Journal of Classification
https://doi.org/10.1007/s00357-024-09491-1

Clustering-Based Oversampling Algorithm for Multi-class Imbalance Learning

Haixia Zhao[1] · Jian Wu[2]

Accepted: 7 August 2024

Abstract

Multi-class imbalanced data learning faces many challenges. Its complex structural characteristics cause severe intra-class imbalance or overgeneralization in most solution strategies. This negatively affects data learning. This paper proposes a clustering-based oversampling algorithm (COM) to handle multi-class imbalance learning. In order to avoid the loss of important information, COM clusters the minority class based on the structural characteristics of the instances, among which rare instances and outliers are carefully portrayed through assigning a sampling weight to each of the clusters. Clusters with high densities are given low weights, and then, oversampling is performed within clusters to avoid overgeneralization. COM avoids intra-class imbalance effectively because low-density clusters are more likely than high-density ones to be selected to synthesize instances. Our study used the UCI and KEEL imbalanced datasets to demonstrate the effectiveness and stability of the proposed method.

Keywords Multi-class imbalance learning · Clustering · Intra-class imbalance · Minority class

1 Introduction

Multi-class imbalance problems occur in many real-world applications, such as medical research, information retrieval, oil reservoir identification, and credit rating models, where certain classes known as minority classes possess far fewer instances than other classes known as majority classes. Learning with multi-class imbalanced data is more complicated as the complex structural features composed of multiple majority and minority classes are highly susceptible to class overlapping and overgeneralization (Wang et al., 2012). The learning algorithms that work well for two-class imbalanced data may fail in multi-class

✉ Haixia Zhao
zhaohx@sxufe.edu.cn

Jian Wu
wujian@sxufe.edu.cn

[1] School of Statistic, Shanxi University of Finance and Economics, Taiyuan 030006, China

[2] Department of Applied Mathematics, Shanxi University of Finance and Economics, Taiyuan 030006, China

Published online: 22 August 2024

imbalance scenarios (Zhou et al., 2006; Hartono et al., 2021). Thus, there is a pressing need to research effective learning methods for multi-class imbalanced datasets.

In imbalanced data learning, oversampling methods have received extensive attention as oversampling a minority class is better than undersampling the majority class, especially for datasets with a high imbalance rate (IR) (García et al., 2012). However, when addressing multi-class imbalanced classification tasks, many oversampling methods may face new challenges. For one thing, increasingly more studies have shown that the distribution of the minority classes significantly affects classification (Rekha et al., 2021; Saez et al., 2015; Wang et al., 2022). If the instances of a minority class are scattered, and even some may be in a region of the majority class, it will exacerbate intra-class imbalance or sample overgeneralization due to the blindness of minority class instance selection (Shaikh et al., 2019). On the other hand, multi-class imbalanced datasets might have multiple minority classes. Certain minority classes instances are easily to be ignored and misclassified by the algorithm as the result of skew distribution with the majority classes.

Currently, most existing resampling methods for handling multi-class imbalanced data implement informed resampling strategies based on the local neighborhood characteristics of minority instances to ascertain the difficulty of their identification. However, it is easy to distort class information and reduce the prediction effect due to multiple majority classes or minority classes. Therefore, in order to strengthen the identification of minority instances and improve the efficiency of multi-class imbalance learning, from the perspective of distribution characteristics of the minority classes, this paper proposes a clustering-based oversampling algorithm (COM) for multi-class imbalance problems. COM spatially clusters the minority instances by identifying soft core instances, designs weights based on the distribution of clusters to sample them, and conducts differentiated oversampling within the selected clusters for minority instances. This means that COM focuses on the overall distribution of minority instances, and at the same time, it considers the intra-class imbalance and overgeneralization in oversampling. The contributions of this study are summarized as follows:

1. Density clustering based on the structural characteristics of the instances is applied to multi-class imbalance learning, which can learn the sample information of the minority class well
2. COM effectively avoids the overgeneralization of samples and solves the problem of intra-class imbalance in imbalanced data learning, which greatly improves the efficiency of multi-class imbalance learning.
3. The effectiveness of the COM algorithm is verified via experiments that COM is superior to other methods in its average classification ability between any two classes.

2 Related Works

Considering that our focus is multi-class imbalanced learning on data structure characteristics, we provide a short review of the learning techniques on the data level in multi-class imbalance scenarios.

For multi-class imbalanced data with complex structures, intra-class imbalance and overgeneralization that often result from oversampling are the main factors contributing to the difficulty of learning. Unreasonable oversampling methods are likely to yield noisy samples or overlapping of classes, which can negatively impact the recognition of the minority class. The usual method is class decomposition, which converts the multi-class imbalance problem into several two-class imbalance problems. Methods of two-class imbalance learning, e.g., OAO and OAA (Dong et al., 2022; Kang et al., 2015; Li et al., 2020), are now effective in dealing with the original multi-class problem. Wu et al. (2010) studied multi-class imbalance learning via clustering and decomposing the majority class. The majority class is clustered by the *k*-means algorithm and decomposed into clusters of equal numbers. Then, each cluster is combined with the minority class instances to form many two-class imbalanced datasets, and then, random oversampling is used to solve the imbalance. Unfortunately, such techniques often result in the loss of class information. The classification effect may decline because the sample information of all classes is not used in training the classifier.

To improve the identification of minority classes, Lin et al. (2013) proposed a memetic algorithm to optimize a radial basis neural network. Next, a dynamic oversampling algorithm uses the SMOTE method to oversample the class with the lowest classification accuracy. Different instances have been given different sampling probabilities based on the multi-layer perceptron classifier (Fernandez et al., 2011). During the training process, a higher sampling probability is given to the minority class to improve the classification accuracy of the minority class instances. The ensemble learning method can also solve multi-class imbalance problems (Krawczyk et al., 2020; Liu et al., 2021). This method generally employs the boosting algorithm, which converts the imbalanced dataset into imbalanced subsets, then implements resampling to train the overall model. Some studies combine the ensemble algorithm with a feature selection algorithm (Guo et al., 2016; Hartono et al., 2021) to solve the problem of overlapping classification boundaries, thus improving the identification of minority class instances. Experiments have shown that this learning technique improves classification but is not independent of specific classifiers.

Abdi and Hashemi (2015) recently proposed an oversampling method based on Mahalanobis distance to solve multi-class imbalance problems. Since the synthesized instance is located on the contour line of the ellipse, the synthesized and the original instances are guaranteed to have the same distance from the class center. However, this method focuses on the instances in the concentrated area of the minority class. It does not consider the boundary instances or the small separation items in the minority class sufficiently. Generative direction (Tang et al., 2017) is becoming an increasingly popular method to avoid the randomness of synthesized instances. For each minority class instance, it selects *k*-nearest sample points of the same class, so the instance has *k* different generation directions. According to the generation weights of different directions, directions are selected to introduce the same number of synthesized samples. However, this method only divides the minority class into outstanding instances and trapped instances (Zhu et al., 2017), which cannot fully reflect the structural characteristics of the minority class.

For multi-class imbalance learning, although a great deal of research works has been done, there are certain aspects that could be even better. Based on the above analysis, the current research strategies are not based on the consideration of the overall distribution of multi-class imbalanced data and thus either may cause the loss of class information or exacerbate the imbalance problem affecting the identification of minority class instances.

3 COM: A Clustering-Based Oversampling Algorithm for Multi-class Imbalance Problems

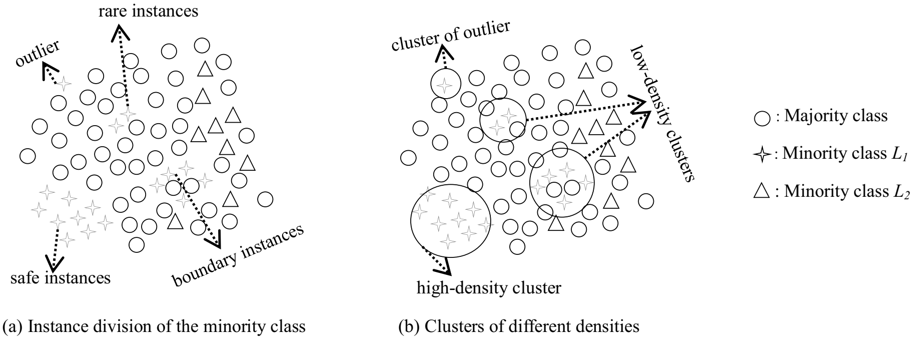
In imbalance learning, one of the main factors causing learning difficulties is the complex distribution characteristics of the dataset. As shown in Fig. 1a, the minority class L_1 has a serious intra-class imbalance problem, in which the instances are divided into four categories: safe instances, boundary instances, rare instances, and outliers (Napierala et al., 2016). Since there are relatively few instances in the minority class, the distribution of the minority class cannot be fully expressed; for example, the outlier is most likely a rare valid instance that cannot be represented by other instances, so it cannot be simply deleted, otherwise it will lead to the loss of information.

To reflect the structural characteristic of minority classes as completely as possible, COM handles the multi-class imbalanced data based on the clustering structure of the minority class. We can observe from Fig. 1b that the density of each cluster is different, and the difficulty of learning the instances in each cluster is also different. If we randomly synthesize the minority class instances, it may aggravate the intra-class imbalance, so we should appropriately increase the synthesis of the instances in low-density clusters to improve the recognition of such instances.

In addition, in the process of synthesizing the minority class instances, if the synthesis instances are inserted between various clusters, it will inevitably aggravate the overgeneralization of samples and affect the efficiency of imbalance learning. Therefore, in this study, different oversampling was carried out in cluster to achieve the purpose of alleviating the intra-class imbalance while avoiding overgeneralization.

Based on the above analysis, the key research ideas of COM can be described as follows:

1. Minority class instances are clustered using density clustering based on the distribution characteristics of the minority class



(a) Instance division of the minority class

(b) Clusters of different densities

Fig. 1 A multi-class imbalanced data

2. Sampling weights are assigned to each cluster according to the number of instances in the cluster and the distance between the instances. A high weight is assigned to a cluster with low density, and the oversampling probability of the instances in this cluster will be high, otherwise the opposite.
3. Various oversampling techniques can be performed in different clusters according to each cluster's structural characteristics.

3.1 Density Clustering for the Minority Class

Research shows that the primary factor affecting classification lies in the occurrence of difficult samples in the dataset. Thus, one effective way to solve imbalanced learning is by analyzing the structural characteristics of imbalanced datasets, especially the instance structure characteristics of the minority class.

Set $D = \left\{ \left(x_1, y_1\right) \cdots, \left(x_i, y_i\right), \cdots \left(x_n, y_n\right) \right\}$ as a multi-class imbalanced dataset with sample size n. $x_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ is an instance of dimension d. The class label variable for instance $\boldsymbol{x}_i$ is y_i. Set the class L in the dataset as the minority class and the set constituted by the L class instances as S^L.

Definition 1 (Zhu et al., 2017): Soft core instance. For any $x_i \in S^L$, if the proportion of L class instances in its k-nearest neighbors set is not less than rTh, then x_i is a soft-core instance of class L.

Definition 2 (Zhu et al., 2017): g_k^L -neighborhood. For any soft-core instance of class L, $\boldsymbol{x}_i$, its g_k^L-neighborhood is defined as an instance set:

$$g_k^L\left(x_i\right) = \left\{x_i\right\} \cup \left\{x_j \in x_i.N_k | x_j \in S^L\right\} \cup \left\{x_j \in x_i.RN_k | x_j \textit{ is a soft core instance of class } L\right\}$$

where N_k and RN_k denote k-nearest and reverse k-nearest neighbor sets, respectively.

The g_k^L-neighborhood satisfies reflexivity and symmetry: for any two soft core instances xi and xj, if $x_j \in g_k^L(x_i)$, then $x_i \in g_k^L(x_j)$.

To better characterize rare instances in the minority, this paper defines soft core instances as follows.

Definition 3 :Soft core instance. For any , if the number of L class instances in its k-nearest neighbors set is not less than 1, then x_i is a soft-core instance of class L.

The clustering in this part mainly includes two steps: First, a soft-core instance set Ω of the minority class is constructed. For any $x_i \in S^L$, if $\boldsymbol{x}_i$ meets Definition 3, it is added to set Ω and given its g_k^L-neighborhood. Second, density clustering is performed on the minority class. The clustering process defines the cluster by determining its g_k^L-neighborhood of the instances in set Ω For the specific implementation of clustering, you can refer to the definition of a cluster function in reference (Zhu et al., 2017).

Let the clustering result be $C = \{C_1, \cdots, C_i, \cdots, C_m\}$. C_i represents the ith cluster, and the cluster label of instance x_i is denoted $x_i.c$. The clustering process in the COM oversampling method is then implemented as follows.

Algorithm 1 Clustering algorithm based on the structural characteristics

```
Input: $\boldsymbol{D}$: a multi-class imbalanced dataset; $k$: neighborhood parameter.
Output: Cluster partition $C = \{C_1, C_2, \cdots, C_m\}$.
Process:
Initialize the soft core instances set to be empty, $\Omega \leftarrow \emptyset$
for each $\boldsymbol{x}_i \in S^L$ do
    Tem ← all $L$ class instances in $\boldsymbol{x}_i.N_k$
    if len(Tem)≥1 then
        $\Omega \leftarrow \Omega \cup \{x_i\}$
    end if
end for
for each $\boldsymbol{x}_i$ in $\Omega$ do
    Determine the $g_k^L$-neighborhood of sample $x_i$.
End for
Initialize the cluster number, $m \leftarrow 0$
Initialize the cluster label of all class L samples to 0, $\boldsymbol{x}_i$.c=0($x_i \in S^L$)
Cluster the soft core instances in $\Omega$.
Output the clustering result.
```

3.2 Acquiring the Sampling Weights for the Clusters

Sampling weights are assigned to the instance clusters formed by clustering to overcome the intra-class imbalance of the minority class. For a cluster with sparse instances of the minority class, the instances in the cluster are often difficult to learn. A higher weight is assigned to these clusters to increase the probability of instance synthesis and improve the degree of recognition of such instances. Conversely, a low weight is usually assigned to a cluster with a high density of minority class instances because learning instances in the cluster are often safe. Therefore, the weight of the cluster will depend on the density of the minority class samples in the cluster. Cluster weighting consists of measuring and converting cluster density into sampling weight.

3.2.1 Density of Clusters

The cluster density is usually related to the distance between its instances and is high if the distance between instances is relatively small. However, if the distance is large, the cluster density is low. To measure the cluster density, the following formula is usually used to calculate the average distance between instances:

$$\mathrm{avg}(d_{C_i}) = \frac{2}{|C_i|(|C_i| - 1)} \sum_{1 \le i < j \le |C_i|} \mathrm{dist}(x_i, x_j) \tag{1}$$

Of these, C_i is the ith cluster, $|C_i|$ represents the number of instances in the cluster, and dist (x_i, x_j) is the Euclidean distance between instances in the cluster.

The cluster density is also related to the data dimension d. Using the average distance between instances, the cluster density is defined as follows:

$$\text{density}\left(C_i\right) = \frac{\left|C_i\right|}{\left[\text{avg}\left(d_{C_i}\right)\right]^d}. \tag{2}$$

The outliers in the minority class are often labeled separately after clustering, usually far away from the instances forming clusters or even located in the regions of other classes. Outliers often affect the classification effect. Imbalanced datasets have fewer minority instances, which is insufficient for representing the distribution of the minority class. The outliers are likely to be rare instances that are not fully represented. Especially in datasets with high IR, deleting outliers directly will likely cause the loss of important information.

To ensure the integrity of the minority class information, this paper regards each outlier separately as a cluster composed of the instance. Then, giving sampling weight to it and increasing the proportion of such instances helps the classifier to learn more useful information. Here, the distance between the outlier and its nearest instance is the average distance of instances within the cluster. Then, formula (2) is used to calculate the cluster density of this kind of cluster.

3.2.2 Sampling Weight

To overcome the influence of intra-class imbalance on classification, the selection weight of each cluster is defined as exp(−density(C_i)). For low-density clusters, there is a greater probability of being drawn to generate synthetic instances of the minority class. Conversely, a small probability is given to high-density clusters because these instances are located in safe areas of the minority class and are not difficult to classify. In the process of imbalance learning, the influence of imbalance between classes is considered based on the distribution of the minority class instances, alleviating the influence of the intra-class imbalance of the minority class on the classification effect.

Converting each cluster's selection weights into a probability distribution is necessary. The sum of the selection weights of all clusters is used to standardize the selection weights of each cluster. That is, the sampling weights of each cluster can be obtained as follows:

$$\text{weight}\left(C_i\right) = \frac{\exp\left(-\text{density}\left(C_i\right)\right)}{\sum_{i=1}^{m} \exp\left(-\text{density}\left(C_i\right)\right)}. \tag{3}$$

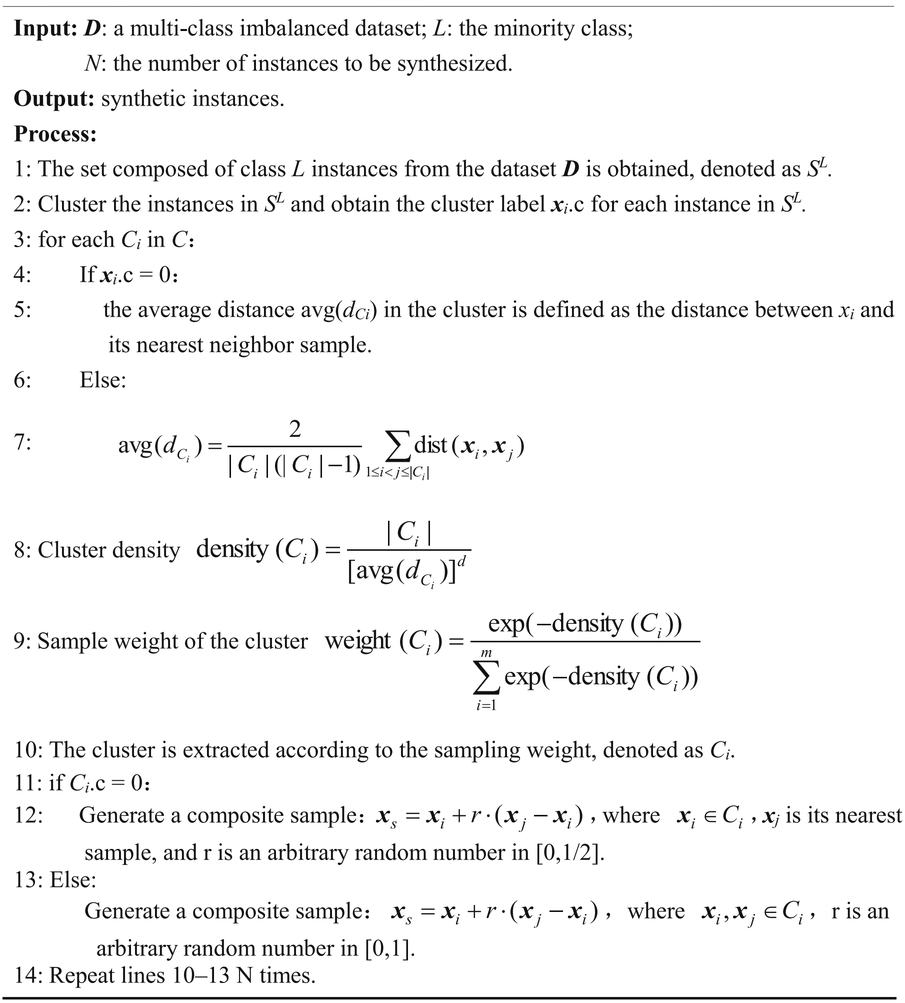
3.3 Generating Synthetic Instances

For a cluster extracted according to the sampling probability, the interpolation method is used to oversample the minority instances in the cluster. Set x_i and x_j as any two instances; then, a new instance is interpolated between the two instances $\boldsymbol{x}_s = \boldsymbol{x}_i + r\,(\boldsymbol{x}_j - \boldsymbol{x}_i)$, where r is any random number in the interval [0,1].

Any two instances in a high-density cluster can be chosen in oversampling, and a new instance is synthesized by the above interpolation method. For a cluster formed by an outlier in the minority class, interpolation is performed between the outlier and its nearest instance to avoid overfitting problems. Since the nearest neighbor instance of an outlier must belong to other classes, r is set as a random number in the interval [0,1/2] during interpolation to ensure that the synthetic instance is as close to the minority class instances as possible.

The specific implementation process of the COM oversampling algorithm is as follows.

Algorithm 2 Clustering-based oversampling algorithm for multi-class imbalance learning (COM)



```
Input: $\boldsymbol{D}$: a multi-class imbalanced dataset; $L$: the minority class;
       $N$: the number of instances to be synthesized.
Output: synthetic instances.
Process:
1: The set composed of class $L$ instances from the dataset $\boldsymbol{D}$ is obtained, denoted as $S^L$.
2: Cluster the instances in $S^L$ and obtain the cluster label $\boldsymbol{x}_i$.c for each instance in $S^L$.
3: for each $C_i$ in $C$：
4:     If $\boldsymbol{x}_i$.c = 0：
5:         the average distance avg($d_{Ci}$) in the cluster is defined as the distance between $x_i$ and
           its nearest neighbor sample.
6:     Else:
7:         $\mathrm{avg}(d_{C_i}) = \frac{2}{|C_i|(|C_i|-1)} \sum_{1 \le i < j \le |C_i|} \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$
8: Cluster density $\mathrm{density}(C_i) = \frac{|C_i|}{[\mathrm{avg}(d_{C_i})]^d}$
9: Sample weight of the cluster $\mathrm{weight}(C_i) = \frac{\exp(-\mathrm{density}(C_i))}{\sum_{i=1}^{m} \exp(-\mathrm{density}(C_i))}$
10: The cluster is extracted according to the sampling weight, denoted as $C_i$.
11: if $C_i$.c = 0：
12:     Generate a composite sample：$\boldsymbol{x}_s = \boldsymbol{x}_i + r \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i)$, where $\boldsymbol{x}_i \in C_i$, $\boldsymbol{x}_j$ is its nearest
        sample, and r is an arbitrary random number in [0,1/2].
13: Else:
        Generate a composite sample：$\boldsymbol{x}_s = \boldsymbol{x}_i + r \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i)$，where $\boldsymbol{x}_i, \boldsymbol{x}_j \in C_i$， r is an
        arbitrary random number in [0,1].
14: Repeat lines 10–13 N times.
```

4 Experimental Study

4.1 Setup

4.1.1 Experimental Data

The multi-class imbalanced datasets selected for the experiment were from the UCI and KEEL databases. Some datasets with very few classes (e.g., the original *E. coli* dataset) had only two samples labeled 2 and 3. Few classes are combined with adjacent classes to ensure the classification effect. Specific data characteristics and distribution information are shown in Table 1.

The feature description of the data in Table 1 includes the sample size (S), number of features (F), number of classes (C), class distribution, and imbalance rate of the dataset. The class distribution represents the number of instances in each class, the bold font indicates that the class is a minority class, and the corresponding imbalance rate is reflected in the column IR_i of the minority class. The IR column is the overall imbalance rate of the multi-class imbalanced dataset.

4.1.2 IR of Multi-class Imbalance Learning

There is no standard definition for the IR of multi-class imbalanced datasets, and different studies give the IR of datasets according to their own research requirements. The definition of the average IR (Zhu et al., 2019) of the dataset we used in our study was.

$$\mathrm{IR} = \frac{1}{l}\sum_{i=1}^{l}\mathrm{IR}_i, \mathrm{IR}_i = \frac{\sum_{q\neq i} n_q}{l \times n_i},$$

Where IR_i is the imbalance rate of class L_i in the dataset, and n_q and n_i are the numbers of instances in classes L_q and L_i, respectively. The fewer the instances a class contains, the higher the IR for that class is. If each class has the same number of instances, then the imbalance rate of each class is

$$\mathrm{IR}_i = \frac{l-1}{l},$$

where IR is approximately 1 if the number of classes is large enough.

Table 1 Description of characteristics of datasets

Data	S	F	C	Class distribution	IR_i	IR
Balance	625	4	3	288/288/**49**	3.92	1.57
Cleveland	297	13	5	160/54/35/35/**13**	4.37	1.69
Dermatology	358	34	6	111/60/71/48/48/**20**	2.82	1.14
Ecoli	336	7	5	143/77/**39**/**25**/52	1.53/2.49	6.79
Glass	214	9	6	70/76/**17**/**13**/**9**/29	1.93/2.58/3.80	1.67
Newthyroid	215	5	3	150/**35**/**30**	1.71/2.06	1.31
Winequality-red	1599	11	6	**10**/**53**/681/638/199/**18**	26.48/4.86/14.63	7.94
Pageblocks	548	10	4	492/**33**/**11**/**12**	3.90/12.20/11.17	6.83
Vowel5	990	13	5	180/**90**/360/270/**90**	2.0/2.0	1.16
Zoo	101	16	7	41/20/**5**/13/**4**/**8**/10	2.74/3.46/1.66	1.56

According to the literature, the threshold of IR is set at 1.5. Classes with an imbalance rate higher than 1.5 are minority classes, while the remaining are majority classes. In this study, only minority classes with an imbalance rate higher than 1.5 were oversampled. For simplicity, the target number of instances for each minority class is set as the average sample size of all majority classes.

4.1.3 Base Classifier and Compared Algorithms

To verify the superiority of the COM learning method in classifying multi-class imbalanced data, our study chose several learning methods—random oversampling (ROS), SMOTE (Chawla et al., 2002), Borderline-SMOTE (B-SM) (Han et al., 2005), and ADASYN (He et al., 2008)—and compared their balance effect on the multi-class imbalanced data. The number of nearest neighbors for each of them was optimally selected among 1, 3, 5, and 7.

For a reasonable evaluation of the data-balancing effect of the compared methods, the study selected decision tree (DT), *k*-nearest neighbor (KNN), and multi-layer perceptrons (MLP) with a single hidden layer as the classifiers. According to the classification effects of classifiers with different evaluation criteria, several resampling methods were compared, and the Euclidean distance was used to measure the distance between instances.

In our experiments, fivefold cross-validation was applied to evaluate the performance of the algorithms; that is, in each stage, 80% of the data were used for training, and 20% were used for testing, and it was ensured that both the training and test sets contained samples of each class. The average value of five tests was used to evaluate the classifier's performance.

4.2 Experimental Results and Analyses

Since total accuracy is not appropriate for multi-class imbalanced data, the micro-F1, MG, and MAUC values were used to compare the performance of the classifiers. The micro-F1 value is denoted as F1 for simplicity. The results in Table 2 are the average values and average rankings of each resampling method for every combination of the datasets, three evaluation metrics, and three classifiers. In addition, we included the performance of the classifiers when oversampling was not used, so they were ranked from 1 to 6. The three evaluation metrics showed that the larger the value, the better the classification. The resampling methods were sorted according to the classification effect. Therefore, if the average ranking is smaller, the classification effect of the resampling method is better. Conversely, the effect is worse. The bold font in the table indicates the resampling method with the best effect.

4.2.1 Average Value

It can be seen from the average values in Table 2 that the COM oversampling method has the best effect when DT is used for classification—its average value for each evaluation metric is the highest, indicating that its data balancing is significantly better than that of other resampling methods. When the KNN and MLP classifiers are used, the COM oversampling method shows obvious advantages in two of three evaluation metrics. Its average F1 and MAUC values are the highest.

Table 2 Average performance results of the oversampling methods across the datasets

Classifier	Metric	None	COM	ROS	SMOTE	ADASYN	B-SM
DT	F1	0.806 ± 0.161	**0.809** ± 0.164	0.796 ± 0.159	0.793 ± 0.164	0.794 ± 0.167	0.790 ± 0.164
	Ranking	2.50	**2.00**	3.60	4.0	3.60	4.60
	MG	0.499 ± 0.352	**0.546** ± 0.319	0.499 ± 0.333	0.525 ± 0.313	0.488 ± 0.363	0.539 ± 0.358
	Ranking	3.70	**2.30**	3.80	2.90	3.50	3.40
	MAUC	0.877 ± 0.098	**0.879** ± 0.101	0.870 ± 0.096	0.869 ± 0.101	0.869 ± 0.101	0.866 ± 0.101
	Ranking	2.5	**1.8**	4	4	3.6	4.5
		None	COM	ROS	SMOTE	ADASYN	B-SM
KNN	F1	0.734 ± 0.176	**0.767** ± 0.187	0.753 ± 0.226	0.742 ± 0.239	0.730 ± 0.235	0.748 ± 0.214
	Ranking	2.6	**2.5**	3.2	3.5	4.3	4.3
	MG	0.351 ± 0.417	0.508 ± 0.326	0.512 ± 0. 367	0.563 ± 0 0.357	0.500 ± 0.385	**0.568** ± 0.3 − 20
	Ranking	5.1	3.7	2.6	**2.5**	3.2	2.7
	MAUC	**0.918** ± 0.089	**0.918** ± 0.096	0.899 ± 0.115	0.892 ± 0.130	0.892 ± 0.126	0.895 ± 0.120
	Ranking	2.1	**1.7**	3.2	4.1	4.9	3.8
		None	COM	ROS	SMOTE	ADASYN	B-SM
MLP	F1	0.777 ± 0.197	**0.803** ± 0.187	0.800 ± 0.211	0.793 ± 0.218	0.767 ± 0.218	0.782 ± 0.217
	Ranking	3.1	3.4	**2.4**	3.1	4	4.2
	MG	0.297 ± 0.416	0.629 ± 0.349	0.624 ± 0.388	**0 0.6 33** ± 0.374	0.605 ± 0.385	0.631 ± 0.341
	Ranking	4.9	**2.2**	2.6	2.8	3.3	3
	MAUC	0.919 ± 0.085	**0.946** ± 0.057	0.938 ± 0.072	0.936 ± 0.073	0.923 ± 0.081	0.937 ± 0.074
	Ranking	2.8	**1.8**	2.6	3.5	4.1	3.6
		None	COM	ROS	SMOTE	ADASYN	B-SM
The total average	F1	0.772 ± 0.012	**0.793** ± 0.018	0.783 ± 0.022	0.776 ± 0.024	0.763 ± 0.026	0.773 ± 0.018
	Ranking	2.733	**2.633**	3.067	3.533	3.967	4.367
	MG	0.382 ± 0.104	0.561 ± 0.063	0.545 ± 0.068	**0.574** ± 0040	0.53 1 ± 0 0.064	0.580 ± 0.047
	Ranking	4.567	**2.733**	3.000	**2.733**	3.333	3.033
	MAUC	0.905 ± 0.024	**0.914** ± 0.033	0.902 ± 0.034	0.899 ± 0.034	0.895 ± 0.027	0.899 ± 0.036
	Ranking	2.467	**1.767**	3.267	3.867	4.200	3.967

4.2.2 Mean Ranking

The ranking results in Table 2 show that the COM oversampling method outperforms all other methods for any evaluation metric when DT classifier is used. For the KNN and MLP classifiers, the COM oversampling method has the best results in two of three evaluation metrics, similar to the average value scenario. For the convenience of comparison, the total average of each resampling method for the three classifiers is also shown in Table 2, from which it can be seen that, compared with the other resampling methods with the MG evaluation metric, the COM method does not show a decisive advantage, and its overall effect with SMOTE is similar. However, with the F1 and MAUC metrics, COM has an absolute advantage in value and ranking.

4.3 Statistical Test of Experimental Results

4.3.1 Friedman Test

To further verify the significant statistical difference in the ranking among resampling methods, the non-parametric statistical method, Friedman test, which mainly tests the overall differences through the rank sum, was applied to the ordering of the resampling methods. The null hypothesis is that there is no significant difference in the average ranking of the resampling methods. The results are shown in Table 3, showing that the null hypothesis is rejected per the three evaluation metrics for the DT and KNN classifiers. That is, the average ranking of various resampling methods is significantly different. When the MLP classifier is used, the Friedman test shows statistical insignificance for the F1 metric. In contrast, test results for the other two metrics show a significant difference in average ranking.

4.3.2 Multiple Comparisons

Based on the statistical significance of the Friedman test results, the mean ranking of resampling methods was tested using multiple comparisons method. Since this research aimed to investigate the performance of the proposed COM oversampling method in solving intraclass imbalance, only the COM method was used as the control in comparing methods. The comparison results are listed in Table 4. In multiple comparisons, the average ranking difference between methods was compared with the critical value determined by the number of methods and instances for different combinations of classifiers and evaluation metrics. The bold font in the table indicates that the results of multiple comparisons are statistically significant. That is, the COM oversampling method is superior to the compared methods.

Table 3 Results for Friedman's test

DT		KNN		MLP	
Metric	χ^2-value	Metric	χ^2-value	Metric	χ^2-value
F1	13.392**	F1	9.478*	F1	5.938
MG	9.414*	MG	14.701**	MG	12.974**
MAUC	14.637**	MAUC	20.196***	MAUC	9.481*

"***," "**," and "*" in the table respectively indicate that the test results are statistically significant when the significance level is 0.01, 0.05, and 0.1

Table 4 Results for multiple comparisons

Metric	None	ROS	SMOTE	ADASYN	B-SM
DT					
F1	0.5	1.6	**2**	1.6	**2.6**
MG	1.4	1.5	0.6	1.2	1.1
MAUC	0.7	**2.2**	**2.2**	**1.8**	**2.7**
KNN					
F1	0.1	0.7	1.0	**1.8**	**1.8**
MG	1.4	1.1	1.2	0.5	1.0
MAUC	0.4	1.5	**2.4**	**3.2**	**2.1**
MLP					
MG	**2.7**	0.4	0.6	1.1	0.8
MAUC	1.0	0.8	**1.7**	**2.3**	**1.8**

For the DT and KNN classifiers, Table 4 shows that the COM oversampling method is statistically significant for the F1 and MAUC evaluation metrics. In the experiment on the MLP classifier, the average F1 of the COM oversampling method is the highest. However, the performance of the Friedman test results is not significant with the F1 evaluation metric, so only the other two metrics are used for multiple comparisons. The results show that the average ranking of the COM method is significantly better than other methods with the MAUC metric.

5 Simulation Experiment of COM Stability

In multi-class imbalance learning, due to the diversity of the number of classes and IR, the complete virtual data is restrictive in reflecting the actual problem, while simulation experiments based on real datasets can better reflect the applicability of the algorithm.

The effectiveness of COM algorithm has been verified in the above research. Next, its stability will be further analyzed through the simulation experiment of noise interference to learning results. Since most of the noise in practice is Gaussian white noise, we add the Gaussian white noise to the multi-class imbalance data in Table 1 according to the signal-to-noise ratio (SNR). Where the SNR indicates the ratio of signal power to noise power, and the smaller the SNR, the more noise in the data, otherwise the opposite.

In order to analyze the stability of COM algorithm, the experiment still use DT, KNN, and MLP classifiers to compare several resampling methods based on different evaluation criteria, where the experimental setup is similar to that in 4.1.3. The minority classes in the new datasets after the addition of noise were oversampled, and the experiments were conducted using a fivefold cross-validation method, where the average of the five trials was used to measure the classification effectiveness of the classifier.

Since the smaller SNR represents the more serious noise in the data, in order to make a reasonable evaluation of the stability of COM algorithm, the SNR is taken to be 15 in the experiment. That is, ten virtual multi-class imbalanced datasets can be obtained by adding Gaussian white noise with SNR of 15 to the data in Table 1. The experiments utilize oversampling algorithms to balance the virtual datasets and then compare the classification effectiveness of the three classifiers.

The experimental results are shown in Fig. 2. It can be seen that COM algorithm has obvious advantage when utilizing DT for classification, indicating that its data balancing effect is significantly better than other resampling methods, while in the case of KNN and

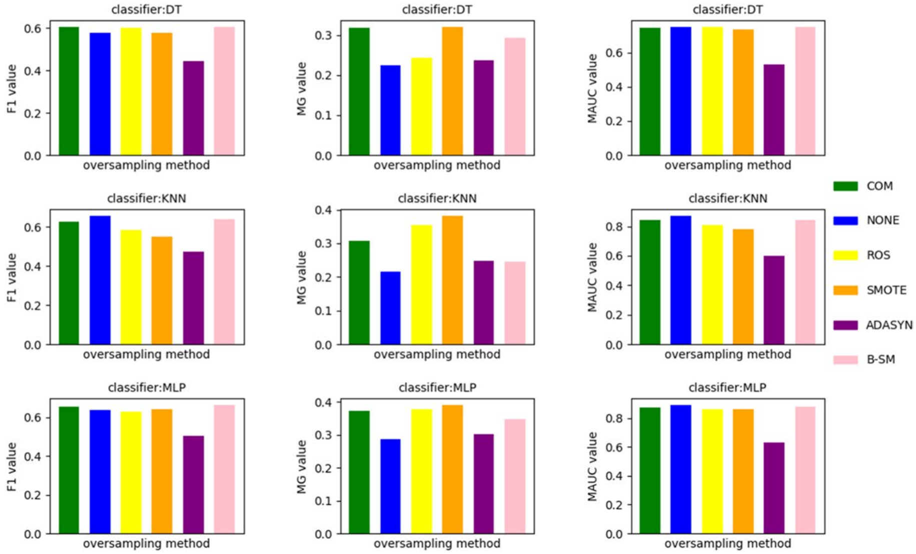


Fig. 2 Average performance results of each oversampling method versus the virtual datasets

MLP, the COM algorithm demonstrates a clear advantage with two evaluation criteria. The results of this experiment show that COM algorithm demonstrates relative stability in the experiment; i.e., the noise does not have a significant effect on its balancing effect. It is worth mentioning that, in terms of the evaluation criterion MAUC, the COM algorithm's ability to categorize any two classes does not seem to be outstanding after adding noise. This is mainly due to the small setting of SNR in the experiment.

6 Conclusions

This paper has proposed a COM oversampling algorithm, which focuses on solving the intra-class imbalance and sample overgeneralization for multi-class imbalance problems. With our proposed method, the minority class instances are first locally clustered, and then, the sampling weights of clusters are set according to the distribution and density of clusters. After this, the oversampling method was performed on the extracted clusters, making full use of the structural characteristics of the minority class instances. The effectiveness of COM was verified on multi-class imbalanced datasets through comparing to multiple oversampling methods. Experiment results demonstrate that COM can alleviate the influence of intra-class imbalance and overgeneralization significantly and can improve classifiers' average classification ability for any two types of instances in multi-class imbalance problems; our proposed method outperforms other compared methods in terms of F1, MAUC, and MG.

Although the training of COM is slightly more complex, it is effective in learning the distribution of minority class and synthesizing minority class instances. In future work, we will continue to explore imbalance learning methods based on data distribution and generalize COM to deal with the multi-class imbalanced data with high dimensions. In addition, raising dimensions for imbalanced data to improve the learning performance is a feasible idea, and we will try to combine COM with raising dimension methods to obtain better performance for the multi-class imbalance learning.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00357-024-09491-1.

Funding This work was supported by Shanxi Philosophy and Social Science Foundation, China (No. 2022YJ075), and Social and Economic Statistics Foundation of Shanxi Province, China (No. KY[2022]274).

Data Availability The data used in this study are available from the UCI machine learning repository (http://archive.ics.uci.edu/ml/index.php) and the KEEL database repository (http://www.keel.es/).

Declarations

Conflict of Interest The authors declare no competing interests.

References

Abdi, L., & Hashemi, S. (2015). To combat multi-class imbalanced problems by means of over-sampling techniques. *Soft Computing, 19*(12), 3369–3385.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*(1), 341–378.

Dong, M., Liu, M., & Jing, C. (2022). One-against-all-based Hellinger distance decision tree for multiclass imbalanced learning. *Front Inform Technol Electron Eng, 23*, 278–290.

Fernandez-navarro, F., Hervásmartínez, C., & Gutiérrez, P. A. (2011). A dynamic over-sampling procedure based on sensitivity for multi-class problems. *Pattern Recognition, 44*(8), 1821–1833.

Guo, H., Li, Y., Li, Y., & Li, J. (2016). BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification. *Engineering Applications of Artificial Intelligence, 49*, 176–193.

García, V., Sánchez, J. S., & Mollineda, R. A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems, 25*(1), 13–21.

H. He, Y. Bai, E. A. Garcia, and S. Li, (2008) "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," 2008 IEEE International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence, pp. 1322–1328.

H. Hartono, E. Ongko, "Combining hybrid approach redefinition-multiclass imbalance (HAR-MI) and hybrid sampling in handling multi-class imbalance and overlapping," JOIV: International Journal on Informatics Visualization, vol. 5, no. 1, pp. 22–26, 2021.

Hartono, H., Ongko, E., & Risyani, Y. (2021). Combining feature selection and hybrid approach redefinition in handling class imbalance and overlapping for multi-class imbalanced. *Indonesian Journal of Electrical Engineering and Computer Science, 21*(3), 1513–1522.

Han, H., Wang, W., & Mao, B. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. *Lecture Notes in Computer Science, 3644*(5), 878–887.

Kang, S., Cho, S., & Kang, P. (2015). Constructing a multi-class classifier using one-against-one approach with different binary classifiers. *Neurocomputing, 149*, 677–682.

Krawczyk, B., Koziarski, M., & Wozniak, M. (2020). Radial-based oversampling for multiclass imbalanced data classification. *IEEE Transactions on Neural Networks and Learning Systems, 31*(8), 2818–2831.

Liu, M., Dong, M., & Jing, C. (2021). A modified real-value negative selection detector-based oversampling approach for multiclass imbalance problems. *Information Sciences, 556*, 160–176.

Li, Q., Song, Y., Zhang, J., & Sheng, V. S. (2020). Multiclass imbalanced learning with one-versus-one decomposition and spectral clustering. *Expert Systems with Application, 147*, 1–14.

Lin, M., Tang, K., & Yao, X. (2013). Dynamic sampling approach to training neural networks for multiclass imbalance classification. *IEEE Transactions on Neural Networks & Learning Systems, 24*(4), 647–660.

Napierala, K., & Stefanowski, J. (2016). Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems, 46*(3), 563–597.

Rekha, G., & Eddy, V. (2021). DDCO - Diversified data characteristic-based oversampling for imbalance classification problems. *Journal of Information Science and Engineering, 37*(5), 1011–1023.

S. Shaikh, C. Liu, M. Rasheed, and S. Rizwan, "Wide research on software defect model with overgeneralization problems," International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), pp.1–6, 2019.

Saez, J., Luengo, J., & Stefanowski, J. (2015). Addressing the noisy and borderline examples problem in classification with imbalanced datasets via a class noise filtering method-based re-sampling technique. *Information Sciences, 291*, 184–203.

Tang, B., & He, H. B. (2017). GIR-based ensemble sampling approaches for imbalanced learning. *Pattern Recognition, 71*, 306–319.

Wu, J., Xiong, H., & Chen, J. (2010). COG: Local decomposition for rare class analysis. *Data Mining and Knowledge Discovery, 20*(2), 191–220.

S. Wang, X. Yao, “Multiclass imbalance problems: Analysis and potential solutions,” IEEE Trans. Syst, Man Cybern. B, Cybern, vol. 42, no. 4, pp. 1119–1130, 2012.

Wang, Q., Zhou, Y., Cao, Z., & Zhang, W. (2022). M2SPL: Generative multiview features with adaptive meta-self-paced sampling for class-imbalance learning. *Expert Systems with Applications, 189*, 115999.

Zhou, Z. H., & Liu, X. Y. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge & Data Engineering, 18*(1), 63–77.

Zhu, T., Lin, Y., & Liu, Y. (2017). Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognition, 72*, 327–340.

Zhu, T., Lin, Y., Liu, Y., Zhang, W., & Zhang, J. (2019). Minority oversampling for imbalanced ordinal regression. *Knowledge-Based Systems, 166*, 140–155.

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