Knowledge-Based Systems 279 (2023) 110938

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Label correlation guided borderline oversampling for imbalanced multi-label data learning

Kai Zhang [a], Zhaoyang Mao [a], Peng Cao [a,b,c,*], Wei Liang [a], Jinzhu Yang [a,b,c], Weiping Li [d], Osmar R. Zaiane [e]

[a] Computer Science and Engineering, Northeastern University, Shenyang, China
[b] Key Laboratory of Intelligent Computing in Medical Image of Ministry of Education, Northeastern University, Shenyang, China
[c] National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Shenyang, China
[d] School of Software and Microelectronics, Peking University, China
[e] Alberta Machine Intelligence Institute, University of Alberta, Edmonton, Alberta, Canada

ARTICLE INFO

Article history:
Received 3 November 2022
Received in revised form 21 July 2023
Accepted 25 August 2023
Available online 28 August 2023

Keywords:
Class imbalance
Multi-label data classification
Oversampling
Label correlation
Critical boundary regions

ABSTRACT

Multi-label data classification has received much attention due to its wide range of application domains. Unfortunately, a class imbalance problem often occurs in multi-label datasets, causing challenges for classification algorithms. Oversampling is one of the most important approaches, as it generates minority label instances to balance the class distribution. However, existing oversampling methods ignore existing label correlations, resulting in the generation of inappropriate synthetic minority samples and making multi-label data classification tasks harder. In this work, we propose an oversampling method that considers label correlations and identifies two critical boundary regions for generating synthetic minority samples. Moreover, we propose a weighting strategy to assign weights to these instances based on their distance information. To evaluate the performance of our proposed method, we conducted experiments on sixteen public datasets. The results show that our approach outperforms the state-of-the-art approaches in terms of various assessment metrics, such as Macro F_1 and Macro AUC. The code is available at https://github.com/IntelliDAL/Multi-label/tree/main/LCOS.

1. Introduction

Traditional supervised learning addresses the problem where each object is associated with a single predefined label. However, across a wide range of real-world applications, e.g., biological data analysis, social network mining, and text categorization [1–6], one object may simultaneously contain multiple labels. Multi-label learning algorithms attempt to learn a mapping from the feature space $X \subseteq \mathbb{R}^d$ to the label space $Y \subseteq \{0, 1\}^c$, where d denotes the dimension of features and c denotes the number of labels. To handle such tasks, multi-label learning has attracted much attention in recent years [7–11]. Class imbalance among labels, as one of the greatest challenges in learning from multi-label data, has not been fully considered by most existing multi-label learning approaches.

In imbalanced learning, the sampling methods aim to balance the distribution between the majority and the minority classes [12–16]. Although it is difficult to model the actual class distribution, it is observed that classifiers learn better from a balanced distribution than from an imbalanced one [17]. One flexible and efficient solution to address the imbalance problem is to employ sampling methods before training a multi-label learning model [18–20]. However, the imbalance issue makes it more complex in multi-label classification settings. Traditional undersampling or oversampling does not easily generalize to the multi-label domain due to the complicated imbalance between labels. The reasons are as follows: (1) there exist multiple minority classes or multiple majority classes, which are more difficult to tackle; (2) the labels tend to co-occur, resulting in the joint appearance of minority and majority labels in the same instances. How to solve the co-occurrence of multiple labels with varying frequencies for the same training example is critical [8]. For example, as shown in Fig. 1, multiple minority labels (multi-minority cases) or multiple majority labels (multi-majority cases) exist at the same time. Moreover, many inherent correlations among the labels with different strengths occur. Both introduce more challenges for oversampling algorithms.

In two-class classification task modeling, the relationships between classes are relatively simple. In a multi-label task, the label relationships are definitely more complex and challenging. Although existing multi-label sampling methods balance the global imbalance of multi-label datasets, they neglect the fact that the

* Corresponding author at: Computer Science and Engineering, Northeastern University, Shenyang, China.
E-mail address: caopeng@cse.neu.edu.cn (P. Cao).

https://doi.org/10.1016/j.knosys.2023.110938
0950-7051/

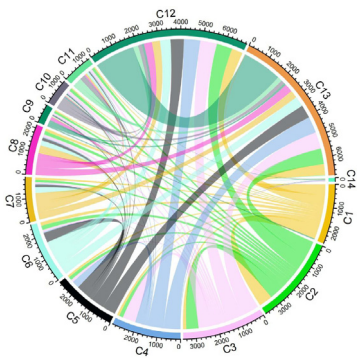


Fig. 1. The concurrence among the labels in the yeast dataset, which exhibits multi-label imbalance. Each arc represents a label, and the width of the bands connecting arcs denotes the number of samples with a pair of labels appearing together.

labels are correlated; especially when some minority labels are correlated with other minority and majority labels with different strengths, the label correlations may provide helpful extra information. In this work, we ask the following question: are label correlations useful for oversampling minority labels in domains that exhibit multiple labels? Our research shows that the answer is yes, and that the key is how to learn the label correlations and how to leverage the label correlations to guide the oversampling for generating exact minority label data. To this end, we propose a label correlation guided oversampling method for multi-label data (named LCOS), which aims to leverage label correlations to alleviate the problems of imbalanced learning in multi-label data and generate the appropriate synthetic minority label instances from the perspective of label correlations. The essence of the proposed method is three-fold: (1) learning the inherent label correlations; (2) selecting an appropriate subset of each minority label sample according to the learned correlation matrix; and (3) assigning weights to the selected samples according to their importance in the data.

The main challenge for oversampling multi-label data lies in identifying the critical regions for generating synthetic samples in the setting of a multi-label task, with the goal of developing a good prediction model. In our work, label correlations are leveraged to guide the oversampling for the multi-label data. More specifically, we model label correlations through sequence learning with the memory mechanism of the Recurrent Neural Network (RNN) layer. In our study, we employ a simple RNN (SRN) [21] to better capture the highly non-linear label correlations. Each iteration in the SRN layer produces an updated prediction considering the label correlations. We formulate label correlation learning as a sequence prediction problem. Given the learned correlations, the specific regions of the minority label to be oversampled are determined. Much of the literature, such as Borderline-SMOTE [22] and MWMOTE [23], has already shown that boundary samples make the greatest contribution to classification performance. In our work, we define two boundaries, outer-boundary and inter-boundary as seeds. At the same time, in addition to the between-class imbalanced distribution, there also exists within-class imbalance, which means that given a pair of labels p and q, there is an imbalance between the region where instances are associated with both labels p and q, and the region where instances are associated with label p without label q. To address this problem, the inner-boundary samples are also identified as candidate seeds. From the subset of candidate seeds, we further propose a weighing scheme for determining the final seeds according to their importance in the data. Finally, different numbers of instances are generated for different minority seeds by interpolation. Instead of performing global oversampling such as MLROS and MLSMOTE [24], our oversampling is a local oversampling method. This locality is reflected in two aspects: only the labels satisfying our assumption are considered rather than all the labels, and only local boundary regions are oversampled rather than the whole region of the minority label. Comprehensive experiments on sixteen benchmark datasets show that LCOS achieves competitive performance compared to state-of-the-art algorithms, in terms of Macro F_1, Macro AUC, Hamming Loss and Ranking Loss.

2. Related work

Recently, the imbalance problem in multi-label learning has been widely studied due to its wide application. In this section, we briefly review the research on multi-label learning, label correlation modeling and multi-label sampling methods.

2.1. Multi-label learning

In multi-label learning, each instance is associated with multiple class labels simultaneously [25]. Formally, let $\mathcal{X} \in \mathbb{R}^d$ denote the d-dimensional input space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_c\}$ denote the c predefined labels. The task of multi-label classification is to learn a function $h : \mathcal{X} \rightarrow \mathcal{Y}$ from the multi-label training set $\mathcal{D} = \{(x_i, Y_i)|1 \leqslant i \leqslant N\}$. Here, $x_i \in \mathcal{X}$ is a d-dimensional instance and $Y_i \subseteq \mathcal{Y}$ is the set of labels associated with x_i. N denotes the number of instances. For any unseen instance x_j, $h(\cdot)$ can assign a set of relevant labels for x_j.

Most multi-label learning methods mainly exploit the correlations among labels to improve the prediction performance [26]. According to the order of the label correlations, these methods can be divided into three categories: first-order, second-order, and high-order. Multi-Label k-Nearest Neighbor (ML KNN) [27], Weighted MLKNN [28], Relative Discernibility Pair Matrix (RDPM) [29] and Binary Relevance (BR) [30] are first-order methods that completely ignore label correlations and treat each class label independently. To solve this limitation, the pairwise correlations between class labels are considered in several works. For example, Calibrated Label Ranking (CLR) [31] exploits pairwise label correlations by transforming the multi-label learning problem into a pairwise label ranking problem. Nan et al. [32] designed a method to exploit the local positive and negative pairwise label correlations. Although the second-order methods are more effective in exploiting the label correlations, the relationships of labels are complex in real-world scenarios and high-order correlations should be taken into account. The previous studies [33–35] have revealed that the higher-order strategy has a stronger ability for modeling the label correlations than the first-order and second-order strategies. For example, Zhang et al. [34] developed an approach to enrich the labeling information by transforming structural information modeled by sparse reconstructions in the feature space. Lin et al. [35] introduced a multi-label learning method based on linear regression and considered the label correlations based on a fuzzy similarity relationship within the label set. The high-order approach can capture more important correlations and exploit more vital information hidden in labels. Therefore, our work focuses on the higher-order strategy.

2.2. Exploiting label correlation

How to exploit the label correlations for improving the classification performance is important in multi-label classification. A common approach is to calculate the co-occurrence frequency of labels in the label space. For example, Chou et al. [36] proposed to capture the relations between labels by calculating a

co-occurrence statistical matrix. Li et al. [37] developed a multi-label method taking advantage of label correlations based on the assumption that if two labels are strongly related, they have similar outputs. Lin et al. [38] proposed two types of mutual information between labels to capture the label correlations under a binary distribution. However, most of these methods obtain the second-order label correlations by calculating the co-occurrence frequency or mutual information, failing to fully exploit the label information. To better model the inherent label correlations, some graph-based methods are proposed to model the high-order label correlations. For example, Huang et al. [39] designed an approach to model the inherent label correlations by learning an embedding matrix from a predefined label correlation graph via graph embedding. Chen et al. [40] introduced a label-aware graph representation learning to acquire more reliable and discriminative graph label representation and graph feature representation. Chen et al. [41] proposed to take advantage of graph convolutional networks to capture the high-order label correlations. However, most graph-based methods have a common limitation: the graph structure for modeling the label correlations is predefined by simply estimating their co-occurrence patterns, which hinders the capability of modeling the appropriate label correlations. To address the above problem, we regard the high-order label correlation modeling as a sequence learning paradigm via the internal memory characteristic in simple RNN (SRN) [21].

2.3. Multi-label sampling methods

Multi-label sampling approaches aim to correct skewed class distributions by removing or adding examples, which is independent of multi-label classification algorithms [8]. Particularly, these studies can be further divided into two categories: undersampling methods and oversampling methods. Multi-label undersampling methods attempt to remove instances associated with the majority label by random and heuristic schemes. For example, two popular multi-label undersampling methods, Label Powerset Random Undersampling (LPRUS) and Multi-Label Random Undersampling (MLRUS) [42], were proposed to alleviate the imbalanced distribution. The former converts multi-label into multi-class classification tasks by treating each label combination (label set) as a class, then randomly removes the instances associated with the most frequent label set. By contrast, the latter instead considers the frequency of individual labels rather than the whole label set. The major limitation of the random undersampling methods is that significant information may be lost. To alleviate this problem, some heuristic undersampling methods were proposed. For example, Multi-Label Tomek Link (MLTL) [43] was designed as a heuristic undersampling by employing Tomek Link for data cleaning in the majority labels. MultiLabel edited Nearest Neighbor (MLeNN) [44] first excludes all the samples with the minority labels and then removes the remaining samples that are significantly different from their neighbor label sets. Contrary to the multi-label undersampling methods, multi-label oversampling methods attempt to generate instances associated with minority labels to balance label distribution by random or heuristic scheme. Similar to LPRUS, Label Powerset Random Oversampling (LPROS) randomly replicates instances associated with the least frequent labels. However, LPROS is prone to overfitting since it often involves replicating minority class samples. To solve this problem, Multilabel Synthetic Minority Over-sampling Technique (MLSMOTE) [24] was proposed to generate new synthetic instances and the associated label set from the selected instance and its neighbors. However, MLSMOTE is prone to produce noise since it ignores the neighbors' distribution during the oversampling process. To alleviate this issue, Multi-Label Synthetic Oversampling based on Local label imbalance (MLSOL) [45] was proposed to select instances with a large degree of local imbalance for oversampling by taking into account the local imbalance.

Our work belongs to the heuristic oversampling method. Although previous works improve the multi-label imbalanced data learning to some extent, they fail to consider the effect of label correlations in oversampling.

3. Definition and assumption

3.1. Definition

To balance the class distribution, we need to oversample the minority label instances. However, not all the minority label instances contribute to the classification and therefore they are not all required in the oversampling. Therefore, we need to select the local regions that are significant for the classification performance, especially when the labels exhibit complicated correlations. Because the instances on the boundary and the ones nearby are more likely to be misclassified than the ones far from the boundary, they are more important for classification. To this end, our method focuses on oversampling the instances from the boundaries of the minority class labels. Furthermore, we categorize the boundaries into outer-boundary and inner-boundary given a pair of labels y_p and y_q. They are defined as follows:

Definition 1. For a pair of labels y_p and y_q, $R_p = \{(x_i, Y_i)|y_p \in Y_i, 1 \leqslant i \leqslant N\}$ denotes the set of instances associated with y_p and $R_{\bar{p}q} = \{(x_i, Y_i)|y_p \notin Y_i, y_q \in Y_i, 1 \leqslant i \leqslant N\}$ denotes the set of instances associated with y_q but not associated with y_p. The **outer-boundary** B^O_{pq} of y_p is the boundary between R_p and $R_{\bar{p}q}$. More specifically, for each $x_i \in R_{\bar{p}q}$, let $NN^O_{pq}(x_i)$ denote the nearest k neighbors of x_i in R_p. Thus, B^O_{pq} is the union of all $NN^O_{pq}(x_i)$, i.e., $B^O_{pq} = \cup_{x_i \in R_{\bar{p}q}} NN^O_{pq}(x_i)$.

Definition 2. For a pair of labels y_p and y_q, $R_{p\bar{q}} = \{(x_i, Y_i)|y_p \in Y_i, y_q \notin Y_i, 1 \leqslant i \leqslant N\}$ denotes the set of instances associated with y_p but not associated with y_q. $R_{pq} = \{(x_i, Y_i)|y_p \in Y_i, y_q \in Y_i, 1 \leqslant i \leqslant N\}$ denotes the set of instances associated with both y_p and y_q. The **inner-boundary** B^I_{pq} of y_p is the boundary between $R_{p\bar{q}}$ and R_{pq}. More specifically, for each $x_i \in R_{pq}$, let $NN^I_{pq}(x_i)$ denote the nearest k neighbors of x_i in $R_{p\bar{q}}$. Thus, B^I_{pq} is the union of all $NN^I_{pq}(x_i)$, i.e., $B^I_{pq} = \cup_{x_i \in R_{pq}} NN^I_{pq}(x_i)$.

Both the outer-boundary and inner-boundary are illustrated in Fig. 2. If we only oversample the outer-boundary B^O_{pq}, it may lead to within-class imbalance for y_p, which indicates that $R_{p\bar{q}}$ and R_{pq} are not balanced. To avoid this, it is necessary to oversample B^I_{pq}. A minority label instance is identified as an instance of B^O_{pq} if it is located in R_p and is one of the k-nearest neighbors of an instance in $R_{\bar{p}q}$. Similarly, a minority label instance is identified as an instance of B^I_{pq} if it is located in $R_{p\bar{q}}$ and is one of the k-nearest neighbors of an instance in R_{pq}.

3.2. Assumption

It is well-known that label correlations are significant for multi-label learning. Many current multi-label learning approaches attempt to incorporate label correlations to improve performance. Label correlations can provide useful additional information, especially when there are insufficient training samples for some labels. In this work, we assume that the stronger correlation between a pair of labels, the more instances exist with label co-occurrence in their outer-boundary, and vice versa. To verify our hypotheses, we carried out a pilot study to investigate the

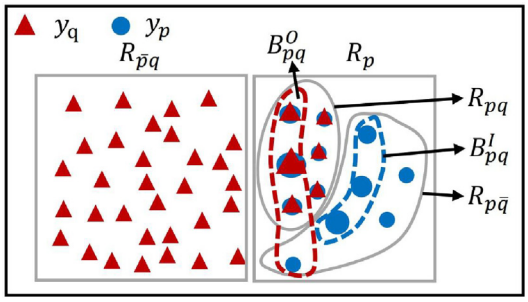


Fig. 2. The set of labels for this multi-label dataset is $\{y_p, y_q\}$. B^O_{pq} is the boundary between R_p and $R_{\_pq}$. B^I_{pq} is the boundary between R_{p_q} and R_{pq}. The size of the shape represents the selection weight.

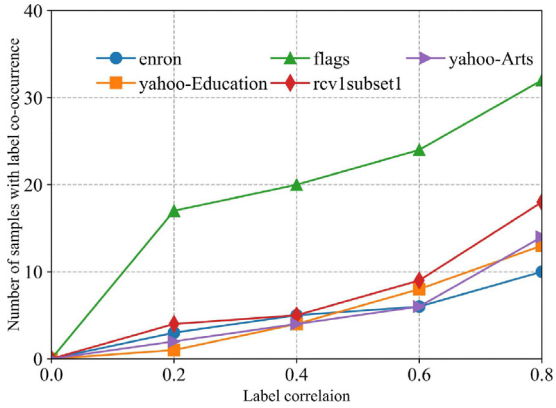


Fig. 3. The number of instances with co-occurrence labels on the outer-boundary under different levels of label correlations. The *x*-axis represents the label correlation strength. The *y*-axis represents the number of label pairs on the outer-boundary for each label correlation strength.

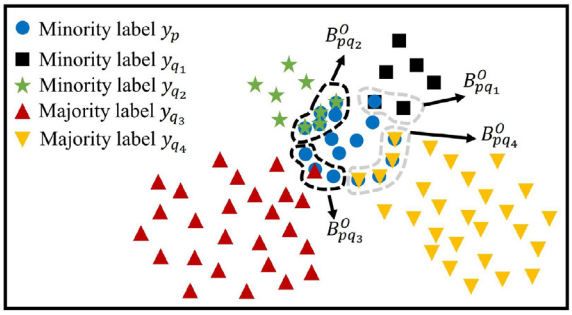


Fig. 4. A multi-label dataset with five labels. $B^O_{pq_j}$ denotes the outer-boundary between label y_p and its related label y_{q_j}.

relationship between the number of instances with co-occurrence labels and the degrees of correlation strength. From Fig. 3, it can be observed that the stronger correlation between a pair of labels, the more instances with co-occurrence labels exist in their outer-boundary. The label correlation matrix V is obtained by the sequence learning described in Section 4.1 and $V(p, q)$ represents the label correlation between labels y_p and y_q. If $V(p, q)$ is greater than a correlation threshold τ, it indicates that y_p has a strong correlation with y_q, otherwise, they are weakly correlated. Then, label correlations can guide us to choose the outer-boundary of each minority for oversampling. For a minority label, we divide its outer-boundary into four types of regions occupied by the minority and its related labels. Fig. 4 depicts the outer-boundary for four different cases of label y_p. As shown in Fig. 4, y_p is a

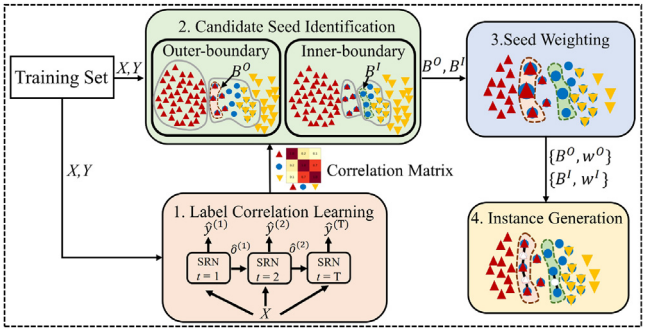


Fig. 5. The architecture of our method.

minority label and is correlated with the other four labels. Thus, the following conditions exist:

- Condition 1: y_{q_1} is a related **minority** class and $V(p, q_1) \leq \tau$. Both y_p and y_{q_1} are minority labels and relatively balanced. Therefore, the outer-boundary of $B^O_{pq_1}$ does not need to be oversampled.
- Condition 2: y_{q_2} is a related **minority** class and $V(p, q_2) > \tau$. In $B^O_{pq_2}$, there are many instances associated with both y_p and y_{q_2} due to their strong correlation. From the view of Label Powerset [46], the combination label of y_p and y_{q_2} is a meaningful label, thus $B^O_{pq_2}$ is desirable to be enhanced by oversampling.
- Condition 3: y_{q_3} is a related **majority** class and $V(p, q_3) \leq \tau$. In $B^O_{pq_3}$, there are fewer instances associated with both y_p and y_{q_2} due to their weak correlation. Oversampling in $B^O_{pq_2}$ can effectively increase minority label instances to alleviate the class imbalance.
- Condition 4: y_{q_4} is a related **majority** class and $V(p, q_4) > \tau$. In $B^O_{pq_4}$, these instances are associated with y_p and y_{q_4} simultaneously. Oversampling y_p in $B^O_{pq_4}$ could inevitably increase the number of co-occurrences of labels y_p and y_{q_4}, failing to alleviate the class imbalanced distribution.

In summary, only Condition 2 and Condition 3 necessitate oversampling.

4. Label correlation guided borderline oversampling

The structure of our proposed method LCOS is shown in Fig. 5. As mentioned before, it consists of four components: (*i*) Label correlation learning for capturing the label correlation. (*ii*) Candidate seed identification for determining the instances that are located on the outer-boundary and inner-boundary. (*iii*) Seed weighting for assigning the proper weight to the candidate seeds and then selecting the final seeds based on their weights. (*iv*) Instance generation for generating new instances from the final seed instances.

4.1. Label correlation learning

Recurrent Neural Network (RNN) is a class of neural network models commonly used to solve sequence prediction problems. To capture the correlations among labels, we use the simple RNN (SRN) model [21], which is a basic variation of RNN. It can iteratively learn the label correlations through its memory structure. Formally, label correlation learning is formulated as a sequence prediction problem as follows:

$$\hat{y}^{(1)} = \sigma\left(UX\right), \tag{1}$$

$$\hat{y}^{(t)} = \sigma\left(UX + V\hat{y}^{(t-1)}\right), \tag{2}$$

where $\hat{y}^{(t)}$ denotes the output vector, $U \in \mathbb{R}^{c\times d}$ is used to transform the feature vector into the output space, d is the dimension of data, c is the number of labels and $V \in \mathbb{R}^{c\times c}$ is used to transform the output of the previous iteration into the same output space as the output of U.

We set the number of iterations of the SRN layer to be T. In the first iteration, the sequence learning produces a prediction based on Eq. (1) without considering other labels. From the second iteration, the sequence learning begins to exploit the output of the previous iteration to make better predictions, with the last predicted $\hat{y}^{(T)}$ as the final result. Particularly, the memory term $V\hat{y}^{(t-1)}$ in Eq. (2) serves as label correlation by taking in the previous output and transforming it to the same output space as the output of U. Therefore, the final prediction $\hat{y}^{(t)}$ is obtained through T iterations, while the label correlation V is also learned.

4.2. Candidate seed identification

The criterion for considering a label y_p to be a minority class label when $IRLbl_p > MeanIR$, where $IRLbl_p = N_{max}/N_p$ is a measure to assess the imbalance ratio of the pth label, $MeanIR = \frac{1}{c}\sum_{p=1}^{c} IRLbl_p$ is obtained by averaging the IRLbl for all labels, N_p is the number of instances belonging to y_p and $N_{max} = max(N_1, N_2, \ldots, N_c)$ is the maximum number of instances contained in the label set. Let Y^{min} and Y^{maj} denote the set of minority labels and the set of majority labels, respectively. For each $y_p \in Y^{min}$, it is noted that $R_p = R_{pq_1} \cup R_{pq_2} \cup \ldots R_{pq_m} \cup R_p^s$, where m denotes the number of associated labels with y_p and R_p^s denotes the set of instances that are only associated with y_p. Therefore, $B_p^O = B_p^O \cup B_{pq}^O (1 \leqslant q \leqslant c, q \neq p, (y_q \in Y^{min}, V(p,q) > \tau)$ or $(y_q \in Y^{maj}, V(p,q) \leq \tau))$. We define L^p as the set of labels that are correlated with the minority label y_p. $B_p^I = \cup_{y_q \in L^p} B_{pq}^I$ represents the inner-boundaries of y_p with other co-occurring labels. Therefore, the candidate seeds consist of the outer-boundaries $B^O = \cup_{y_p \in Y^{min}} B_p^O$ and the inner-boundaries $B^I = \cup_{y_p \in Y^{min}} B_p^I$.

4.3. Seed weighting

Although the instances on the boundary are more important, the information provided by these instances is still different. Therefore, we propose a weighting scheme to assign different weights for each instance according to a distance factor.

Step 1 Outer-boundary Instances Weighting For each $x_i \in B_p^O$, the farther distance from the same label instances or the closer distance from other label instances, the more important x_i is. Let S_i^p be the set of k-nearest neighbors of x_i in R_p and $S_i^{\overline{p}}$ be the set of k-nearest neighbors of x_i in $R_{\overline{p}}$. Therefore, the distance factor, denoted by $w_p^O(x_i)$, is defined as

$$w_p^O(x_i) = {dist2(x_i, S_i^p)}/{dist2(x_i, S_i^{\overline{p}})}, \tag{3}$$

where

$$dist2(x_i, S_i^p) = \sum_{x_j \in S_i^p} dist1(x_i, x_j) \Big/ \left|S_i^p\right|. \tag{4}$$

The function $dist2(x, S)$ is the distance from a point x to a sample subset S, and the function $dist1(x_i, x_j)$ is the Euclidean distance between two points x_i and x_j. $dist2(x_i, S_i^{\overline{p}})$ is calculated in Eq. (4). It represents the distance from x_i to $S_i^{\overline{p}}$. Because an instance may be associated with multiple minority labels, let $Y_i^{min} = Y_i \cap Y^{min}$ be the set of minority labels associated with x_i. Y^{min} is the set of minority labels. Thus, $w^O(x_i) = max(\{w_p^O(x_i) | y_p \in Y_i^{min}\})$. Then, it is normalized as $\hat{w}^O(x_i) = \frac{w^O(x_i)}{\sum_{i=1}^{|B^O|} w^O(x_i)}$.

Step 2 Inner-boundary Instances Weighting For $x_i \in B_p^I$, let x_{oi} be the opposite of x_i, which is the nearest neighbor of x_i in the instances of labels correlated with y_p. Let $S_i^{L_p}$ be the set of k-nearest neighbors of x_{oi} in R_{L_p}, where $R_{L_p} = \cup_{g \in L_p} R_g$. The farther the distance of x_i from its same label instances or the closer the distance of x_{oi} from its same label instances, the more important x_i is. The weighting strategy is similar to outer-boundary instance weighting. Therefore, $w_p^I(x_i)$ is defined as:

$$w_p^I(x_i) = {dist2(x_i, S_i^p)}/{dist2(x_{oi}, S_i^{L_p})}. \tag{5}$$

It is calculated in the same way as Eqs. (3) and (4). L^p represents the set of labels correlated with y_p. To avoid excessive oversampling on the inner-boundary, $w^I(x_i) = min(\{w_p^I(x_i) | y_p \in Y_i^{min}\})$. Then, it is normalized as $\hat{w}^I(x_i) = \frac{w^I(x_i)}{\sum_{i=1}^{|B^I|} w^I(x_i)}$. We further set a threshold β for $\hat{w}^I(x_i)$. If $\hat{w}^I(x_i)$ is less than β, it is set to 0.

Step 3 Seed Instance Selection $\hat{w}^O$ and $\hat{w}^I$ represent the weights of the instances on the outer-boundaries and the inner-boundaries, respectively. The number of synthetic instances for B^O and B^I are defined as $G^O = \left|B^O\right| \times p$ and $G^I = \left|\hat{w}^I > 0\right|$, respectively. $\left|\hat{w}^I > 0\right|$ is the number of instances whose weight is greater than 0 and p is the sampling ratio of B^O. A candidate seed instance is selected using the roulette algorithm [47] based on its weight. A candidate seed instance is selected according to the weights of the instances. Candidate instances that carry more information will be oversampled more times than those containing less information, thereby improving the quality of the newly generated minority instances.

4.4. Instance generation

The next issue after selecting seed instances is how to generate new instances according to the selected seed instances. Given a seed instance, a new instance is generated based on the seed instance and the reference instance through linear interpolation. Note that the selected reference instance needs to have the same set of labels as the selected seed instance. Since that region is carefully determined, the labels can be safely assigned to be the same as the seed labels.

4.5. LCOS pseudo-code

Algorithm 1 illustrates the pseudocode of the proposed LCOS, in which the training set D is the input data and the output data is the resampled data D' including the synthetic instances generated by LCOS. Line 3 of Algorithm 1 returns the minority labels that need to be oversampled. Line 4 is used to obtain the learned label correlations. Lines 5–21 depict the process of candidate seed identification. Candidate seeds consist of the outer-boundary and inner-boundary of minority labels. Line 9 is to obtain the inner-boundary of y_p. According to the learned label correlations, the outer-boundary of a minority label is the outer-boundary formed by it and strongly correlated minority labels (Lines 10–12) or weakly correlated majority labels (Lines 14–16). Lines 20–21 are the outer-boundaries and inner-boundaries, which constitute the candidate seeds. Line 23 calculates the weight of candidate seeds and the number of newly generated samples according to Eqs. (3)–(5) in Section 4.3 Seeds Weighting. Then, we select the seeds to generate the synthetic instances based on the selective weight of seeds and the number of newly generated samples (Line 24).

Algorithm 1: LCOS

Input: Training set D (X, Y), Outer-boundary sampling ratio p, Inner-boundary sampling threshold β, Number of nearest neighbors k, Correlation threshold τ

Output: Oversampled dataset D'

```
Initialize Y^min = ∅;
Initialize B^O = B^I = ∅;
Obtain Y^min based on Y;
V = LabelCor(X, Y);                    ▷ obtain label correlation matrix
for y_p in Y^min do
    for y_q in Y do
        Obtain B^O_pq and B^I_pq according to Definition 1 and Definition 2;
        if y_p ≠ y_q then
            B^I_p = B^I_p ∪ B^I_pq;                ▷ inner-boundary of y_p
            if V(p, q) > τ and y_q ∈ Y^min then
                ▷ strongly correlated with minority labels
                B^O_p = B^O_p ∪ B^O_pq;
            end
            if V(p, q) ≤ τ and y_q ∉ Y^min then
                ▷ weakly correlated with majority labels
                B^O_p = B^O_p ∪ B^O_pq;
            end
        end
    end
    B^O = B^O ∪ B^O_p;
    B^I = B^I ∪ B^I_p;
end
Calculate ŵ^O, ŵ^I, G^O, G^I according to Eqs. (3)–(5);
Generate new instances D' based on ŵ^O, ŵ^I, G^O, G^I;
D' = D' ∪ D;
return D';
```

4.6. Complexity analysis

In our proposed LCOS algorithm, the time complexity is mainly composed of four components: (1) The complexity of learning the label correlations is $\mathcal{O}(Tc^2)$, where T is the number of iterations and c is the number of labels. In our work, we set T to 100. (2) The complexity of searching kNN is $\mathcal{O}(nd)$, where d is the dimension of data. Therefore, the complexity of candidate seed identification is $\mathcal{O}(n'nd)$, where n is the number of all samples and n' is the number of all minority class samples. (3) The complexity of the seed weighting is $\mathcal{O}(n^S nd)$, where n^S is the number of candidate seeds. (4) The complexity of instance generation is $\mathcal{O}(n^G(c+d))$, where n^G is the number of synthetic instances. In summary, the complexity of the LCOS algorithm is $\mathcal{O}(Tc^2 + n'nd + n^S nd + n^G(c+d))$. The complexity of the state-of-the-art oversampling method, MLSOL [45], is $\mathcal{O}(n^2 d + n^2 k + nkc + n^G(c+d))$. The most time-consuming steps for LCOS and MLSOL are $\mathcal{O}(n'nd)$ and $\mathcal{O}(n^2 d)$, respectively. Because $n' < n$, our LCOS is less time-consuming than MLSOL.

5. Experiment

In this section, we verify the performance of the proposed LCOS by evaluating it on real datasets and comparing it with other state-of-the-art multi-label oversampling algorithms. Firstly, Section 5.1 presents the experimental setup including datasets, evaluation metrics, classification algorithms, etc. Then, the experimental results are reported in Section 5.2. The influence of the parameters of our algorithm is extensively investigated in Section 5.3. The effectiveness of learned label correlations is verified in Section 5.4. We conduct an ablation study in Section 5.5. The influences of imbalance level and label number are discussed in Section 5.6. Finally, the effectiveness of exploiting label correlations is examined in Section 5.7.

5.1. Experimental setup

5.1.1. Datasets

To thoroughly verify the effectiveness of the proposed algorithm, sixteen multi-label datasets from different domains are employed in experiments. These datasets are commonly used in multi-label classification tasks and the detailed properties of the datasets are presented in Table 1. We can observe variations in the imbalance levels in the data sets, which are meant to represent the average imbalance rate for different levels.

To maintain uniformity, we preprocess the datasets as suggested by [10,48]. We remove labels with fewer than 20 instances. We also reduce the feature sets for large datasets namely Corel5k, bibtex, rcv1subset1, rcv1subset2, yahoo-Arts, yahoo-Education. We use a simple feature selection method to retain only the top 1% of the features sorted by the number of non-zero values, similar to [45].

5.1.2. Evaluation metrics

Under class-imbalanced scenarios, F_1 and Area Under the ROC Curve (AUC) are the most commonly used evaluation metrics [10]. F_1 combines the metrics of precision and recall, and AUC consists of the metrics of sensitivity and specificity [49]. Both of them are comprehensive evaluation metrics that provide great insights into classification performance. We compute the metric average over all labels called macro-averaging, which is a way of aggregating binary metrics in multi-class and multi-label tasks. Macro averaging is more suitable for imbalanced learning because it treats all labels equally [50]. Additionally, Hamming Loss and Ranking Loss can evaluate the percentage of misclassified labels and the average fraction of reversely ordered pairs, respectively. Therefore, we employ Macro-averaged F_1 (Macro F_1), Macro-averaged AUC (Macro AUC), Hamming Loss and Ranking Loss to measure the performance of methods. They are described as follows.

- **Macro F_1**: It is calculated from the average of F_1 values across all labels.

$$Macro\ F_1 = \frac{1}{c}\sum_{j=1}^{c} F_{1j} \tag{6}$$

$$F_{1j} = \frac{2 \times TP_j}{2 \times TP_j + FP_j + FN_j} \tag{7}$$

  c indicates the total number of labels. TP_j, TN_j, FP_j and FN_j represent the number of true positive, true negative, false-positive and false-negative test examples with respect to label y_j, respectively.

- **Macro AUC**: It is calculated from the average AUC of all labels. Let AUC_j denote the AUC score for y_j.

$$Macro\ AUC = \frac{1}{c}\sum_{j=1}^{c} AUC_j \tag{8}$$

- **Hamming Loss**: It calculates the fraction of misclassified labels, i.e., the instance associated with a wrong label or a label assigned to the instance which is not predicted.

$$Hamming\ Loss = \frac{1}{N}\sum_{i=1}^{N} \frac{|\hat{Y}_i \oplus Y_i|}{\mathrm{c}}, \tag{9}$$

  where $\oplus$ represents the XOR operation.

Table 1
The multi-label datasets used in the experiments.

Dataset	Samples	Features	Labels	Cardinality	Density	MeanIR	MaxIR	Domain
enron[a]	1702	1001	35	3.273	0.094	19.128	52.286	Text
medical[a]	978	1449	14	1.191	0.085	5.968	12.765	Text
yeast[a]	2417	103	14	4.246	0.303	7.281	54.185	Biology
scene[a]	2407	294	6	1.067	0.178	1.232	1.404	Image
birds[a]	645	260	12	1.778	0.148	2.716	4.778	Audio
flags[a]	194	19	7	3.329	0.476	2.297	5.591	Image
Corel5k[a]	5000	499	50	2.341	0.047	7.459	13.191	Image
bibtex[a]	7395	183	50	1.350	0.027	6.131	9.372	Text
rcv1subset1[a]	6000	472	50	2.594	0.052	7.042	14.520	Text
rcv1subset2[a]	6000	472	50	2.354	0.047	8.388	18.435	Text
yahoo-Arts[a]	7484	231	24	1.655	0.069	13.686	78.211	Text
yahoo-Education[a]	12 030	275	27	1.460	0.054	53.751	188.689	Text
chemistry[b]	6856	540	50	1.639	0.033	9.82	19.821	Text
chess[b]	1538	585	46	1.861	0.04	12.048	24.857	Text
cooking[b]	10 157	578	50	1.161	0.023	5.055	8.674	Text
philosophy[b]	3814	842	50	1.702	0.034	12.197	29.682	Text

("Samples" is the number of samples, "Features" is the dimensionality of features, "Labels" is the total number of labels, "Cardinality" is the average number of labels per sample, "Density" is equal to cardinality divided by labels, "MeanIR" and "MaxIR" individually denote the average and maximum imbalance ratio of a data set, and "Domain" is the domain of the datasets.)

[a] https://mulan.sourceforge.net/datasets.html.

[b] https://www.uco.es/kdis/mllresources/.

Table 2
Parameters of sampling methods.

Sampling methods	Parameters of methods
MLROS	$P = 0.5$
MLRUS	$P = 0.1$
MLSMOTE	$k = 5$
MLSOL	$P = 0.3$, $k = 5$
LCOS	$p = 5$, $k = 5$, $\tau = 0.6$, $\beta = 0.2$

- **Ranking Loss**: It calculates the fraction of reversely ordered label pairs, i.e., an irrelevant label is ranked higher than a relevant label.

$$Ranking\ Loss = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i \parallel \overline{Y_i}|}|\{(y\prime, y\prime\prime)|f(x_i, y\prime) \leqslant f(x_i, y\prime\prime), (y\prime, y\prime\prime) \in Y_i \times \overline{Y_i}\}|, \tag{10}$$

where $\overline{Y_i}$ represents the complementary set of Y_i. For Macro F_1 and Macro AUC, larger values indicate better performance, while for Hamming Loss and Ranking Loss, smaller values indicate better performance.

5.1.3. Classification algorithms

To evaluate the performance of the proposed algorithm, we compare it with the following multi-label sampling algorithms.

- **MLROS, MLRUS** [42]: MLROS is a multi-label oversampling method that randomly clones samples associated with minority label sets, and MLRUS is a multi-label undersampling algorithm that randomly deletes samples of majority label sets.
- **MLSMOTE** [24]: It generates synthetic instances associated with minority labels and takes advantage of label information in the neighborhood to label the new synthetic instance.
- **MLTL** [43]: It considers the intrinsic characteristics of multi-label classification problems and proposes a multi-label imbalanceness measure to search for samples from the majority labels. Then, the majority label instance with a large difference in the label set of its neighbors is removed with Tomek Link.
- **MLSOL** [45]: It focuses on the local distribution of labels to deal with the class imbalance problem in multi-label data. Then, it employs the local label distribution to calculate the selection weight vector of the seed instance and generates more diverse and well-labeled synthetic instances considering all informative labels.

To comprehensively compare our method with the state-of-the-art sampling methods, we used six different multi-label classification algorithms: BR [30], CC [33], MLKNN [27], CLR [31], HOMER [51] and ECC [33]. For the problem transformation algorithms, the well-known SVM algorithm is used as the base learner due to its popularity in classification studies. The implementation of these algorithms is provided by MULAN open source library [52] based on the Weka platform.

All experiments are conducted through 2 × 5-fold cross-validation and the average results are reported. To investigate the statistical significance of the differences among the compared algorithms, the Wilcoxon signed rank test at the 0.05 significance level is employed [53]. Parameter settings of the sampling methods are shown in Table 2. To make fair comparisons, the ranges of the hyperparameters to be tuned for the sampling methods are consistent. In MLSMOTE, MOSOL and LCOS, the number of nearest neighbors k is selected from {3, 4, 5, 6, 7}. The sampling ratio P is selected from {0.1, 0.3, 0.5, 0.7, 0.9} for MLRUS, MLROS and MLSOL. For LCOS, the outer-boundary sampling ratio p, the inner-boundary sampling threshold β and the label correlation threshold τ are selected from {3, 4, 5, 6, 7}, {0.1, 0.2, 0.3, 0.4, 0.5} and {0.5, 0.6, 0.7, 0.8, 0.9}, respectively. Parameter selection is conducted for each data partition using a 2 × 5-fold cross-validation on the training data.

5.2. Experimental results

We conduct experiments based on sixteen public datasets in terms of Macro F_1, Macro AUC, Hamming Loss and Ranking Loss. Figs. 6–7 present the results of the compared sampling methods using six base learners in terms of Macro F_1 and Macro AUC. Overall, our proposed LCOS method outperforms the state-of-the-art methods on most datasets. Especially on some specific datasets (medical, yeast, scene, birds, Corel5k, bibtex, yahoo-Arts, chemistry, chess, cooking and philosophy), LCOS consistently performs better than the other compared methods. To make it clearer, Table 3 shows the average rank of each method as well as its significant wins/losses compared to other methods in terms of four evaluation metrics on six base multi-label classification methods.

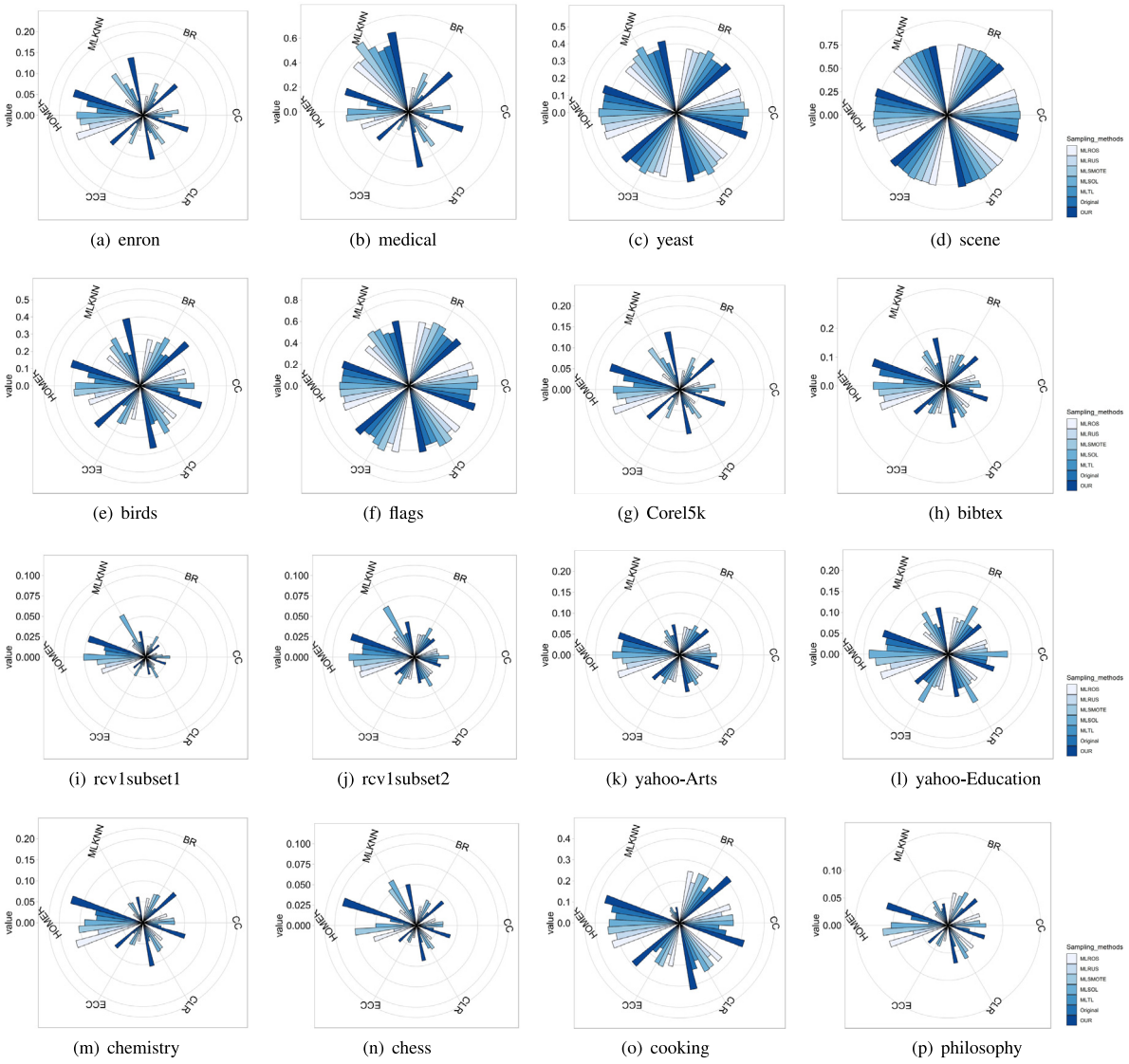


Fig. 6. The performance of the multi-label sampling methods in terms of Macro F_1 across six different classification methods.

The proposed LCOS method clearly achieves the top average rank on all metrics and has the most significant wins without suffering any significant loss. It can be observed that LCOS and MLSOL outperform MLSMOTE. This result reflects that selecting more informative seeds is more effective than directly using all minority seeds for oversampling. We can also observe that MLROS performs worse than the other oversampling methods and MLRUS is inferior to MLTL, which implies that heuristic sampling is more effective than random sampling. An interesting observation is that MLTL and MLRUS even perform worse than the original datasets. The main reason is that they removed some crucial instances, resulting in the loss of vital information. Although MLSOL also considers local imbalance, it ignores the inherent correlations among labels, resulting in limited performance improvement.

Additionally, regarding the overall performance of the base learner, MLKNN has the best performance, achieving an average Macro F_1/Macro AUC of 27.6% and 74.7%. Because MLKNN is a neighborhood-based base learner, the proposed LCOS also depends on local imbalance. Moreover, the synthetic instance is generated by the seed instance and its neighbors. However, the one with the largest performance improvement attracts more attention. From the perspective of performance improvement, although the performance of each classifier has been improved to some extent after oversampling, the performance of CC has improved the most. When CC is used as the base classifier, our LCOS improves the Macro F_1 and Macro AUC by up to 10%, and 1.7%, respectively. The main reason is that CC also considers the high-order correlations among labels, which are more effective than first-order and second-order multi-label classification methods, e.g. BR and CLR. We also compare LCOS with MLROS and MOSOL under the condition that the amount of oversampling is equal. As shown in Table 4, the results indicate that LCOS performs better than the two comparable methods in terms of Macro F_1 and Macro AUC, demonstrating the advantage of exploiting the label correlations for guiding the oversampling. Overall, LCOS achieves a new state-of-the-art on all sixteen datasets against the existing cutting-edge methods.

5.3. Influence of parameters

To explore the influence of parameters p (outer-boundary sampling ratio), β (inner-boundary threshold), τ (correlation threshold), and k (number of nearest neighbors), we employ different settings for each parameter and provide the results in Fig. 8. The outer-boundary sampling ratio p is used to determine the amount of outer-boundary oversampling. From Fig. 8(a), it can

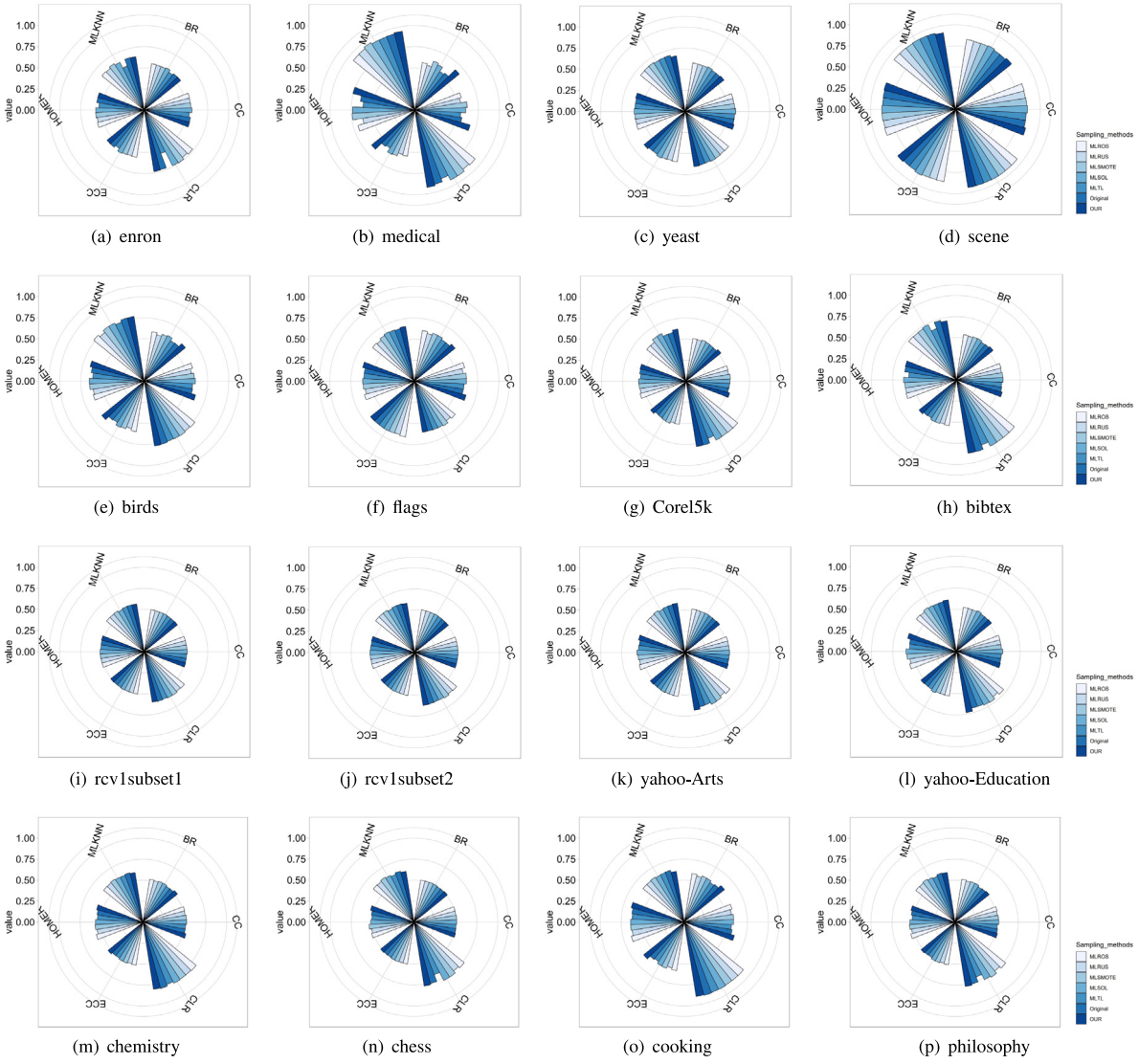


Fig. 7. The performance of the multi-label sampling methods in terms of Macro AUC across six different classification methods.

be seen that the average rank first decreases and then increases with increasing p. This suggests that an appropriate sampling ratio is important for the oversampling method. We can see that when $p = 5$, our method achieves the best performance. The correlation threshold τ is used as a threshold to remove the weaker strength of the relationship between labels. From Fig. 8(b), with the increase in τ, the variation in rank is different for each dataset. This is because the label correlation of distinct datasets is quite different. If τ is too small or too large, most labels are considered strongly correlated or weakly correlated, resulting in less oversampling on the outer-boundaries constituted by the majority labels or the minority labels according to Condition 4 or Condition 1 in our assumption. The inner-boundary threshold β is used to control the sampling of the inner-boundary. With increasing β, the number of oversampling on the inner-boundary decreases. From Fig. 8(c), it can be observed that the rank of most datasets has a similar trend as in Fig. 8(a). This demonstrates that too much oversampling on the inner-boundary may cause an additional within-class problem, while too little oversampling does not introduce sufficient information. From Fig. 8(d), it can be observed that the performance of most datasets first increases and then decreases with increasing k. This is because when k is small, it is susceptible to noisy neighbors. When k is large, it will inaccurately determine seed samples, which negatively influences the following sampling procedure. We observe that $k = 5$ achieves the best performance.

5.4. Effectiveness of learned label correlation

To verify the superiority of our learned label correlation, we compare the performance of our method under label correlations obtained in three different ways: (1) LCOS-ST, which calculates the label co-occurrences to obtain label correlations; (2) LCOS-MI, which uses mutual information [54] to calculate label correlations; (3) LCOS-CL, which employs the SRN model to learn the high-order label correlations. Among them, LCOS-CL is our proposed approach. LCOS-ST and LCOS-MI belong to the method based on low-order label correlations, which only captures the pairwise relationships between labels.

In Table 5, we show the results of LCOS-ST, LCOS-MI and LCOS-CL on Yeast, Enron and Bibtex datasets. From Table 5, it can be observed that LCOS-MI performs better than LCOS-ST. This shows that the label correlations calculated based on mutual information are more effective than the ones based on label co-occurrence. In addition, it can be observed that LCOS-CL outperforms both LCOS-ST and LCOS-MI on the three datasets in terms

Table 3
Average rank of the compared sampling methods using six base learners in terms of four evaluation metrics. The (n1/n2) represents the correction based on the Wilcoxon signed rank test at the 5% level. The corresponding method is significantly better than the n1 methods and worse than the n2 methods. The best method is highlighted in bold (Lower is better).

Metric	Base	Original	MLROS	MLRUS	MLSMOTE	MLTL	MLSOL	LCOS
Macro F_1	BR	4.94(0/3)	3.75(2/1)	6.56(0/4)	3.19(2/0)	5.94(0/3)	2.00(3/0)	**1.38(4/0)**
	CC	5.19(0/3)	4.00(1/1)	6.56(0/4)	3.13(2/0)	5.75(0/2)	2.00(3/0)	**1.31(4/0)**
	MLKNN	5.31(0/3)	4.63(1/2)	6.19(0/4)	2.56(3/0)	5.69(0/2)	1.94(3/0)	**1.69(4/0)**
	CLR	5.06(0/3)	3.69(2/1)	6.56(0/4)	3.19(2/0)	6.06(0/3)	1.94(3/0)	**1.38(4/0)**
	HOMER	5.06(1/3)	3.31(2/1)	6.31(0/4)	3.81(2/0)	6.38(0/4)	2.31(3/0)	**1.75(4/0)**
	ECC	4.81(1/2)	3.88(1/1)	6.38(0/4)	3.19(2/0)	6.31(0/4)	2.13(3/0)	**1.31(4/0)**
	Avg(Total)	5.06(2/17)	3.88(9/7)	6.43(0/24)	3.01(13/0)	6.02(0/18)	2.05(18/0)	**1.47(24/0)**
Macro AUC	BR	4.88(0/3)	3.94(2/1)	6.56(0/4)	3.06(2/0)	5.94(0/3)	2.00(3/0)	**1.31(4/0)**
	CC	5.13(0/3)	4.00(1/2)	6.50(0/4)	3.00(3/0)	5.69(0/2)	2.13(3/0)	**1.25(4/0)**
	MLKNN	3.06(1/0)	3.56(0/0)	5.06(0/1)	**2.00(1/0)**	5.44(0/1)	4.31(0/0)	4.38(0/0)
	CLR	3.88(1/2)	**2.75(3/0)**	6.00(0/4)	2.81(3/0)	6.25(0/4)	3.00(3/0)	3.31(1/0)
	HOMER	5.00(1/2)	3.25(1/1)	6.31(0/4)	2.88(2/1)	6.44(0/4)	2.63(4/1)	**1.44(5/0)**
	ECC	5.06(0/3)	3.69(2/1)	6.38(0/4)	3.44(2/1)	5.81(0/3)	2.00(4/0)	**1.44(4/0)**
	Avg(Total)	4.50(3/13)	3.53(9/5)	6.14(0/21)	2.86(13/2)	5.93(0/17)	2.68(17/1)	**2.19(18/0)**
Hamming loss	BR	4.33(0/2)	3.67(2/1)	5.22(0/4)	3.33(2/0)	5.44(0/4)	2.22(3/0)	**1.44(4/0)**
	CC	4.11(0/2)	3.89(1/1)	5.22(0/3)	2.89(2/0)	5.33(0/4)	2.22(3/0)	**2.00(4/0)**
	MLKNN	5.67(0/3)	**2.44(3/0)**	5.78(0/4)	3.44(1/0)	4.00(0/1)	3.11(1/0)	2.78(2/0)
	CLR	3.89(2/2)	3.78(2/1)	5.22(0/4)	2.78(2/0)	5.78(0/4)	2.22(3/0)	**1.89(4/0)**
	HOMER	4.44(0/1)	3.00(1/0)	5.44(0/3)	2.67(2/0)	4.89(0/2)	5.00(0/2)	**2.11(3/0)**
	ECC	4.11(0/3)	3.56(1/1)	6.33(0/4)	2.44(3/0)	4.56(0/3)	2.67(3/0)	**2.00(4/0)**
	Avg(Total)	4.43(2/13)	3.39(10/4)	5.54(0/22)	2.93(12/0)	5.00(0/18)	2.91(13/2)	**2.04(21/0)**
Ranking Loss	BR	4.13(0/1)	4.75(0/2)	5.63(0/3)	3.63(2/0)	5.75(0/3)	2.38(3/0)	**1.63(4/0)**
	CC	4.25(0/2)	4.75(0/2)	5.63(0/3)	3.88(2/0)	5.63(0/3)	2.38(4/0)	**1.50(4/0)**
	MLKNN	5.38(0/2)	5.50(0/4)	3.00(2/0)	4.13(0/1)	3.88(1/0)	**2.88(3/0)**	3.13(1/0)
	CLR	3.88(1/0)	**2.5(3/0)**	6.25(0/4)	2.88(2/0)	5.00(0/3)	4.38(1/1)	2.88(2/0)
	HOMER	4.50(0/2)	3.63(2/1)	5.50(0/3)	2.75(3/0)	5.38(0/3)	4.25(0/0)	**1.75(4/0)**
	ECC	4.75(0/2)	4.63(0/2)	5.75(0/3)	3.25(2/0)	5.50(0/3)	2.38(4/0)	**1.75(4/0)**
	Avg(Total)	4.48(1/9)	4.29(5/11)	5.29(2/16)	3.42(11/1)	5.19(1/15)	3.10(15/1)	**2.11(19/0)**

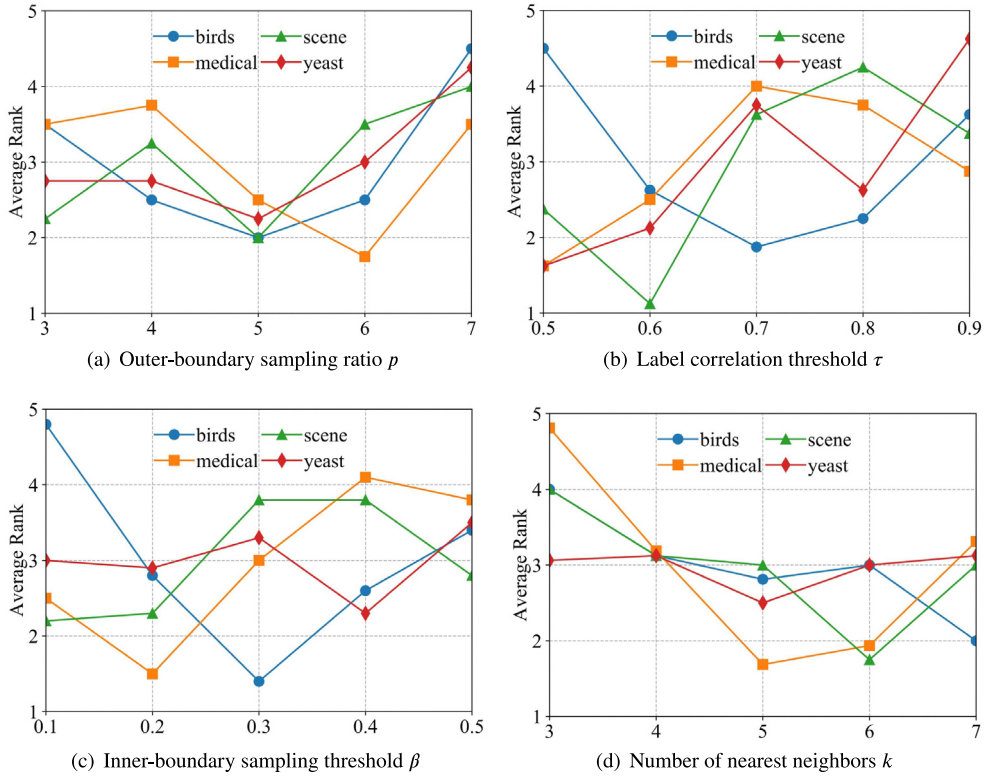


Fig. 8. Results of LCOS with various parameter settings on birds, medical, scene and yeast datasets in terms of average rank on Macro F_1 and Macro AUC with six base learners.

Table 4
Average rank of MacroF_1 and Macro AUC for MLROS, MOSOL, and LCOS on sixteen datasets with six base learners under the setting of the equal oversampling amount (lower is better).

Metric	Base	MLROS	MLSOL	LCOS
Macro F1	BR	2.88	1.69	**1.44**
	CC	2.94	1.69	**1.38**
	MLKNN	2.69	1.75	**1.56**
	CLR	2.88	1.69	**1.44**
	HOMER	2.31	2.00	**1.69**
	ECC	2.81	1.63	**1.56**
	Avg	2.75	1.74	**1.51**
Macro AUC	BR	2.75	1.81	**1.44**
	CC	2.88	1.81	**1.31**
	MLKNN	1.88	2.25	**1.88**
	CLR	2.06	2.19	**1.75**
	HOMER	2.19	2.13	**1.69**
	ECC	2.88	1.75	**1.38**
	Avg	2.44	1.99	**1.57**

Table 5
Results of LCOS with different label correlations using MLKNN on Yeast, Enron and Bibtex datasets.

Metrics	Methods	Yeast	Enron	Bibtex
MacroF1	LCOS-ST	0.4023	0.1233	0.1454
	LCOS-MI	0.4103	0.1243	0.1452
	LCOS-CL	**0.4268**	**0.1385**	**0.1641**
Macro AUC	LCOS-ST	0.6474	0.6102	0.6422
	LCOS-MI	0.6618	0.6122	0.6577
	LCOS-CL	**0.6722**	**0.6270**	**0.6612**
Hamming loss	LCOS-ST	0.2347	0.0829	0.0299
	LCOS-MI	0.2318	0.0818	0.0283
	LCOS-CL	**0.2262**	**0.0814**	**0.0272**
Ranking loss	LCOS-ST	0.2093	0.1415	0.1948
	LCOS-MI	0.2027	0.1409	0.1762
	LCOS-CL	**0.1950**	**0.1373**	**0.1747**

of all evaluation metrics. This demonstrates that our learned high-order label correlations are superior to the low-order ones, which can fully mine the label information and provide more useful information.

5.5. Ablation study

To validate the effect of label correlations, the identification and oversampling of outer-boundary and inner-boundary, we consider three model variants of (1) LCOS-LC, which oversamples on all the outer-boundary and the inner-boundary without considering label correlations; (2) LCOS-O, which only oversamples on the inner-boundary; (3) LCOS-I, which only oversamples on the outer-boundary under the guidance of the learned label correlations.

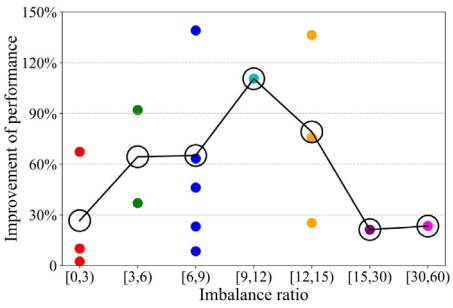
In Table 6, we show the average rank of Macro F_1 and Macro AUC on six base learners and sixteen datasets. It can be observed that LCOS performs better than all variants, regardless of Macro F_1 and Macro AUC. This demonstrates that the proposed method integrates these components in a principled manner to exploit the strengths of each part. Additionally, LCOS-LC is inferior to LCOS indicating that label correlations can effectively guide oversampling. LCOS-O without oversampling the outer-boundary performs worse than LCOS-I without oversampling the inner-boundary, which suggests that the outer-boundary is more essential than the inner-boundary.

5.6. Influences of imbalance level and label number

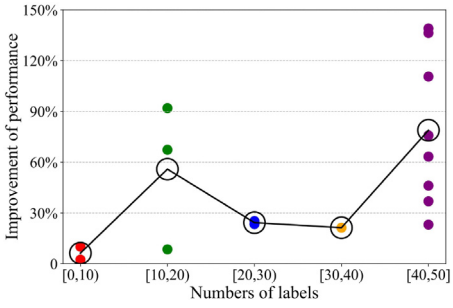
To investigate the influence of different imbalance levels, we compare the performance of our method on different datasets

Table 6
Average rank of the MacroF_1 and Macro AUC metrics for LCOS-LC, LCOS-I, LCOS-O and LCOS with six base learners and sixteen datasets (Lower is better).

Metric	LCOS-LC	LCOS-I	LCOS-O	LCOS
Macro F_1	1.78	2.68	3.88	**1.66**
Macro AUC	1.94	2.85	3.29	**1.86**



(a) The improvement of F_1 under different MeanIR.



(b) The improvement of F_1 under different number of labels.

Fig. 9. Results of LCOS with various imbalance levels and label numbers. The solid circle represents the improvement for each dataset. The hollow circle represents the average of improvements across all datasets with the same scope of imbalance ratio.

with varying imbalance ratio levels in terms of MeanIR. We divide the MeanIR into seven intervals: [0, 3), [3, 6), [6, 9), [9, 12), [12, 15), [15, 30) and [30, 60). We evaluate the average improvements by our LCOS compared with the traditional classification method without any oversampling for each imbalance ratio level. As shown in Fig. 9(a), it can be observed that performance improvement increases gradually at first, then begins to decrease, and finally tends to be stable with increasing MeanIR. This demonstrates that the LCOS algorithm can realize obvious improvements for multiple datasets. Nevertheless, a large imbalanced class distribution will inevitably affect label correlation learning, resulting in limited performance improvement.

In addition, we use the same comparison strategy to investigate the influence of the different label numbers. We divide the label number into five intervals: [0, 10), [10, 20), [20, 30), [30, 40) and [40, 50]. As shown in Fig. 9(b), it can be observed that the improvement in performance increases at first, then decreases to some extent, and increases again as the number of labels increases. The reason for the decreased improvement is that there are some datasets with larger MeanIR resulting in introducing noisy synthetic samples, e.g., the meanIR of yahoo-Education is 53.8. The improvement of our LCOS is still greater than 20% compared with the one without any oversampling. When the number of labels increases, our method can better exploit label correlations to guide oversampling, facilitating oversampling quality and classification.

Table 7
Average rank of Macro F_1 and Macro AUC for different ways to exploit label correlations on six base learners on sixteen datasets (Lower is better). "Learning" and "Statistics" indicate the learned label correlations and the statistical label correlations, respectively.

Correlation			Macro F_1	Macro AUC
Estimation manner	Guidance manner			
	Minority	Majority		
Statistics	Weak	Weak	4.98	4.93
	Weak	Strong	5.58	5.28
	Strong	Strong	6.45	5.81
	Strong	Weak	3.13	3.63
Learning	Weak	Weak	3.62	3.75
	Weak	Strong	4.62	4.60
	Strong	Strong	4.82	4.58
	Strong	Weak	**2.56**	**3.00**

Table 8
Outer boundaries for the minority class labels in the flags dataset based on learned and statistical label correlations.

Label	Label correlation manner	Outer boundaries
y6	Statistics	$\{B^O_{61}, B^O_{62}, B^O_{63}, B^O_{64}\}$
	Learning	$\{B^O_{61}, B^O_{62}, B^O_{63}, B^O_{64}, B^O_{67}\}$
y7	Statistics	$\{B^O_{71}, B^O_{72}, B^O_{73}, B^O_{74}, B^O_{75}\}$
	Learning	$\{B^O_{72}, B^O_{73}, B^O_{74}, B^O_{75}\}$

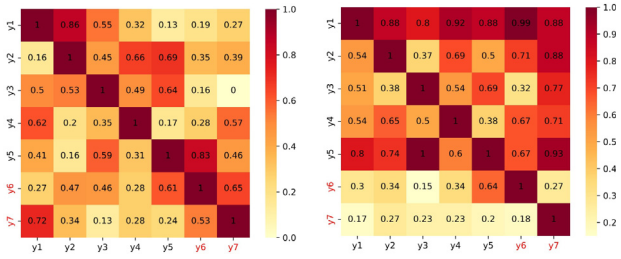
5.7. Effectiveness of exploiting label correlation

To verify our assumption, we perform extensive experiments in four different ways to exploit label correlations. Statistical label correlations are obtained by calculating the label co-occurrences. To verify the best way to guide oversampling with label correlations, we exploit different oversampling conditions for each minority label: weak correlation with a minority class label, strong correlation with a minority class label, weak correlation with a majority class label, and strong correlation with a majority class label.

The oversampling condition of our LCOS is a strong correlation with a minority class label and a weak correlation with a majority class label for each minority class label. Table 7 shows the average rank of the Macro F_1 and Macro AUC for different ways to exploit label correlations with six base learners. From Table 7, it can be observed that LCOS performs best on Macro F_1 and Macro AUC, validating the assumption of our algorithm on how to leverage the label correlations to guide the oversampling. Moreover, the learned label correlations are more appropriate, regardless of any correlation guidance means. For example, Fig. 10(a) and (b) show the learned correlation matrix and the statistical correlation matrix on the flags dataset, respectively. According to IRLBl and MeanIR, the labels y_6 and y_7 are minority class labels, and the other labels are majority class labels. The correlation threshold τ is set to 0.6. Under our assumption, Table 8 shows the outer boundaries of minority class labels based on the statistical and learned label correlations. It can be observed that different label correlations lead to selecting different outer boundaries for the minority class label, affecting the performance of oversampling. The Macro F_1 results of oversampling the minority class label based on different label correlations using MLKNN are shown in Fig. 11. It can be observed that the learned label correlations guided oversampling performs better, which implies that the learned label correlations are more accurate.

6. Conclusion

We present an oversampling method called LCOS to address the multi-label imbalanced data classification problem. The main



(a) Learned correlation matrix. (b) Statistical correlation matrix.

Fig. 10. The label correlation matrices obtained by learning and statistics in the flags dataset.

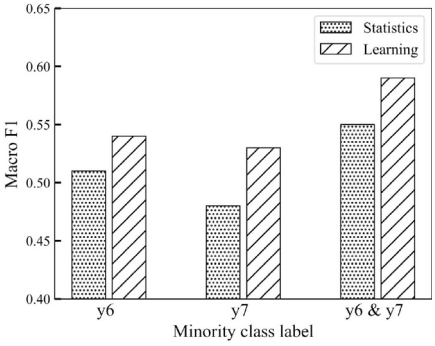


Fig. 11. The comparison of different label correlations using MLKNN in terms of Macro F_1 under the correlation guidance manner of our assumption.

features of LCOS are summarized in three points. First, LCOS can learn the inherent label correlations from complex multi-label datasets. Second, it can select an appropriate subset of each minority class label sample according to the learned correlation matrix. Third, it can assign exact weights to the selected instances according to their importance in the data. Furthermore, we not only demonstrate that label correlations are beneficial for multi-label data oversampling but also suggest an effective strategy to leverage it to guide oversampling. We conducted experiments on sixteen public datasets and the results show that LCOS achieves highly competitive performance against state-of-the-art algorithms in terms of various assessment metrics, such as Macro F_1 and Macro AUC. Our study provides a research direction to extend sampling methods to the multi-label imbalanced learning task.

In future work, we need to improve the efficiency of candidate seed identification and take label correlations into consideration when assigning labels to newly generated instances. Additionally, we need to consider the case of missing labels, because it is too expensive to obtain complete labeled data, and there are often incorrect labels.

CRediT authorship contribution statement

Kai Zhang: Writing – review & editing, Writing – original draft, Methodology. **Zhaoyang Mao:** Validation, Software, Data curation. **Peng Cao:** Supervision, Resources, Formal analysis, Conceptualization. **Wei Liang:** Visualization, Validation. **Jinzhu Yang:** Writing – review & editing, Validation. **Weiping Li:** Writing – review & editing. **Osmar R. Zaiane:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research is supported by the National Key Research and Development Program of China (No. 2020YFC0833302), the National Natural Science Foundation of China (No. 62076059), the Science Project of Liaoning Province (2021-MS-105) and the 111 Project (B16009). O. Zaiane is supported by Amii and CIFAR.

References

[1] W. Zhao, S. Kong, J. Bai, D. Fink, C. Gomes, Hot-vae: Learning high-order label correlation for multi-label classification via attention-based variational autoencoders, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 15016–15024.
[2] Y. Sun, L. Cai, B. Liao, W. Zhu, Minority sub-region estimation-based oversampling for imbalance learning, IEEE Trans. Knowl. Data Eng. 34 (5) (2022) 2324–2334, http://dx.doi.org/10.1109/TKDE.2020.3010013.
[3] L.A. Cabrera-Diego, N. Bessis, I. Korkontzelos, Classifying emotions in Stack Overflow and JIRA using a multi-label approach, Knowl.-Based Syst. 195 (2020) 105633.
[4] T. Pham, X. Tao, J. Zhang, J. Yong, Y. Li, H. Xie, Graph-based multi-label disease prediction model learning from medical data and domain knowledge, Knowl.-Based Syst. 235 (2022) 107662.
[5] B. Al-Salemi, M. Ayob, G. Kendall, S.A.M. Noah, Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms, Inf. Process. Manage. 56 (1) (2019) 212–227.
[6] L. Li, P. Cao, J. Yang, O.R. Zaiane, Modeling global and local label correlation with graph convolutional networks for multi-label chest X-ray image classification, Med. Biol. Eng. Comput. 60 (9) (2022) 2567–2588.
[7] Z.A. Daniels, D.N. Metaxas, Addressing imbalance in multi-label classification using structured hellinger forests, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 1826–1832.
[8] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, Dealing with difficult minority labels in imbalanced mutilabel data sets, Neurocomputing 326 (2019) 39–53.
[9] A.N. Tarekegn, M. Giacobini, K. Michalak, A review of methods for imbalanced multi-label classification, Pattern Recognit. 118 (2021) 107965.
[10] M.-L. Zhang, Y.-K. Li, H. Yang, X.-Y. Liu, Towards class-imbalance aware multi-label learning, IEEE Trans. Cybern. (2020).
[11] S. Pouyanfar, T. Wang, S. Chen, A multi-label multimodal deep learning framework for imbalanced data classification, in: 2nd IEEE Conference on Multimedia Information Processing and Retrieval, MIPR 2019, San Jose, CA, USA, March 28-30, 2019, IEEE, 2019, pp. 199–204, http://dx.doi.org/10.1109/MIPR.2019.00043.
[12] Y. Yan, M. Tan, Y. Xu, J. Cao, M. Ng, H. Min, Q. Wu, Oversampling for imbalanced data via optimal transport, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 5605–5612.
[13] P. Majumdar, R. Singh, M. Vatsa, On learning deep models with imbalanced data distribution, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 15720–15721.
[14] L. Wang, S. Xu, X. Wang, Q. Zhu, Addressing class imbalance in federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 10165–10173.
[15] T. Zhu, Y. Lin, Y. Liu, Improving interpolation-based oversampling for imbalanced data learning, Knowl.-Based Syst. 187 (2020) 104826.
[16] T. Zhang, Y. Li, X. Wang, Gaussian prior based adaptive synthetic sampling with non-linear sample space for imbalanced learning, Knowl.-Based Syst. 191 (2020) 105231.
[17] C.-F. Tsai, W.-C. Lin, Y.-H. Hu, G.-T. Yao, Under-sampling class imbalanced datasets by combining clustering analysis and instance selection, Inform. Sci. 477 (2019) 47–54.
[18] A.Y. Taha, S. Tiun, A.H. Abd Rahman, A. Sabah, Multilabel over-sampling and under-sampling with class alignment for imbalanced multilabel text classification, J. Inf. Commun. Technol. 20 (3) (2021).
[19] M.A. Tahir, J. Kittler, F. Yan, Inverse random under sampling for class imbalance problem and its application to multi-label classification, Pattern Recognit. 45 (10) (2012) 3738–3750.
[20] F. Charte, A. Rivera, M.J.d. Jesus, F. Herrera, On the impact of dataset complexity and sampling strategy in multilabel classifiers performance, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2016, pp. 500–511.
[21] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[22] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, Springer, 2005, pp. 878–887.
[23] S. Barua, M.M. Islam, X. Yao, K. Murase, MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning, IEEE Trans. Knowl. Data Eng. 26 (2) (2014) 405–425, http://dx.doi.org/10.1109/TKDE.2012.232.
[24] F. Charte, A.J. Rivera, M.J. Del Jesus, F. Herrera, MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation, Knowl.-Based Syst. 89 (NOV.) (2015) 385–397.
[25] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2013) 1819–1837.
[26] R. Wang, S. Kwong, X. Wang, Y. Jia, Active k-labelsets ensemble for multi-label classification, Pattern Recognit. 109 (2021) 107583, http://dx.doi.org/10.1016/j.patcog.2020.107583.
[27] M.-L. Zhang, Z.-H. Zhou, ML-KNN: A lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.
[28] X. Wen, D. Li, C. Zhang, Y. Zhai, A weighted ML-KNN based on discernibility of attributes to heterogeneous sample pairs, Inf. Process. Manage. 59 (5) (2022) 103053.
[29] E. Yao, D. Li, Y. Zhai, C. Zhang, Multilabel feature selection based on relative discernibility pair matrix, IEEE Trans. Fuzzy Syst. 30 (7) (2021) 2388–2401.
[30] K. Brinker, J. Fürnkranz, E. Hüllermeier, A unified model for multilabel classification and ranking, in: Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29–September 1, 2006, Riva del Garda, Italy, 2006, pp. 489–493.
[31] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, K. Brinker, Multilabel classification via calibrated label ranking, Mach. Learn. 73 (2) (2008) 133–153.
[32] G. Nan, Q. Li, R. Dou, J. Liu, Local positive and negative correlation-based k-labelsets for multi-label classification, Neurocomputing 318 (2018) 90–101.
[33] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (3) (2011) 333–359.
[34] Q.-W. Zhang, Y. Zhong, M.-L. Zhang, Feature-induced labeling information enrichment for multi-label learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018, pp. 4446–4453.
[35] L. Sun, T. Yin, W. Ding, Y. Qian, J. Xu, Feature selection with missing labels using multilabel fuzzy neighborhood rough sets and maximum relevance minimum redundancy, IEEE Trans. Fuzzy Syst. 30 (5) (2021) 1197–1211.
[36] H.-C. Chou, C.-C. Lee, C. Busso, Exploiting co-occurrence frequency of emotions in perceptual evaluations to train a speech emotion classifier, in: Proc. Interspeech, Vol. 2022, 2022, pp. 161–165.
[37] J. Li, P. Li, X. Hu, K. Yu, Learning common and label-specific features for multi-Label classification with correlation information, Pattern Recognit. 121 (2022) 108259.
[38] L. Sun, T. Wang, W. Ding, J. Xu, Y. Lin, Feature selection using Fisher score and multilabel neighborhood rough sets for multilabel classification, Inform. Sci. 578 (2021) 887–912.
[39] J. Huang, Q. Xu, X. Qu, Y. Lin, X. Zheng, Improving multi-label learning by correlation embedding, Appl. Sci. 11 (24) (2021) 12145.
[40] Y. Chen, C. Zou, J. Chen, Label-aware graph representation learning for multi-label image classification, Neurocomputing 492 (2022) 50–61, http://dx.doi.org/10.1016/j.neucom.2022.04.004.
[41] Z.-M. Chen, X.-S. Wei, P. Wang, Y. Guo, Multi-label image recognition with graph convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5177–5186.
[42] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, Addressing imbalance in multilabel classification: Measures and random resampling algorithms, Neurocomputing 163 (2015) 3–16.
[43] R.M. Pereira, Y.M.G. Costa, C.N.S. Jr., MLTL: A multi-label approach for the Tomek Link undersampling algorithm, Neurocomputing 383 (2020) 95–105, http://dx.doi.org/10.1016/j.neucom.2019.11.076.
[44] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, MLeNN: A first approach to heuristic multilabel undersampling, in: E. Corchado, J.A. Lozano, H. Quintián, H. Yin (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2014 - 15th International Conference, Salamanca, Spain, September 10-12, 2014. Proceedings, in: Lecture Notes in Computer Science, vol. 8669, Springer, 2014, pp. 1–9, http://dx.doi.org/10.1007/978-3-319-10840-7_1.
[45] B. Liu, K. Blekas, G. Tsoumakas, Multi-label sampling based on local label imbalance, Pattern Recognit. 122 (2022) 108294.
[46] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771, http://dx.doi.org/10.1016/j.patcog.2004.03.009.
[47] T. Blickle, L. Thiele, A comparison of selection schemes used in evolutionary algorithms, Evol. Comput. 4 (4) (1996) 361–394.
[48] B. Liu, G. Tsoumakas, Making classifier chains resilient to class imbalance, in: Asian Conference on Machine Learning, PMLR, 2018, pp. 280–295.

[49] Y. Xie, M. Qiu, H. Zhang, L. Peng, Z. Chen, Gaussian distribution based oversampling for imbalanced data classification, IEEE Trans. Knowl. Data Eng. 34 (2) (2020) 667–679.
[50] B. Liu, G. Tsoumakas, Dealing with class imbalance in classifier chains via random undersampling, Knowl.-Based Syst. 192 (2020) 105292.
[51] K. Sechidis, G. Tsoumakas, I. Vlahavas, On the stratification of multi-label data, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 145–158.
[52] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, I. Vlahavas, Mulan: A java library for multi-label learning, J. Mach. Learn. Res. 12 (2011) 2411–2414.
[53] A. Benavoli, G. Corani, F. Mangili, Should we really use post-hoc tests based on mean-ranks? J. Mach. Learn. Res. 17 (2016) 5:1–5:10.
[54] D. You, Y. Wang, J. Xiao, Y. Lin, M. Pan, Z. Chen, L. Shen, X. Wu, Online multi-label streaming feature selection with label correlation, IEEE Trans. Knowl. Data Eng. (2021).