1-measure, precision, recall, G-mean, and AUC (area under the ROC curve). Each of them is commonly used in classification problem in data mining. In this work, we consider a credit risk prediction as a binary imbalance classification problem. For imbalanced classification problems, the accuracy (or error rate) is not a sufficient evaluation criterion. On the other hand, F-measure and G-mean are two commonly used measures to evaluate the performance of imbalanced data classification.

In the classification process, after all the testing instances are classified, the confusion matrix of classification can be obtained. In the confusion matrix, the representative TP divides the samples that belong to the positive class into a positive class. Similarly, TN, FP, and FN are the number of true negatives, the number of false positives, and the number of false negatives, respectively. The confusion matrix is shown in Table 5. Generally, the minority class is called the

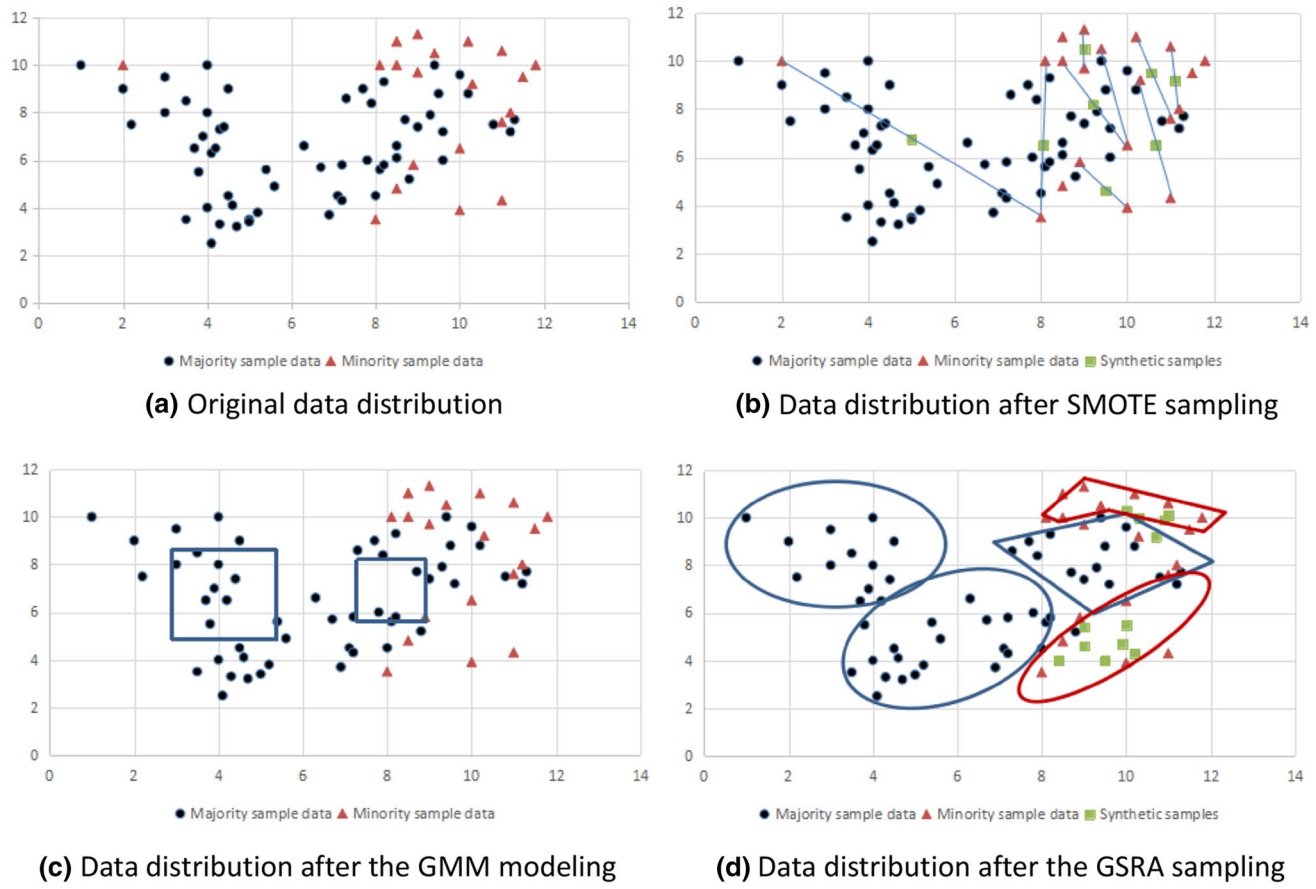

Fig. 2 Data distribution contrast diagram after sampling (i.e., the two-dimensional data set)

Table 1 Data set information

Data set	No. of data samples	No. of features	Imbalance ratio
Australian	690	14	1.24
German	1000	24	2.33

Table 2 Different proportions of data

Data set	No. of data samples	No. of features	Imbalance ratio
German	1000	24	2.33
German_250	950	24	2.8
German_200	900	24	3.5
German_150	850	24	4.6
German_100	800	24	7
Germann_50	750	24	14
Australian	690	14	1.24
Australian_250	640	14	1.56
Australian_200	590	14	1.95
Australian_150	540	14	2.6
Australian_100	490	14	3.9

Table 3 Data set information

Data set	No. of data samples	No. of features	Number of noise samples	Noise ratio (%)
5_German	1000	24	50	5
10_German	1000	24	100	10
20_German	1000	24	200	20
30_German	1000	24	300	30

Table 4 Enterprise data set information

Data set	No. of data samples	No. of features	Imbalance ratio
2015.7_2016.1	2660	25	17.3

negative class. The accuracy, precision, recall, F-measure, and G-mean are defined by (4–8), respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Table 5 Imbalanced confusion matrix of bi-classification problems

	Predicted negative	Predicted positive
Actual negative	TN	FP
Actual positive	FN	TP

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$F - measure = \frac{2TP}{2TP + FN + FP} \tag{7}$$

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}. \tag{8}$$

Namely, accuracy is a common performance index for classifiers to characterize the accuracy of classification; recall reflects the proportion of positive samples that are correctly judged by the classifier to the total positive samples; precision reflects the proportion of real negative samples in negative classes determined by the classifier; F-measure is the weighted harmonic mean of recall and precision; G-mean is another important indicator to measure unbalanced datasets, which balances the sensitivity and specificity, where sensitivity and specificity are the accuracy of the positive and negative classes, respectively.

The other evaluation metric we used is the AUC proposed by Baesens et al. [4]. The AUC relates to the area under the receiver operating characteristic (ROC) curve. The receiver operating characteristic curve (ROC) is a two-dimensional graphical illustration of the trade-off between the true positive rate (sensitivity) and false positive rate (1-specificity). The ROC curve illustrates the behavior of a classifier without consideration of a class distribution or a misclassification cost. In order to compare the ROC curves of different classifiers, the area under the receiver operating characteristic curve (AUC) should be computed [4].

As already mentioned, the indicators used here are all frequently used indicators for the classification in the machine learning domain. These six evaluation indexes are commonly used to evaluate the performance index of classifiers, especially the latter three.

4 Results and discussion

In this section, the experimental results are presented from three aspects, samples distribution, classification performance, and influence of parameters.

4.1 Sampling distribution contrast of resampling algorithm

The working principle of the GSRA sampling and its difference from the SMOTE sampling (high dimensional data are drawn by the t-Distributed Stochastic Neighbor Embedding dimension reduction) are presented in Fig. 3.

The original data distribution used in the experiments is shown in Fig. 3a, where black circles denote rare data, and white circles denote the data majority.

The sampled data distribution after the SMOTE is presented in Fig. 3b, where it can be seen that the SMOTE algorithm did not consider data distribution in most classes. The result of the GMM modeling and decomposition for most classes is presented in Fig. 3c. As we can see from Fig. 3a, some of the data that most classes interact were deleted and the boundaries between the two classes are more obvious. Of course, we can deepen this effect by adjusting parameters The results of the combination of the methods presented in Fig. 3b, c are presented in Fig. 3d.

4.2 Performance comparison of different resampling algorithms

The approach proposed in this work was validated by comparison with thirteen most commonly used state-of-the-art approaches: ClusterCentroids, NearMiss, NeighbourhoodCleaningRule (NCR), OneSidedSelection (OSS), TomekLinks, ADASYN, SMOTE, SMOTE + RUS, and SMOTE + Tomek. The listed approaches are typical representatives of undersampling, oversampling, and combined sampling. In addition, because the Gaussian hybrid clustering is essentially a clustering algorithm, the Kmeans and Affinity Propagation clustering algorithms are compared with the undersampling results.

All of the algorithms were written in Python 2.7 language. The computer used in the experiments had Intel Core i5 2.47 GHz CPU, 4G memory. The operating system was Windows 7.

In the first experiment, the logical regression classifier and the decision tree (DT) classifier were selected as the base classifiers because they are often used as the baseline classifiers in most related studies.

As can be seen in Tables 6 and 7, the GSRA had a good performance on the majority of data sets. For the LR classifier, there were 12 data sets in total, and 11 data sets were

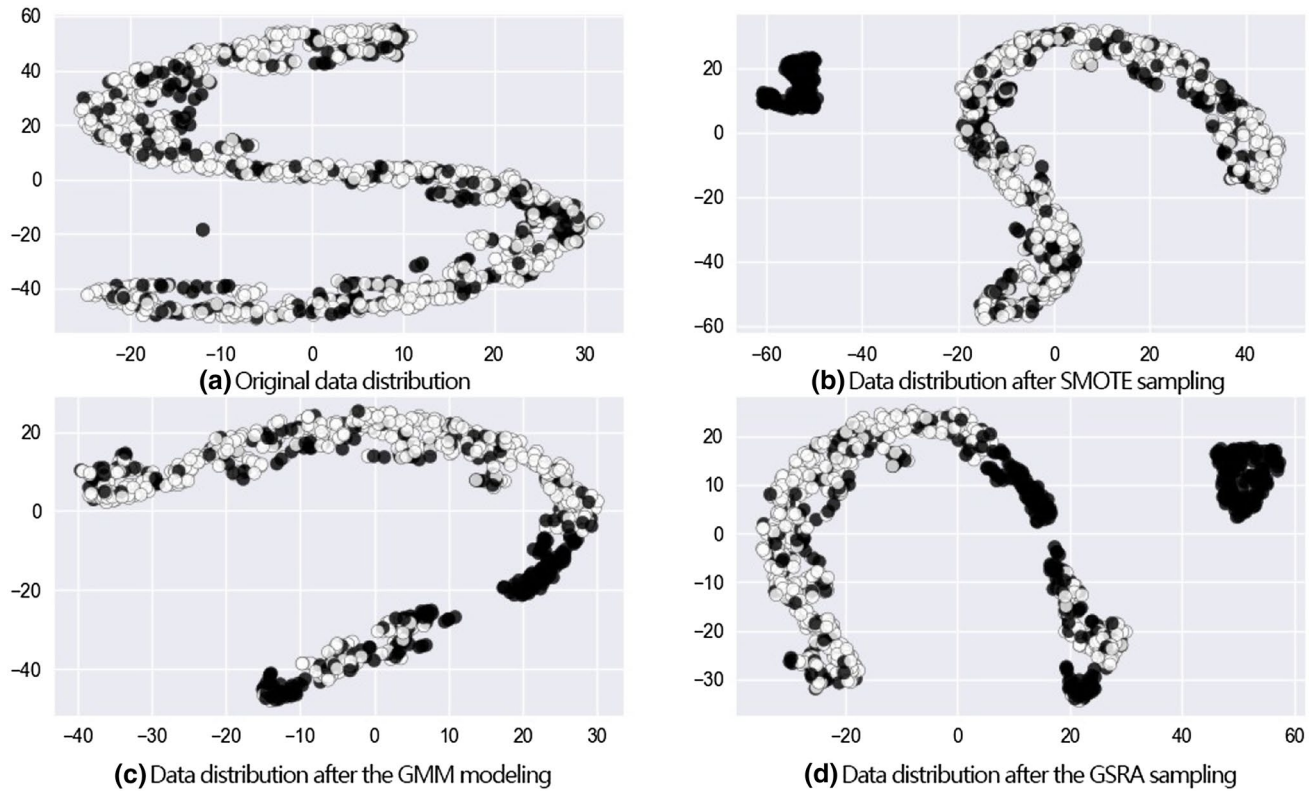

Fig. 3 Data distribution after sampling of data sets (i.e., the German data set)

optimal. On the other hand, for the DT classifier, there were 12 data sets in total, and 8 data sets were optimal.

By detailed analysis of Tables 6 and 7, it was found that performance of the combined sampling method was generally better than that of a single sampling method. Besides, the performance of the GSRA was relatively stable, and the GSRA algorithm had better classification performance than other tested algorithms.

4.3 The effect of sampling coefficient *R* on classification performance

The sampling coefficient *R* of GSRA determines the performance of focused sampling affecting the classification accuracy. To observe the influence of sampling coefficient *R* on classification performance, the experiments with Australian data set and German data set were performed.

In Tables 8 and 9, it can be seen that sampling coefficient *R* directly affects the classification performance. Namely, when *R* was close to 0, the sampling mode tended to a single undersampling, which might cause potential loss of the important data. On the other hand, when *R* was close to 1, the sampling mode tended to a single oversampling, and a large number of new samples were synthesized, which denotes the "overfitting" phenomenon. In Fig. 4, it can be seen that the peak values of Australian data set and German data set range from 0 to 1. Consequently, a mixed sampling mode can avoid the shortcoming of a single sampling mode. In Fig. 4, the focus in on three main indicators: F1-measure, G-mean, AUC.

In Fig. 4, as the value of R increases, under-sampling deletes some information that is not conducive to classification, so that the performance of the classifier gradually increases until the peak value is reached. Then, with the further increase of the oversampling ratio, the overfitting causes a decrease in classification performance, which confirms the above conclusion.

4.4 Algorithm robustness to noise data

Inevitably, there are many noise data in any data set. Noise data denote the wrong values of samples such as an error in category label. In this work, we used the sample noise containing the error class label as an example to study the robustness of the proposed algorithm. In order to systematically verify the algorithm robustness to noise data, we manually added the noise data in the experiment. We selected the German data and observed the algorithm robustness under given noise ratio.

Table 6 LR classifier results

	LR	No resample	GSRA	Affinity-Propagation	Kmeans	ClusterCentroids	NearMiss	NCR	OSS	TomekLinks	ADASYN	SMOTE	SMOTE+RUS	SMOTE+Tomek
1	German	0.7859	0.8141	0.7969	0.7701	0.7885	0.747	0.8389	0.8052	0.7991	0.7444	0.7778	0.7736	0.7897
2	German_250	0.7791	0.8429	0.783	0.7453	0.7913	0.7572	0.8419	0.7923	0.794	0.7213	0.792	0.7897	0.7992
3	German_200	0.7705	0.8486	0.7755	0.743	0.7818	0.7601	0.8221	0.7844	0.7836	0.7221	0.7867	0.7925	0.8109
4	German_150	0.7662	0.8823	0.7764	0.7353	0.7892	0.7328	0.8203	0.7849	0.7638	0.7176	0.7995	0.8058	0.8187
5	German_100	0.7515	0.8936	0.7579	0.706	0.7723	0.7362	0.7992	0.7686	0.7558	0.7366	0.8032	0.8147	0.8249
6	German_50	0.7737	0.9119	0.7664	0.709	0.7936	0.6537	0.7949	0.7752	0.7866	0.7562	0.855	0.8612	0.8690
7	Australian	0.9239	0.9339	0.9193	0.9207	0.9271	0.9228	0.9265	0.9267	0.9275	0.9105	0.9185	0.9160	0.9266
8	Australian_250	0.9148	0.9349	0.9105	0.9179	0.9229	0.9136	0.9168	0.9214	0.919	0.8859	0.9053	0.9051	0.9122
9	Australian_200	0.9132	0.9395	0.9117	0.9133	0.9197	0.9088	0.9172	0.9143	0.9142	0.8778	0.9038	0.9079	0.9062
10	Australian_150	0.9187	0.9419	0.9085	0.9174	0.9191	0.9202	0.9186	0.9155	0.915	0.8644	0.9178	0.9208	0.9232
11	Australian_100	0.9142	0.9483	0.9001	0.9062	0.9202	0.9066	0.9191	0.9094	0.9078	0.8679	0.9162	0.9185	0.9267
12	2015.7_2016.1_ (One car loan)	0.6397	0.8018	0.7024	0.7099	0.713	0.7105	0.7134	0.6892	0.6904	0.7071	0.7879	0.7529	0.7524

Table 7 DT classifier results

	DT	No resample	GSRA	Affinity-Propagation	Kmeans	ClusterCentroids	NearMiss	NCR	OSS	TomekLinks	ADASYN	SMOTE	SMOTE+RUS	SMOTE+Tomek
1	German	0.6455	0.7608	0.6415	0.64	0.75	0.5717	0.6925	0.6331	0.6351	0.714	0.75	0.7364	0.7548
2	German_250	0.6249	0.7686	0.6548	0.6277	0.754	0.638	0.7337	0.6546	0.6511	0.7422	0.7793	0.7593	0.7910
3	German_200	0.6214	0.8043	0.6507	0.6192	0.755	0.62	0.6954	0.6384	0.6265	0.7632	0.7886	0.79	0.7944
4	German_150	0.5783	0.8209	0.643	0.6265	0.8267	0.6567	0.6869	0.5971	0.638	0.8087	0.8043	0.8129	0.8197
5	German_100	0.6164	0.8743	0.6497	0.6011	0.87	0.645	0.7093	0.6209	0.6179	0.8325	0.8521	0.8598	0.8781
6	German_50	0.6157	0.9343	0.5713	0.5684	0.8497	0.69	0.6631	0.5575	0.5464	0.8901	0.9135	0.9027	0.9262
7	Australian	0.8009	0.8545	0.7985	0.8186	0.832	0.8253	0.8055	0.8458	0.8411	0.8154	0.8341	0.8376	0.8485
8	Australian_250	0.8015	0.8568	0.7822	0.8144	0.836	0.8187	0.8069	0.8427	0.8448	0.8221	0.8356	0.8304	0.8343
9	Australian_200	0.8111	0.8800	0.7815	0.7991	0.855	0.8175	0.8111	0.8273	0.8303	0.8452	0.8523	0.8612	0.8375
10	Australian_150	0.8271	0.8914	0.7714	0.8051	0.87	0.8067	0.8302	0.795	0.8031	0.8499	0.8759	0.8809	0.8691
11	Australian_100	0.8055	0.9183	0.7768	0.8032	0.865	0.785	0.8086	0.799	0.8173	0.8877	0.8851	0.8787	0.8839
12	2015.7_2016.1_ (One car loan)	0.5818	0.9274	0.612	0.6009	0.8517	0.6414	0.602	0.604	0.605	0.9395	0.9217	0.9288	0.9366

Table 8 The results of the experiment with German data set under different sampling coefficient *R*

	R = 0	R = 0.1	R = 0.2	R = 0.3	R = 0.4	R = 0.5	R = 0.6	R = 0.7	R = 0.8	R = 0.9	R = 1
German											
Accuracy	72.8571	74.4903	74.3548	74.2241	73.5185	74.5	75.2174	73.8095	72.8555	72.2059	74.2843
Recall	0.7357	0.7561	0.7468	0.7466	0.7444	0.762	0.7609	0.7429	0.7307	0.7324	0.7567
Precision	0.7258	0.7399	0.7421	0.7396	0.7309	0.7377	0.7475	0.7367	0.7267	0.7177	0.7385
F1-measure	0.7303	0.7478	0.7439	0.7421	0.7369	0.7493	0.7533	0.7385	0.7268	0.7245	0.7462
G-mean	0.7281	0.7447	0.7431	0.7414	0.7345	0.7445	0.7513	0.737	0.7267	0.7215	0.7415
AUC	0.8088	0.8141	0.8138	0.8131	0.8017	0.8082	0.8121	0.805	0.8011	0.7888	0.8014

Table 9 The results of the experiment with Australian data set under different sampling coefficient *R*

	R = 0	R = 0.1	R = 0.2	R = 0.3	R = 0.4	R = 0.5	R = 0.6	R = 0.7	R = 0.8	R = 0.9	R = 1
Australian											
Accuracy	86.2953	86.5333	86.4235	85.8333	87.3843	86.3768	86.0558	85.7576	80.9495	86.8254	86.798
Recall	0.8655	0.8987	0.8914	0.8889	0.9093	0.8928	0.8903	0.8879	0.8942	0.8921	0.8989
Precision	0.8414	0.8425	0.8474	0.8398	0.8501	0.8425	0.8399	0.839	0.8527	0.854	0.8475
F1-measure	0.8672	0.8695	0.8679	0.8623	0.8783	0.8679	0.8642	0.8616	0.8724	0.8716	0.8721
G-mean	0.862	0.8646	0.8631	0.8571	0.8728	0.863	0.8599	0.8562	0.8688	0.8671	0.8672
AUC	0.9254	0.9295	0.9312	0.9312	0.9332	0.9304	0.9339	0.9304	0.9291	0.9297	0.9315

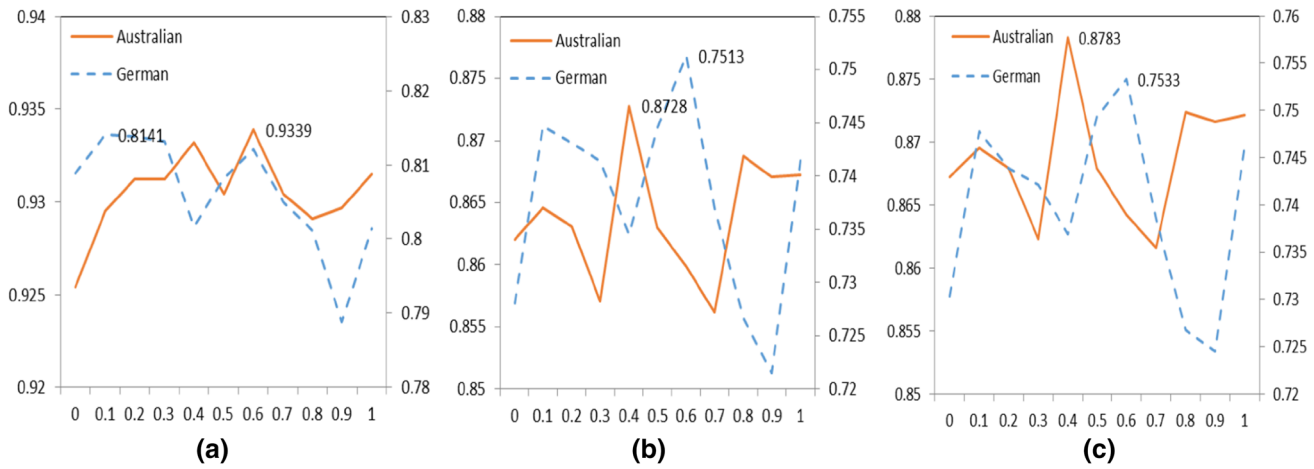

Fig. 4 The value of AUC (**a**), G-mean (**b**), and F1-measure (**c**) under different sampling coefficient *R*

Table 10 Algorithm robustness to noise data

	No resample			GSRA			SMOTE + RUS			SMOTE + Tomek		
	F1_measure	G-mean	AUC	F1_measure	G-mean	AUC	F1_measure	G-mean	AUC	F1_measure	G-mean	AUC
5_German	0.5455	0.6414	0.7762	0.7567	0.7539	0.8397	0.6818	0.6815	0.7582	0.6891	0.6892	0.7633
10_German	0.5019	0.6038	0.7176	0.6844	0.6836	0.7429	0.6786	0.6761	0.7345	0.6774	0.6761	0.7376
20_German	0.45	0.5522	0.6474	0.6141	0.6116	0.6665	0.6171	0.6182	0.6638	0.6143	0.6162	0.6623
30_German	0.4083	0.5034	0.5721	0.5737	0.572	0.6093	0.568	0.5665	0.5925	0.5681	0.5715	0.5974

According to the results presented in Table 10, it can be concluded that GSRA is more robust to noise data than other algorithms, especially at high noise ratio. This is because GSRA considers the distribution of most classes, and can delete data according to the aggregation degree of data, thereby reducing the influence of noise data on sampling and classification learning.

5 Conclusions and recommendations for further work

Focused on the classification problem of imbalanced credit data sets, this paper proposes a Gaussian mixture model based combined resampling algorithm. In the proposed algorithm, the oversampling is used for the minority class, and undersampling is used for the majority class, and the sampling coefficient is determined according to the ratio of the number of minority classes and the number of the majority classes. We compared the proposed algorithm with other commonly used resampling methods and studied their performance on various credit data sets. The classification ability of all tested algorithm was assessed based on the following metrics: accuracy, F1-measure, precision, recall, G-mean, and AUC (area under the ROC curve). The obtained numerical results show that the GSRA is excellent on most credit data and robust to noise data. Also, it was found that with the adjustment of the sampling coefficient R, the classification result changed; thus, the selection of an appropriate sampling coefficient is very important.

In our future work, we will apply the GSRA to more data sets, and we also intend to study the time efficiency of the GSRA to improve its time performance. In addition, the problem of sampling ratio and multi-classification is also worth studying carefully.

Acknowledgements The authors are grateful to the support of the National Natural Science Foundation of China (71671123, 71571132). Meanwhile, the author is grateful for the help of relevant enterprises and professors in the process.

References

1. Albisua I, Arbelaitz O, Gurrutxaga I, Lasarguren A, Muguerza J, Pérez JM (2013) The quest for the optimal class distribution: an approach for enhancing the effectiveness of learning via resampling methods for imbalanced data sets. Prog Artif Intell 2(1):45–63
2. Altman EI, Marco G, Varetto F (2004) Corporate distress diagnosis: comparisons using linear discriminant analysis and neural networks (the Italian experience). J Bank Financ 18(3):505–529
3. Arminger G, Enache D, Bonne T (1997) Analyzing credit risk data: a comparison of logistic discrimination, classification tree analysis, and feedforward networks. Comput Stat 12(2):293–310
4. Baesens B, Gestel TV, Viaene S, Stepanova M, Suykens J, Vanthienen J (2003) Benchmarking state-of-the-art classification algorithms for credit scoring. J Oper Res Soc 54(6):627–635
5. Baesens B, Mues C, Martens D, Vanthienen J (2009) 50 years of data mining and OR: upcoming trends and challenges. J Oper Res Soc 60(1):S16–S23
6. Beyan C, Fisher R (2015) Classifying imbalanced data sets using similarity based hierarchical decomposition. Pattern Recognit 48(5):1653–1672
7. Błaszczyński J, Stefanowski J (2015) Neighbourhood sampling in bagging for imbalanced data. Neurocomputing 150:529–542
8. Brown I, Mues C (2012) An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Syst Appl 39(3):3446–3453
9. Chawla NV (2009) Data mining for imbalanced datasets: an overview. In: Data mining and knowledge discovery handbook. Springer, Boston, MA, pp 875–886
10. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16(1):321–357
11. Chawla NV, Cieslak DA, Hall LO, Joshi A (2008) Automatically countering imbalance and its empirical relationship to cost. Data Min Knowl Discov 17(2):225–252
12. Chawla NV, Japkowicz N, Kotcz A (2004) Editorial: Special issue on learning from imbalanced data sets. ACM Sigkdd Explor Newsl 6(1):1–6
13. Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) SMOTEBoost: improving prediction of the minority class in boosting. Lect Notes Comput Sci 2838:107–119
14. Cieslak DA, Chawla NV, Striegel A (2006) Combating imbalance in network intrusion datasets. In: IEEE international conference on granular computing, IEEE. Atlanta, USA
15. Cohen WW (1995) Fast effective rule induction. In: Twelfth international conference on machine learning. Morgan Kaufmann Publishers Inc. Tahoe City, California, pp 115–123
16. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B Methodol 39(1):1–22
17. Desai VS, Crook JN, Jr GO (1996) A comparison of neural networks and linear scoring models in the credit union environment. Eur J Oper Res 95(1):24–37
18. Domingos P (1999) Metacost: a general method for making classifiers cost-sensitive. In: KDD'99 proceedings of the ifth ACM SIGKDD international conference on knowledge discovery and data mining. San Diego, USA, vol 99, pp 155–164
19. Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27(8):861–874
20. Freitas A (2011) Building cost-sensitive decision trees for medical applications. AI Commun 24(3):285–287
21. Galar M, Barrenechea E, Herrera F (2013) EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. Pattern Recognit 46(12):3460–3471
22. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2012) A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans Syst Man Cybern Part C Appl Rev 42(4):463–484
23. García V, Marqués AI, Sánchez JS (2012) On the use of data filtering techniques for credit risk prediction with instance-based models. Expert Syst Appl 39(18):13267–13276
24. Ghazikhani A, Monsefi R, Yazdi HS (2013) Ensemble of online neural networks for non-stationary and imbalanced data streams. Neurocomputing 122:535–544
25. Guo H, Li Y, Shang J, Gu M, Huang Y, Gong B (2016) Learning from class-imbalanced data: review of methods and applications. Expert Syst Appl 73:220–239

26. Han H, Wang WY, Mao BH (2005) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing. Springer, Berlin, Heidelberg. Ulsan, Korea, pp 878–887
27. Hand DJ, Henley WE (1997) Statistical classification methods in consumer credit scoring: a review. J R Stat Soc 160(3):523–541
28. Hartigan JA, Wong MA (1979) Algorithm AS 136: a k-means clustering algorithm. J R Stat Soc 28(1):100–108
29. Hu S, Liang Y, Ma L, He Y (2009) MSMOTE: improving classification performance when training data is imbalanced. In: 2009 second international workshop on computer science and engineering, IEEE. Qingdao, China, vol 2, pp 13–17
30. Huang Z, Chen H, Hsu CJ, Chen WH, Wu S (2004) Credit rating analysis with support vector machines and neural networks: a market comparative study. Decis Support Syst 37(4):543–558
31. Jackowski K, Krawczyk B, Woźniak M (2012) Cost-sensitive splitting and selection method for medical decision support system. In: Intelligent data engineering and automated learning—IDEAL 2012. Springer, Berlin
32. Li DC, Liu CW, Hu SC (2010) A learning method for the class imbalance problem with medical data sets. Comput Biol Med 40(5):509–518
33. Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intell Data Anal 6(5):429–449
34. Kasabov N (2002) Evolving connectionist systems for adaptive learning and knowledge discovery: methods, tools, applications. In: Proceedings first international IEEE symposium intelligent systems, IEEE. Varna, Bulgaria, vol 1, pp 24–28
35. Kasabov N, Feigin V, Hou ZG, Chen Y, Liang L, Krishnamurthi R, Parmar P (2014) Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke. Neurocomputing 134(4):269–279
36. Kasabov NK, Doborjeh MG, Doborjeh ZG (2016) Mapping, learning, visualization, classification, and understanding of fMRI data in the NeuCube evolving spatiotemporal data machine of spiking neural networks. IEEE Trans Neural Netw Learn Syst PP(99):887–899
37. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: The international joint conference on artiicial intelligence, Morgan Kaufmann. Los Angeles, CA, vol 14, no 2, pp 1137–1145
38. Kotsiantis S, Kanellopoulos D, Pintelas P (2006) Handling imbalanced datasets: a review. GESTS Int Trans Comput Sci Eng 30(1):25–36
39. Krawczyk B, Woniak M, Schaefer G (2014) Cost-sensitive decision tree ensembles for effective imbalanced classification. Appl Soft Comput 14(1):554–562
40. Kubat M, Matwin S (1997) Addressing the curse of imbalanced training sets: one-sided selection. In: the 14th international conference on machine learning. Nashville, TN, USA, vol 97, pp 179–186
41. Lenca P, Lallich S (2008) A comparison of different off-centered entropies to deal with class imbalance for decision trees. Lect Notes Comput Sci 5012:634–643
42. Li Y, Sun G, Zhu Y (2010) Data imbalance problem in text classification. In: 2010 third international symposium on information processing, IEEE. Qingdao, China, pp 301–305
43. Lin Y, Huang X, Xu K (2013) Research on extreme risk warning for financial market based on RU-SMOTE-SVM. Forecasting 32(4)
44. Liu TY (2012) Feature selection based on mutual information for gear imbalanced problem faulty diagnosis. In: IET conference publications, 2012, pp 54–54. https://doi.org/10.1049/cp.2012.0506
45. Liu W, Chawla S (2011) Class confidence weighted kNN algorithms for imbalanced data sets. In: Computer science. https://doi.org/10.1007/978-3-642-20847-8, pp 345–356 **(chapter 29)**
46. Liu W, Chawla S, Cieslak DA, Chawla NV (2010) A robust decision tree algorithm for imbalanced data sets. In: Paper presented at the SIAM international conference on data mining, SDM 2010, April 29–May 1, 2010, Columbus, Ohio, USA
47. Lomax S, Vadera S (2013) A survey of cost-sensitive decision tree induction algorithms. ACM Comput Surv 45(2):1–35
48. Maalouf M, Trafalis TB (2011) Robust weighted kernel logistic regression in imbalanced and rare events data. Comput Stat Data Anal 55(1):168–183
49. Marqués AI, García V, Sánchez JS (2013) On the suitability of resampling techniques for the class imbalance problem in credit scoring. J Oper Res Soc 64(7):1060–1070
50. Mena L, Gonzalez JA (2006) Machine learning for imbalanced datasets: application in medical diagnostic. In: Paper presented at the nineteenth international Florida artificial intelligence research society conference, Melbourne Beach, Florida, USA, May
51. Min F, Zhu W (2012) A competition strategy to cost-sensitive decision trees. Springer, Berlin
52. Altman EI (1968) Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. J Financ 23(4):589–609
53. Perols J (2013) Financial statement fraud detection: an analysis of statistical and machine learning algorithms. Soc Sci Electron Publ 30(2):19–50
54. Pluto K, Tasche D (2005) Estimating probabilities of default for low default portfolios. Dirk Tasche 6(3):79–103
55. Rodda S, Mogalla S (2011) A normalized measure for estimating classification rules for multi-class imbalanced datasets. Int J Eng Sci Technol 3(4):3216–3220
56. Rousseeuw P (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20(20):53–65
57. Steenackers A, Goovaerts MJ (1989) A credit scoring model for personal loans. Insur Math Econ 8(1):31–34
58. Sun Y, Kamel MS, Wong AKC, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit 40(12):3358–3378
59. Thomas C (2013) Improving intrusion detection for imbalanced network traffic. Secur Commun Netw 6(3):309–324
60. Thomas LC, Crook J, Edelman D (2002) Credit scoring and its applications. SIAM, Philadelphia
61. Tomek I (1976) Two modifications of CNN. IEEE Trans Syst Man Cybern SMC 6(11):769–772
62. Wang G, Hao J, Ma J, Jiang H (2011) A comparative assessment of ensemble learning for credit scoring. Expert Syst Appl 38(1):223–230
63. Wang S, Yao X (2009) Diversity analysis on imbalanced data sets by using ensemble models. In: 2009 IEEE symposium on computational intelligence and data mining, IEEE. Nashville, TN, USA, pp 324–331
64. West D (2000) Neural network credit scoring models. Comput Oper Res 27(11):1131–1152
65. Wiginton JC (1980) A note on the comparison of logit and discriminant models of consumer credit behavior. J Financ Quant Anal 15(3):757–770
66. Yang Y (2007) Adaptive credit scoring with kernel learning methods. Eur J Oper Res 183(3):1521–1536
67. Yobas MB, Crook JN, Ross P (2000) Credit scoring using neural and evolutionary techniques. IMA J Manag Math 11(2):111–125
68. Zheng Z, Wu X, Srihari R (2004) Feature selection for text categorization on imbalanced data. Sigkdd Explor 6(1):80–89