基于安全欠采样的不均衡多标签数据集成学习方法

孙中彬[1,2],刁宇轩[1,2],马苏洋[2]
(1. 中国矿业大学计算机科学与技术学院,江苏徐州 221116;2. 矿山数字化教育部工程研究中心,江苏徐州 221116)

摘 要: 多标签分类任务广泛存在于现实生活中,然而其经常存在不均衡数据问题,严重影响了分类性能. 目前解决该问题的主流技术为重采样方法,主要分为过采样和欠采样,过采样通过生成与少数类标签相关的样本,欠采样则是通过删除与多数类标签相关的样本. 然而,这些方法都专注于解决一种不均衡问题,即标签内不均衡或标签间不均衡,导致在解决一种不均衡的同时可能引入另一种不均衡. 针对该问题,本文提出一种基于安全欠采样的不均衡多标签数据集成学习方法ESUS(Ensemble learning method based on Safe Under-Sampling). 首先通过标签划分将多标签不均衡数据集划分成单标签数据集和标签对数据集,针对单标签数据集,提出一种安全欠采样方法解决标签内不均衡问题,并利用采样后的均衡数据集构建二分类模型. 对于标签对数据集,进行数据剪枝后利用集成学习解决标签间不均衡问题,在保持分类性能的同时降低时空复杂度. 最后将单标签数据集模型和标签对数据集模型集成为最终的分类模型. 在六个多标签不均衡数据集上的实验结果表明:和七种对比方法相比,ESUS方法在四个评价指标上更稳定有效.

关键词: 多标签分类;不均衡数据;标签划分;安全欠采样;数据剪枝;集成学习
基金项目: 中央高校基本科研业务费专项资金资助(No.2021QN1075)
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112(2024)10-3392-17
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20240210

An Imbalanced Multi-Label Data Ensemble Learning Method Based on Safe Under-Sampling

SUN Zhong-bin[1,2], DIAO Yu-xuan[1,2], MA Su-yang[2]
(1. *School of Computer Science and Technology*, *China University of Mining and Technology*, *Xuzhou*, *Jiangsu* 221116, *China*;
2. *Mine Digitization Engineering Research Center of the Ministry of Education*, *Xuzhou*, *Jiangsu* 221116, *China*)

Abstract: The task of multi-label classification is widely present in real life, but there is often an issue of imbalanced data, which seriously affects the classification performance. At present, the mainstream technology for solving this problem is resampling, which are mainly divided into over-sampling and under-sampling. Particularly, over-sampling generates samples related to minority class labels while under-sampling removes samples related to majority class labels. However, these methods all focus on solving an imbalance problem, namely intra label imbalance or inter label imbalance, which may introduce another imbalance problem while solving one imbalance problem. In response to this issue, this paper proposes an imbalanced multi-label data ensemble learning method ESUS (Ensemble learning method based on Safe Under-Sampling) based on safe under-sampling. Firstly, the imbalanced multi-label dataset is divided into single label datasets and label pair datasets through label partitioning. For single label datasets, this paper proposes a secure under-sampling method to solve the problem of intra label imbalance, and constructs binary classification models using the sampled balanced dataset. For label pair datasets, ensemble learning is used on the pruned data to solve the problem of inter label imbalance, which may maintain the classification performance of the model and reduce spatiotemporal complexity. Finally, the single label dataset models and label pair dataset models are integrated into the final classification model. The experimental results on six imbalanced multi-label datasets show that compared with seven comparison methods, the ESUS method is more stable and effective on four evaluation metrics.

Key words: multi-label classification; imbalanced data; label partitioning; safe under-sampling; data pruning; ensemble learning
Foundation Item(s): The Fundamental Research Funds for the Central Universities (No.2021QN1075)

收稿日期:2024-03-04;修回日期:2024-07-22;责任编辑:覃怀银

1 引言

多标签分类方法在数据挖掘与机器学习研究领域备受关注,其在生物医学[1],信息安全[2]和金融[3]等实际场景中有着广泛的应用. 近年来,研究学者对于多标签分类问题的关注越来越多,提出了大量的多标签分类方法[4,5]. 具体而言,多标签分类方法主要分为问题转化方法[6]和算法自适应方法[7~9]两大类. 算法自适应方法主要基于多标签分类问题的数据特性对原有的分类算法进行改进,即扩展单标签分类方法适应多标签数据集,比如MLKNN[10]和Rank-SVM[9]都是对传统分类算法进行改进解决多标签分类问题. 问题转化方法则是通过将多标签分类问题转化为二分类问题或者多分类问题解决,常用的问题转化方法包括Binary Relevance[11] (BR), Label Power-set[12] (LP)和Classifier Chains[13](CC)等.

在分类问题研究中,尽管收集到的数据量非常大,但是对人们有用的类别数据往往非常有限,通常仅占全部数据的一小部分,这种某类样本数量明显少于其他类样本数量的数据称为不均衡数据[14,15]. 不均衡数据的存在会严重影响分类性能,针对二分类问题,研究人员提出了很多方法来解决不均衡数据问题[16~18]. 不均衡数据同样影响多标签数据的分类性能,为了解决多标签分类中的不均衡数据问题,研究人员也已经提出了很多有效的不均衡数据挖掘方法来提升分类性能[19]. 重采样方法是最常用的方法,因为其通常独立于所使用的分类算法[20,21]. 重采样方法主要包括欠采样方法和过采样方法[22],然而现有的多标签重采样方法大都只专注于一类不均衡问题,即标签内不均衡或标签间不均衡,这些方法在解决一类不均衡问题的同时可能会引入另一类不均衡问题,这极有可能导致分类性能下降.

针对上述问题,本文提出一种基于安全欠采样的不均衡多标签数据集成学习方法ESUS(Ensemble learning method based on Safe Under-Sampling). ESUS方法首先通过标签划分将多标签不均衡数据集划分为多个单标签数据集和标签对数据集. 针对单标签数据集,提出了一种安全欠采样方法,通过该方法创建多数类的安全子区域,并从安全子区域中选取安全样本,利用安全样本来构建分类模型,从而解决标签内不均衡问题;针对标签对数据集,进行数据剪枝后利用集成学习解决标签间不均衡问题,在保持分类性能的同时降低时空复杂度. 最后,将使用单标签数据集和标签对数据集构建的模型利用特定的集成规则进行集成,得到最终的多标签数据分类模型. 在实验中,选取六个不同领域的多标签不均衡数据集,使用了三种分类算法和七种对比方法. 在四个常用多标签分类性能评价指标上的实验结果表明,ESUS方法比七种对比方法更稳定有效. 此外,本文还进行了消融实验研究,分别验证了安全欠采样和数据剪枝模块的有效性.

2 相关工作

近年来,研究人员提出了许多不均衡数据分类方法来解决多标签类不均衡问题[19,23]. 在这些方法中,重采样因其独立于所使用的分类算法,成为了最常用的多标签类不均衡问题处理方法,取得了不错的分类性能. 在现有的重采样方法中,根据其处理样本策略的不同可划分为欠采样方法和过采样方法.

欠采样方法通过去除与多数类标签相关的样本来解决不均衡数据问题. LPRUS[24]是一种基于LP策略[6]的代表性欠采样方法,它会随机删除有最频繁标签集的实例,直到多标签数据集中的样本数量减少到指定的百分比,通过这种方式使多标签数据集达到均衡,从而解决多标签分类任务中的不均衡数据问题. 然而LPRUS方法通常会受到标签稀疏性的限制,故研究人员提出了MLRUS[22]方法来解决这一限制. MLRUS是基于单个标签的,而不是完整的标签集,它通过删除携带多数类标签的样本来对数据集进行重采样. 除了上述这种随机欠采样方法外,还有一些启发式欠采样方法,比如:MLTL[25]算法基于经典的Tomek Link[26]算法来解决不均衡数据,MLeNN[21]方法则是建立在ENN[27]规则的基础上来处理不均衡数据问题. 陈旭等人还提出了一种USIB[28]方法,它是一种基于迭代提升欠采样的分类方法,通过对多数类迭代的进行欠采样来解决不均衡问题. Liu等人提出了一种基于局部标签不均衡的多标签欠采样方法MLUL[29]. MLUL通过删除有害的样本来使少数类更易学习,但是并不是直接删除该样本,而是选择出重要样本的子集并删除其余的样本,从而解决不均衡问题.

过采样方法通过生成与少数类标签相关的样本来处理不均衡问题. LPROS[24]是与LPRUS对应的过采样算法,它通过随机复制少数标签集样本来平衡数据集,直到多标签数据集的大小增加到指定的百分比. MLROS[22]则是与MLRUS对应的过采样方法,它通过克隆带有少数标签的样本来进行重采样. 除了上述随机过采样方法外,还有一些启发式过采样方法. 基于SMOTE[30]算法,研究人员提出了MLSMOTE[31]方法和FCSMI[32]方法. MLSMOTE方法设计了少数类标签列表,它将这些标签出现的实例作为种子,生成新的实例;FCSMI方法则是通过计算少数类样本与其他样本的距离作为特征,然后利用SMOTE方法来解决不均衡问题. Payel Sadhukhan等人[33]根据近邻原则设计了一种基于反向最近邻域的过采样方法,它通过在每个标

签的少数点的反向最近邻域中添加标签特定的合成少数实例来解决类不均衡问题. LIU 等人提出了一种MLSOL[29]方法,它是一种基于局部标签不均衡的过采样方法,利用过采样来解决局部区域的不均衡问题,从而一定程度上提升模型的分类性能. Wonkeun Jo 等人提出了一种基于生成对抗网络的过采样方法OBGAN[34],它通过在损失函数中引入超参数,增加对少数类的识别精度,从而解决类不均衡问题. Kai Zhang 等人基于标签的相关性提出了一种过采样方法LCOS[35],该方法通过定义两个边界区域,从中选取候选种子,然后利用加权策略从候选种子的子集中根据重要性选取最终的种子,最后通过插值的方法生成少数类实例,从而解决不均衡问题.

综上所述,过采样和欠采样都是通过对原始数据集删除或者生成样本来解决多标签分类任务中的不均衡数据问题. 尽管这些方法在性能上取得了一定的进步,但只解决了标签内不均衡或是标签间不均衡,它们没有考虑在解决一类不均衡的同时可能会引入另一类不均衡,从而导致分类性能下降. 为此,本文旨在同时处理两类不均衡问题,并提高模型的分类性能.

3 ESUS方法

3.1 方法框架

本文提出了一种基于安全欠采样的不均衡多标签数据集成学习方法 ESUS. 针对多标签数据集,考虑将其转化为二分类数据集进行处理,因此依据多标签数据集的标签特性,将其转化为单标签数据集,并基于单标签数据集构建分类模型来预测实例是否属于该标签. 然而如果预测结果为不属于该标签,单标签模型则不能确定当前实例属于其余哪个标签,因此考虑同步构建标签对数据集,并基于标签对数据集构建标签对分类模型. 综上所述,ESUS方法主要由五个部分组成:标签划分,安全欠采样,数据剪枝,二分类模型构建及模型集成,其框图如图1所示.

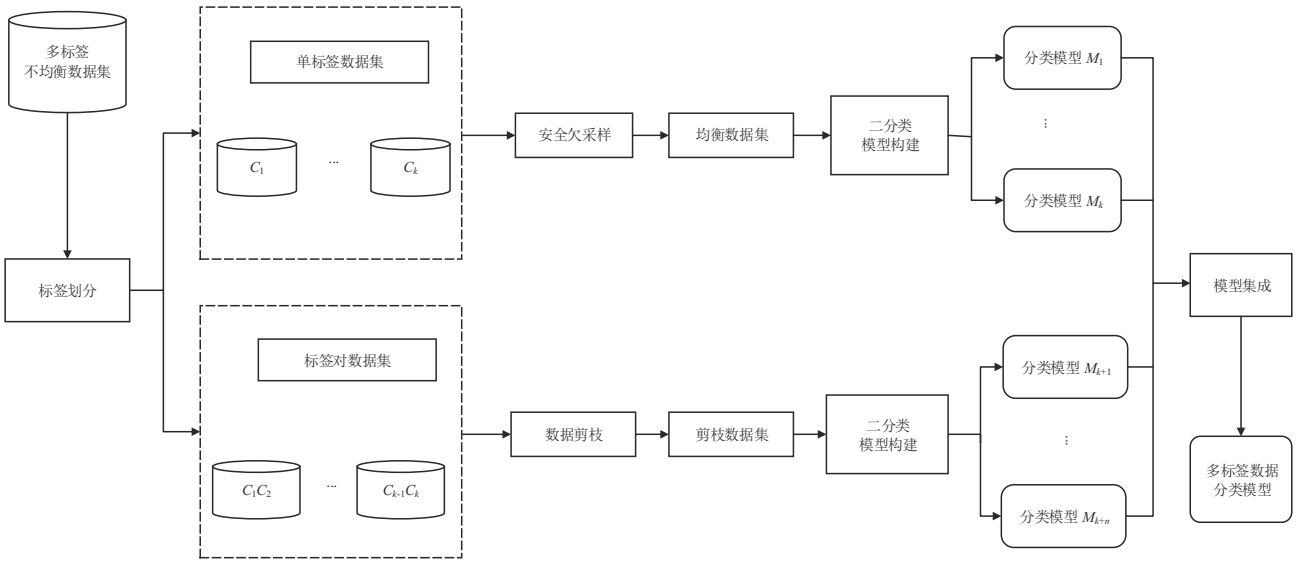

图1 ESUS方法框图

如图1所示,对于一个含有k个标签值的多标签不均衡数据集,首先通过标签划分方法将其转化成k个单标签数据集和C_k^2个标签对数据集. 其次,针对单标签数据集,先进行安全欠采样后构建k个二分类模型;针对标签对数据集,先进行数据剪枝后得到n个标签对数据集,然后针对剪枝数据集学习构建n个二分类模型. 最后利用模型集成将这$k+n$个二分类模型集成得到最终的多标签数据分类模型.

为了方便介绍,首先对问题进行了以下定义:假设$X=\mathrm{R}^d$表示d维的实例属性空间,$Y=\{y_1,y_2,\cdots,y_k\}$表示具有k个可能类标签的标签空间,$D=\{(x_i,Y_i)|1\leqslant i\leqslant m\}$表示具有$m$个实例的训练集. 对于每个多标签样本$(x_i,Y_i),x_i\in X$是一个$d$维的特征向量$(x_{i1},x_{i2},\cdots,x_{id})^{\mathrm{T}}$,$Y_i\subseteq Y$是跟样本$x_i$相关的标签集. 多标签学习的目标就是从具有$m$个样本的训练集$D$中学习一个映射函数$h:X\to 2^y$. 对于未知的样本$x\in X$,通过多标签分类器$h$可以得到其预测的标签集$h(x)\in Y$.

3.2 标签划分

对于具有k个标签的多标签不均衡数据集,考虑到该数据集中同时存在标签内不均衡和标签间不均衡两个问题,首先使用标签划分方法将该数据集转化为k个单标签数据集和C_k^2个标签对数据集.

单标签数据集:针对每个标签C_i,标签划分去除标

签 C_i 外的其余 k-1 个标签信息,只保留一个标签值,从而将多标签不均衡数据集转化为 k 个单标签数据集. 这些单标签数据集的样本数量是相同的,每个样本的标签值为是否包含对应的标签,由于包含和不包含该标签的样本数量可能存在较大的差异,即单标签数据集可能存在标签内不均衡问题,因此后面要考虑解决单标签数据集存在的标签内不均衡问题.

标签对数据集:标签划分将标签 C_i 和 C_j 进行两两组合生成 C_k^2 个标签对组合,从而生成 C_k^2 个标签对数据集. 因此标签对数据集只保留了 C_i 和 C_j 的标签信息,去除了标签 C_i 和 C_j 外的其余标签信息. 根据标签的组合信息生成新的标签值,对于只携带标签 C_i 而不携带 C_j 的实例标记为1,而只携带标签 C_j 而不携带标签 C_i 的实例标记为0. 由于标签 C_i 和 C_j 对应的样本数量可能存在较大的差异,即标签对数据集可能存在标签间不均衡问题,因此后面要考虑解决标签对数据集存在的标签间不均衡问题.

3.3　安全欠采样

标签划分之后得到的单标签数据集可能会存在标签内不均衡问题,例如,对于标签 C_i 来说,不包含该标签的实例较多,包含该标签的实例较少,这便是标签内不均衡问题,该问题会导致构建的分类模型对标签 C_i 的预测精度较低. 针对这种二类不均衡问题,现有一些过采样方法可以解决该问题,然而过采样方法会增加训练样本的数量,会增加计算的复杂度. 和过采样方法相比,欠采样方法降低了计算复杂度,但是在欠采样的过程中有可能会删除潜在的有用样本信息. 因此,如何在欠采样过程中保留这些潜在的有用样本是十分重要的.

Tuanfei Zhu 等人提出了清洁子区域[36]的概念. 在清洁子区域概念中,首先选取一个样本作为中心点样本,然后计算得到中心点样本的所有近邻样本,并根据距离从小到大排序. 在排序后的近邻样本中,如果有连续 q 个近邻样本与中心点样本的类别不同,则将 q 个样本前的区域称为该中心点样本的清洁子区域,其中 q 是人为设置的用于划分清洁子区域的阈值. 由此可看出,如果选择一个多数类样本作为中心点样本,其清洁子区域中的所有多数类样本都是相对安全有用的样本. 因此基于清洁子区域概念,本文提出了一种安全欠采样方法,该方法主要包括两个部分:清洁子区域划分以及安全样本选取.

在安全欠采样方法中,为了便于寻找多数类样本的清洁子区域,首先对多数类样本进行 KMeans[37] 聚类,使得同一簇内的样本相似度更高,这样便于尽快找到多数类样本的清洁子区域;在完成多数类样本聚类划分后,针对每个簇中的样本利用清洁子区域划分算法来进行划分,进而获得每个多数类样本的清洁子区域同类样本集合. 然而,有时可能会存在清洁子区域中的样本数量无法满足所需安全样本的需求,这时就需要选取那些相对安全的样本来填充安全样本集合. 这些相对安全的样本,本文称之为次安全样本. 针对这些次安全样本,安全欠采样方法通过计算每个多数类样本与所有少数类样本的平均距离,并根据平均距离的大小来选取次安全样本. 某个多数类样本的平均距离越大说明其与所有少数类样本距离较远,也就说明该样本相对安全. 安全欠采样方法的伪代码如算法1所示.

算法1　SafeUnderSampling

输入:

训练样本集 D
清洁子区域阈值 q
类别个数 k
多数类样本采样比例 ratio

输出:

安全样本集合 SafeSample

```
1: 初始化:
    Maj ← 多数类样本集合
    Min ← 少数类样本集合
    N ← Maj样本实例个数 * ratio
    CleanSample ← ∅
    SafeSample ← ∅
    SecondarySafeSample ← ∅
2: KMeans(Maj)
3: FOR i = 1 → k
4:   Cluster_i ← KMeans.get(i)
5:   μ ←Cluster_i * ratio
6:   CleanSample_i ← CleanAreaSpilt(Cluster_i, D, q)
7:   u ←CleanSample_i的样本数量
8:   IF (u < μ)
9:    SafeSample ← SafeSample.add(CleanSample_i)
10:    SecondarySafeSample_i ←
     ObtainSecondarySafeSample(Cluster_i, Min, CleanSample_i, μ−u)
11:    SafeSample ← SafeSample.add(SecondarySafeSample_i)
12:  ELSE
13:     FOR j = 1 → μ
14:        SafeSample ← SafeSample.add(CleanSample_i.get(j))
15:     END FOR
16:  END IF
17: END FOR
18: Return SafeSample
```

在算法1中,首先对多数类样本进行了 KMeans 聚类,将多数类样本聚为 k 个簇(行2);其次针对每个簇,通过设置的采样比例计算每个簇中需要选取的安全多数类样本数量 μ,同时使用清洁子区域划分方法

CleanAreaSplit对每个簇进行划分，得到该簇对应清洁子区域中的清洁样本集合CleanSample（行4~7）；然后根据清洁样本集合的数量u与需要选取的安全样本数量μ的关系来进行安全样本的选取. 如果$u<\mu$的话，那就需要进行次安全样本的选取（行8~11）；否则就从清洁样本集合中选取μ个安全样本（行12~16）. 最后返回安全样本集合（行18）. 安全样本集合中的多数类样本将与原始训练集中的少数类样本构成新的训练集，用于下一步的二分类模型构建.

在清洁子区域划分中，旨在划分出一个多数类样本的安全区域，即该区域中尽可能少出现少数类样本，最好是没有少数类样本，从而在欠采样中保留安全的多数类样本. 具体来说，清洁子区域划分首先将多数类簇中的每个样本依次作为中心点样本，计算获得中心点样本在训练集中的所有近邻并根据距离进行排序；然后根据近邻样本的类别来进行划分，如果有连续q个样本为少数类，则将q个样本前的区域划分为该样本的清洁子区域. 清洁子区域划分方法的伪代码如算法2所示.

在算法2中，依次选取多数类簇中的每个多数类样本作为中心点样本，然后计算获得中心点样本与训练集中其他样本的距离，并根据距离进行升序排序（行3~4）；之后根据近邻列表判断近邻实例是否为少数类，如果有连续q个实例为少数类，则将q个少数类实例之前的多数类实例集合称之为该中心点的清洁子区域（行5~13）. 最后返回清洁子区域中的多数类样本集合（行15）.

算法2 CleanAreaSpilt

```
输入:
    簇样本集合Cluster
    训练集D
    清洁子区域阈值q
输出:
    清洁样本集合CleanSample
1: 初始化:
    N ← 簇中所有样本个数
    n ← 训练集样本个数
    CleanSample ← ∅
    MinIndex ← 少数类样本类别序号
2: FOR i = 1 → N
3:     distance[n-1] ← ObtainDistance(Instance i, D)
4:     Sort (distance) //对distance升序排列
5:     FOR m = 1 → n-1
6:         count ← 0
7:         neighbor ← ObtainNeighbor(distance[m])
8:         IF(neighbor.classvalue != MinIndex && count < q)
9:             CleanSample.add(neighbor)
10:        ELSE
11:            count++
12:        END IF
13:    END FOR
14: END FOR
15: Return CleanSample
```

在安全样本选取中，如果簇中清洁子区域的安全样本数量u大于该簇中需要选取的安全多数类样本数量μ，则从清洁子区域中选择μ个多数类样本，否则进行次安全样本的选取. 次安全样本主要通过多数类与少数类样本的平均距离来进行选取，如果多数类样本与少数类样本的平均距离较大，则说明该样本离少数类较远，同时也说明了其相对安全. 次安全样本方法的伪代码如算法3所示.

算法3 ObtainSecondarySafeSample

```
输入:
    簇样本集合Cluster
    少数类样本集合Min
    清洁样本集合CleanSample
    需要选取的次安全样本数量μ-u
输出:
    次安全样本集合SecondarySafeSample
1: 初始化:
    N ← 簇中所有样本个数
    n ← 少数类样本个数
    SecondarySafeSample ← ∅
2: FOR i = 1 → N
3:     IF (!CleanSample.contains(Instance i))
4:         Mean ← ObtainMeanDistance(Instance i, Min)
5:     END IF
6: END FOR
7: Sort(Mean) //对平均距离进行降序排列
8: FOR j = 1 → μ-u
9:     Sample ← ObtainSample(Mean [j])
10:    SecondarySafeSample.add(Sample)
11: END FOR
12: Return SecondarySafeSample
```

在算法3中，对于清洁子区域外的多数类样本，计算它们与少数类近邻的平均距离（行3~4），并对平均距离进行降序排列（行7），然后将距离少数类样本较远的多数类样本加入次安全样本集合中（行8~11），最后返回次安全样本集合（行12）.

3.4 数据剪枝

针对标签划分后的标签对数据集，如果原始多标签数据集中的标签数量很多，则导致划分的标签对数据集数量很大，对如此多的标签对数据集进行学习构建二分类模型，时空复杂度都比较大. 例如在多标签分

类中常用的两个数据集Bibtex和CAL500,Bibtex具有159个类标签,CAL500具有174个类标签,在使用标签划分后,Bibtex会生C_{159}^{2}=12 561个标签对数据集,而CAL500会生成C_{174}^{2}=15 051个标签对数据集.这些标签对数据集的数量庞大,而其中的一部分标签对数据集对于最终的分类性能影响较小,因此考虑在进行模型构建前使用数据剪枝的方法先删除一些对整体分类性能影响较小的数据集,在保证模型分类性能的同时降低其时空复杂度.

在多标签不均衡数据集中,如果同时携带两个标签的实例数量比较少,即两个标签的共现性比较低,说明一条实例同时属于这两个标签的可能性比较低,只能属于其中一个标签.例如有两个共现性较低的标签A和B,对于一条实例,如果标签A对应的单标签分类模型预测其不属于A标签,则标签B对应的单标签分类模型能够正确预测其属于B标签,因此考虑到单标签分类模型对该实例正确分类的能力已经很强,使用由这两个标签构建的标签对数据集构建模型的话对整体性能不会有太大的影响.因此,为了保证模型性能同时降低整体方法的时空复杂度,计划对那些标签共现性较低的标签对数据集进行剪枝.

针对该问题,本文设计了一个标签共现性度量LCO(Label Co-Occurrence)来体现标签之间的共现程度,其计算公式如式(1)所示.在公式(1)中,Xor表示只携带标签C_i或C_j的实例数量,Or则表示携带标签的实例数量,即携带标签C_i或C_j和同时携带两个标签的实例数量.两者的比值越高,则说明二者的实例数量越接近,同时携带两个标签的实例就越少,故LCO的值越大就说明标签的共现性就越低.因此,只需对LCO数值较高的数据集进行剪枝即可,这样既可以降低时空复杂度又可以保证模型的分类性能:

$$\mathrm{LCO}(C_i,C_j)=\frac{\mathrm{Xor}(C_i,C_j)}{\mathrm{Or}(C_i,C_j)} \tag{1}$$

3.5　二分类模型构建

在安全欠采样和数据剪枝完成后,得到一些二分类数据集.针对这些二分类数据集,一些流行的分类算法可以用来学习这些数据集,以构建二分类模型,如C4.5,SMO和KNN.对于单标签数据集,可以使用构建的二分类模型来预测实例是否属于相应的标签.模型会根据输入的特征,给出一个预测结果,判别实例是否属于对应的标签类别.对于具有标签C_i和C_j的标签对数据集,所构建的二分类模型用于将实例分类为C_i或C_j.

3.6　模型集成

二分类模型构建完成之后,利用单标签数据集和标签对数据集共构建了$k+n$个二分类模型,然后将单标签二分类模型和标签对二分类模型进行最终的模型集成.对于单标签数据集的k个二分类模型,每个模型用于分类实例是否属于相应的标签;而标签对数据集进行数据剪枝后构建的n个二分类模型用于分类实例属于标签C_i还是属于标签C_j.

对于单标签数据集,将二分类模型的结果合并到一个k维的二分数组BS(Bipartition array for Single label datasets)中,如果第i个分类模型预测实例携带标签C_i,则把BS数组中标签C_i的对应值设置为1,否则设置为0.这样便把k个二分类模型的结果集成到了一个数组,通过BS数组获得所有单标签二分类模型的分类结果.

对于标签对数据集,则是将二分类模型的结果集成到一个k维的二分数组BP(Bipartition array for label Pair datasets)中.对于标签C_i,选择所有包含C_i的标签对二分类模型,当超过一半的模型预测实例携带C_i标签时,把BP数组中标签C_i对应值设置为1,否则设置为0.如此,n个标签对二分类模型的结果也集成到了一个k维的BP数组中.

得到BS和BP数组之后,根据式(2)集成得到最终的k维分类结果数组BF(Bipartition array for Final result).对于每个标签C_i,只有当BS和BP中该标签对应位置的值都为1的时候,其在BF中相应位置的值才为1,否则置为0.这样便得到了最终的预测结果BF,BF每个位置的值都对应一个标签,根据相应位置上的值便可得到模型的预测结果:

$$\mathrm{BF}=(\mathrm{BS})\oplus(\mathrm{BP}) \tag{2}$$

4　实验结果与分析

4.1　数据集

在实验中,采用了六个来自不同领域的多标签不平衡数据集,详细的统计信息如表1所示.在表1中,给出了数据集的名称,数据集领域,实例数量,属性数量,标签数量和MeanIR值.需要说明的是,MeanIR值表示相应的多标签数据集的平均不平衡程度,MeanIR值越大,数据集的不平衡程度越高.MeanIR的计算公式如式(3)和式(4)所示:

$$\mathrm{IRLbl}(\lambda)=\frac{\max\limits_{\lambda'\in Y}\left(\sum\limits_{i=1}^{m}h(\lambda',Y_i)\right)}{\sum\limits_{i=1}^{m}h(\lambda,Y_i)},\quad h(\lambda,Y_i)=\begin{cases}1 & \lambda\in Y_i\\ 0 & \lambda\notin Y_i\end{cases} \tag{3}$$

$$\mathrm{MeanIR}=\frac{1}{k}\sum_{\lambda\in Y}\mathrm{IRLbl}(\lambda) \tag{4}$$

式(4)中,λ是一个标签,m是多标签不均衡数据集的实例数量,Y_i是第i个实例的标签集合,IRLbl是多数类标签和标签λ出现次数的比例,其中最频繁标签的IRLbl值为1,其余标签的IRLbl值均大于1,其值越高说明相

应标签的不均衡程度越高，而MeanIR则是多标签数据集中所有标签不均衡程度的平均值，即MeanIR值代表了整个数据集的不均衡程度.

从表1可以看出实验选取了不同类型的数据集，不仅来自不同的应用领域，其实例数量，属性数量和标签数量也有着比较大的差异. 例如，Bibtex数据集的实例数量高达7 395，而CAL500和Birds数据集则只有502和645个实例样本；Bibtex的属性数量最多，达到了1 836个，CAL500则只有68个属性；CAL500有174个标签，Scene只有6个标签. 此外，选用的这6个数据集的不均衡程度也差异较大.

表1 实验数据集统计信息

数据集	领域	实例数量	属性数量	标签数量	MeanIR
Bibtex	text	7 395	1 836	159	12.498 3
Birds	audio	645	260	19	5.407 0
CAL500	music	502	68	174	20.577 8
Enron	text	1 702	1 001	53	73.952 8
Scene	image	2 407	294	6	1.253 8
Yeast	biology	2 417	103	14	7.196 8

4.2 评价指标

为了验证方法的有效性，实验选取了四个通用的多标签不均衡数据分类指标：Micro-averaged Recall，Macro-averaged Recall，Micro-averaged AUC和Macro-averaged AUC. 上述四个指标都是基于标签的性能评价指标. 这些指标首先分别评估每个标签，然后对所有的标签进行平均. 基于标签的指标通过两种方式来进行平均，分别是微平均(micro averaged)和宏平均(macro averaged)，其计算公式如式(5)和式(6)所示：

$$\mathrm{EM}_{\mathrm{micro}}=\mathrm{EM}\left(\sum_{\lambda=1}^{k}\mathrm{TP}_{\lambda},\ \sum_{\lambda=1}^{k}\mathrm{FP}_{\lambda},\ \sum_{\lambda=1}^{k}\mathrm{TN}_{\lambda},\ \sum_{\lambda=1}^{k}\mathrm{FN}_{\lambda}\right) \tag{5}$$

$$\mathrm{EM}_{\mathrm{macro}}=\frac{1}{k}\sum_{\lambda=1}^{k}\mathrm{EM}\left(\mathrm{TP}_{\lambda},\mathrm{FP}_{\lambda},\mathrm{TN}_{\lambda},\mathrm{FN}_{\lambda}\right) \tag{6}$$

其中EM表示任意的二分类器评估指标，例如Recall. TP_{λ}(True Positive)表示正确分类为λ的样本个数，FP_{λ}(False Positive)表示错误分类为λ的样本个数，TN_{λ}(True Negative)表示正确分类为非λ的样本个数，FN_{λ}(False Negative)表示错误分类为非λ样本的个数. Recall的计算公式见式(7)：

$$\mathrm{Recall}=\frac{1}{p}\sum_{i=1}^{p}\frac{\left|Y_i\cap h(x_i)\right|}{\left|Y_i\right|} \tag{7}$$

式(7)中p为测试集样本的数量，Micro-averaged和Macro-averaged Recall的值可以通过结合式(5)~(7)获得. Micro-averaged及Macro-averaged AUC的计算如式(8)及式(9)所示：

$$\mathrm{AUC}_{\mathrm{micro}}=\frac{\left|\left\{\left(x,x',y,y'\right)\mid\ f(x,y)\geqslant f(x',y'),\ (x,y)\in S^{+},\ (x',y')\in S^{-}\right\}\right|}{\left|S^{+}\right|\left|S^{-}\right|} \tag{8}$$

$$\mathrm{AUC}_{\mathrm{macro}}=\frac{1}{q}\sum_{j=1}^{q}\frac{\left|\left\{\left(x,x'\right)\mid\ f(x,y_j)\geqslant f(x',y_j),\ (x,x')\in Z_j\times\overline{Z}_j\right\}\right|}{|Z_j||\overline{Z}_j|} \tag{9}$$

式(8)和式(9)中的$f(x,y)$函数为实值函数，其返回值代表y是样本x正确标签的概率值. 式(8)中的$S^{+}=\{(x_i,y)|y\in Y_i,1\leqslant i\leqslant p\}$，其对应于相关的实例标签对集合，$S^{-}=\{(x_i,y)|y\notin Y_i,1\leqslant i\leqslant p\}$是不相关的实例标签对集合. 式(9)中的$Z_j=\{x_i|y_j\in Y_i,1\leqslant i\leqslant p\}$，该集合为测试集中携带标签$y_j$的实例集合，$\overline{Z}_j=\{x_i|y_j\notin Y_i,1\leqslant i\leqslant p\}$则是测试集中不携带标签$y_j$的实例集合.

4.3 研究问题

为了验证方法的性能，针对ESUS方法提出了三个研究问题并设计了相应的实验方案.

(1)ESUS方法的有效性及稳定性如何？针对这个问题，通过将ESUS方法在六个实验数据集和三个分类算法上的性能结果与MLUL，MLSOL，MLTL，LPROS，LPRUS，MLROS及MLRUS这七个基线方法进行比较来进行验证.

(2)安全欠采样方法的有效性如何？针对该问题，通过消融实验，只对数据集进行标签划分，然后针对单标签数据集进行二分类模型构建，而对于标签对数据集则是先构建二分类模型再进行数据剪枝，将二分类模型集成起来得到最终的分类模型. 最后将不使用安全欠采样方法的性能与ESUS方法性能进行比较，验证安全欠采样的有效性.

(3)数据剪枝的有效性如何？针对数据剪枝的问题，本文进行了消融实验，通过对比标签对数据集进行数据剪枝与不进行数据剪枝的性能，及对算法运行时间进行对比来综合评估数据剪枝的效果.

4.4 实验设置

在实验中，采取了广泛使用的十折交叉验证方法，并使用了三种比较流行的分类算法：C4.5，SMO和KNN. 其中，根据相关研究，KNN算法的k值设定为奇数[38]. 当k取值为1时会使整体模型变得复杂，且容易发生过拟合；k取值太大相当于用较大邻域中的训练数据进行预测，这时与输入实例较远的训练实例也会对预测起作用，可能导致整体预测性能的降低. 考虑到本文方法经过标签划分后得到的单标签数据集和标签对数据集的数量可能会比较多，本文选取k值为3. 此外，为了验证ESUS方法

的性能,实验共选取了五种基线方法和四个常用的分类指标进行比较. 这四种分类指标包括Micro-averaged Recall, Macro-averaged Recall, Micro-averaged AUC 和 Macro-averaged AUC,详细内容已经在4.2节进行了介绍. 此外,选取的七种基线方法包括MLUL,MLSOL,MLTL,LPROS, LPRUS, MLROS以及MLRUS,并且为了比较,MLROS和MLRUS选取了两种不同的采样比例,分别是10%和25%.

4.5 结果分析

(1)ESUS方法的有效性和稳定性如何?

为了验证ESUS方法的有效性和稳定性,本节选用了六个来自不同领域的多标签不均衡数据集,使用了三种分类算法和四个性能评价指标,并与七种多标签不均衡数据集的基线方法进行了比较. 在下文中,表2给出了Micro-averaged Recall指标在三种分类算法下所有方法的结果,表3给出了Macro-averaged Recall指标的结果,表4提供了Micro-averaged AUC指标的结果,表5为Macro-averaged AUC指标的结果. 需要说明的是,在表2~表5中,每个表格中对每个数据集上在给定分类算法下表现最好的方法结果进行了加粗.

表2 Micro-averaged Recall结果

Classification	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
C4.5	ESUS	**0.425 3**	**0.457 8**	0.340 1	**0.528 7**	**0.709 9**	0.517 6
	MLUL	0.329 1	0.356 5	0.279 7	0.460 2	0.611 5	**0.578 4**
	MLSOL	0.335 4	0.362 1	0.305 1	0.469 2	0.636 7	0.562 1
	MLTL	0.229 7	0.316 5	**0.344 7**	0.455 7	0.627 8	0.566 8
	LPROS	0.356 1	0.368 0	0.277 7	0.475 9	0.615 4	0.553 7
	LPRUS	0.331 8	0.346 3	0.277 7	0.463 1	0.619 9	0.575 6
	MLROS-10	0.338 1	0.335 0	0.326 0	0.481 6	0.613 5	0.551 7
	MLROS-25	0.347 2	0.360 6	0.336 7	0.482 3	0.622 5	0.554 3
	MLRUS-10	0.320 5	0.353 5	0.279 2	0.460 0	0.631 7	0.568 4
	MLRUS-25	0.310 1	0.311 5	0.292 1	0.454 0	0.625 2	0.538 1
SMO	ESUS	**0.469 9**	**0.548 4**	**0.234 1**	**0.548 6**	**0.722 2**	0.505 4
	MLUL	0.384 9	0.409 7	0.226 0	0.496 7	0.629 4	0.568 2
	MLSOL	0.394 5	0.441 1	0.232 0	0.503 1	0.657 2	0.577 8
	MLTL	0.194 3	0.414 5	0.270 7	0.480 5	0.640 5	0.570 4
	LPROS	0.396 1	0.458 8	0.226 5	0.502 3	0.660 2	**0.597 1**
	LPRUS	0.380 6	0.428 3	0.226 5	0.499 1	0.639 5	0.579 9
	MLROS-10	0.393 9	0.433 4	0.229 3	0.499 8	0.634 0	0.569 8
	MLROS-25	0.394 3	0.438 9	0.232 1	0.498 4	0.635 9	0.569 1
	MLRUS-10	0.372 8	0.400 3	0.227 4	0.496 7	0.638 1	0.566 4
	MLRUS-25	0.351 7	0.353 1	0.226 6	0.493 3	0.636 6	0.568 2
KNN	ESUS	0.172 6	**0.560 1**	0.327 9	0.350 4	0.674 5	0.557 4
	MLUL	0.153 0	0.471 0	0.304 3	0.315 1	0.679 9	0.608 8
	MLSOL	0.165 3	0.525 4	0.318 9	0.328 2	**0.701 3**	0.617 5
	MLTL	0.074 9	0.385 6	**0.333 2**	0.290 9	0.682 6	0.613 7
	LPROS	**0.202 4**	0.556 8	0.304 9	**0.391 2**	0.695 1	**0.618 3**
	LPRUS	0.148 8	0.462 7	0.304 9	0.335 1	0.675 6	0.608 9
	MLROS-10	0.170 0	0.510 0	0.321 7	0.334 9	0.687 7	0.613 0
	MLROS-25	0.183 2	0.499 6	0.328 0	0.335 0	0.684 6	0.611 4
	MLRUS-10	0.148 6	0.439 0	0.305 1	0.318 6	0.677 6	0.601 3
	MLRUS-25	0.136 8	0.344 5	0.301 6	0.325 3	0.675 3	0.590 3

从表2可以看出,使用C4.5和SMO分类算法时,ESUS方法在绝大多数数据集上都取得了最好的性能. 具体而言,使用C4.5分类算法时,ESUS在4个数据集上取得了最好的Micro-averaged Recall结果;使用SMO分类算法时,ESUS在5个数据集上取得了最好的Micro-averaged Recall结果. 平均而言,使用C4.5算法时,ESUS方法的平均性能

比排名第二的MLSOL提升了12.6%;而使用SMO分类算法时,ESUS方法的平均性能比排名第二的LPROS提升了6.6%.此外,使用KNN分类算法时,ESUS方法仅在Birds数据集上取得了最好的Micro-averaged Recall结果,此时最好的方法是LPROS,而ESUS方法在平均性能上排名第3,与排名第2的MLSOL方法相差了0.001 5,优于其他五种方法.因此在使用Micro-averaged Recall作为性能评价指标时,ESUS方法在处理多标签不均衡数据集时是有效的.

表3 Macro-averaged Recall结果

Classification	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
C4.5	ESUS	**0.340 2**	**0.405 9**	0.242 6	**0.335 1**	**0.720 1**	0.394 2
	MLUL	0.243 5	0.358 9	0.214 4	0.301 5	0.624 8	0.385 2
	MLSOL	0.249 7	0.319 3	0.228 7	0.296 4	0.648 1	0.388 8
	MLTL	0.151 2	0.326 0	**0.248 4**	0.292 0	0.638 5	0.378 8
	LPROS	0.274 2	0.305 1	0.210 1	0.276 2	0.626 6	**0.402 2**
	LPRUS	0.240 6	0.344 6	0.210 1	0.296 3	0.630 4	0.387 7
	MLROS-10	0.254 5	0.331 5	0.216 0	0.264 4	0.627 6	0.395 1
	MLROS-25	0.264 6	0337 1	0.216 0	0.267 7	0.633 9	0.399 0
	MLRUS-10	0.236 9	0.359 3	0.210 3	0.297 5	0.645 3	0.380 2
	MLRUS-25	0.226 3	0.321 3	0.214 6	0.295 9	0.638 6	0.360 3
SMO	ESUS	**0.419 9**	**0.475 0**	**0.335 2**	0.498 7	**0.680 8**	0.591 8
	MLUL	0.295 9	0.398 5	0.177 2	0.320 9	0.642 0	0.320 1
	MLSOL	0.310 9	0.427 1	0.181 5	0.321 4	0.669 1	0.327 8
	MLTL	0.194 3	0.414 5	0.270 7	0.480 5	0.640 5	0.570 4
	LPROS	0.396 1	0.458 8	0.226 5	**0.502 3**	0.660 2	**0.597 1**
	LPRUS	0.380 6	0.428 3	0.226 5	0.499 1	0.639 5	0.579 9
	MLROS-10	0.393 9	0.433 4	0.229 3	0.499 8	0.634 0	0.569 8
	MLROS-25	0.394 3	0.438 9	0.232 1	0.498 4	0.635 9	0.569 1
	MLRUS-10	0.372 8	0.400 3	0.227 4	0.496 7	0.638 1	0.566 4
	MLRUS-25	0.351 7	0.353 1	0.226 6	0.493 3	0.636 6	0.568 2
KNN	ESUS	0.108 8	**0.542 8**	0.236 1	**0.281 0**	0.686 2	0.403 9
	MLUL	0.083 2	0.459 3	0.230 6	0.255 2	0.690 3	0.422 3
	MLSOL	0.097 0	0.516 0	0.236 8	0.269 7	**0.709 9**	0.451 7
	MLTL	0.045 3	0.421 9	**0.244 7**	0.253 7	0.691 8	0.422 8
	LPROS	**0.136 9**	0.511 9	0.231 8	0.275 9	0.703 5	**0.454 8**
	LPRUS	0.083 0	0.466 7	0.231 8	0.263 4	0.683 0	0.419 8
	MLROS-10	0.102 5	0.459 8	0.208 8	0.258 2	0.697 3	0.437 9
	MLROS-25	0.121 8	0.451 9	0.189 4	0.254 4	0.693 7	0.447 2
	MLRUS-10	0.079 4	0.444 8	0.231 0	0.261 7	0.689 0	0.411 8
	MLRUS-25	0.071 2	0.386 3	0.230 2	0.260 3	0.687 0	0.401 5

从表3可以看出,当使用Macro-averaged Recall作为性能评价指标时,使用C4.5和SMO分类算法,ESUS方法能够在绝大多数数据集上取得了最好的分类性能.具体而言,使用C4.5分类算法时,ESUS在4个数据集上取得了最好的Macro-averaged Recall结果;使用SMO分类算法时,ESUS也在4个数据集上取得了最好的Macro-averaged Recall结果.平均而言,在使用C4.5分类算法时,ESUS方法的平均性能比排名第二的MLSOL要提高了14.4%;使用SMO分类算法时,ESUS方法的平均性能则是比排名第二的LPROS提高了6.6%.此外,使用KNN分类算法时,虽然ESUS方法仅在Bird和Enron两个数据集上取得了最好的Macro-averaged Recall结果,整体而言,ESUS方法在平均性能(0.376 5)上略次于最好的LPROS方法(0.385 8)和排名第二的MLSOL方法(0.380 2),但仍优于其他的对比算法.因此在使用Macro-averaged Recall作为性能评价指标时,ESUS方法在处理多标签不均衡数据集时是有效的.

表4 Micro-averaged AUC结果

Classification	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
C4.5	EUS	0.820 2	**0.751 5**	0.652 1	0.811 6	**0.786 4**	0.658 8
	MLUL	0.827 1	0.723 0	**0.695 4**	0.807 2	0.754 9	0.692 1
	MLSOL	0.827 6	0.730 6	0.665 2	0.807 4	0.766 6	0.673 8
	MLTL	0.649 1	0.705 7	0.648 2	0.813 2	0.749 2	0.682 8
	LPROS	0.794 4	0.687 7	0.694 6	0.762 3	0.744 0	0.667 3
	LPRUS	0.822 2	0.740 5	0.694 6	0.795 6	0.759 2	0.678 8
	MLROS-10	**0.834 7**	0.718 5	0.634 3	**0.817 6**	0.744 9	0.671 4
	MLROS-25	0.831 4	0.727 3	0.608 2	0.815 9	0.750 3	**0.685 3**
	MLRUS-10	0.826 4	0.732 6	0.690 0	0.807 8	0.759 0	0.681 2
	MLRUS-25	0.825 9	0.714 1	0.689 9	0.807 7	0.761 9	0.675 5
SMO	ESUS	**0.729 1**	**0.752 7**	0.602 8	**0.752 2**	**0.817 6**	0.734 2
	MLUL	0.689 2	0.697 8	0.600 5	0.733 3	0.789 0	0.734 9
	MLSOL	0.693 6	0.712 9	0.603 0	0.735 8	0.802 6	0.737 7
	MLTL	0.596 6	0.697 8	**0.616 7**	0.729 0	0.794 9	0.735 9
	LPRS	0.694 3	0.718 0	0.601 0	0.734 8	0.799 7	**0.741 4**
	LPUS	0.686 7	0.705 5	0.601 0	0.733 4	0.793 1	0.738 0
	ROS-10	0.693 4	0.709 2	0.602 1	0.734 5	0.792 5	0.735 3
	MLROS-25	0.693 5	0.710 7	0.603 2	0.733 8	0.793 2	0.732 6
	MLRUS-10	0.683 4	0.693 2	0.601 4	0.733 9	0.795 8	0.734 0
	MLRUS-25	0.673 2	0.668 4	0.600 9	0.732 8	0.794 1	0.734 5
KNN	ESUS	**0.697 3**	0.829 1	0.623 0	0.748 6	0.884 6	0.777 8
	MLUL	0.687 3	0.840 2	0.701 4	0.791 1	0.888 6	0.799 2
	MLSOL	0.688 0	**0.841 6**	0.699 0	0.791 6	0.892 4	0.789 7
	MLTL	0.584 5	0.762 3	**0.706 1**	0.744 0	0.891 1	**0.801 8**
	LPROS	0.674 6	0.790 0	0.700 3	0.787 5	0.886 9	0.768 7
	LPRUS	0.682 9	0.834 7	0.700 3	0.791 7	0.887 7	0.793 7
	MLROS-10	0.685 2	0.822 7	0.695 6	0.791 2	**0.893 6**	0.789 6
	MLROS-25	0.680 1	0.799 8	0.682 1	0.788 2	0.891 0	0.775 2
	MLRUS-10	0.685 1	0.833 2	0.701 4	0.789 9	0.887 6	0.798 5
	MLRUS-25	0.678 0	0.799 2	0.699 7	**0.791 9**	0.888 1	0.795 4

表4给出了ESUS方法和对比方法在三个分类算法上的Micro-averaged AUC结果. 从表4中可以看出,使用SMO分类算法时,ESUS方法在4个数据集上都取得了最好的结果,它的平均性能比排名第二的LPROS方法提高了2.3%. 使用C4.5分类算法时,ESUS方法在Birds和Scene两个数据集上取得了最好的Micro-averaged AUC结果,其平均性能(0.746 8)略低于表现最好的MLSOL方法(0.749 9),以及表现相对较好的MLRUS-10方法(0.749 5)和LPRUS方法(0.748 5). 使用KNN分类算法时,ESUS方法仅在Bibtex数据集上取得了最好的结果,在平均性能上也仅仅优于MLTL方法. 由此可见,当使用Micro-averaged AUC作为性能评价指标时,ESUS方法在使用SMO和C4.5分类算法时处理多标签不均衡数据集是比较有效的.

ESUS方法及对比方法在3个分类算法下的Macro-averaged AUC指标结果在表5中进行了展示. 从表5中可以看出,ESUS方法在三个分类算法上表现都非常的好. 具体而言,当使用C4.5分类算法时,ESUS在4个数据集上取得了最好的Macro-averaged AUC结果,其平均性能比排名第二的MLROS-25提升了1.8%;当使用SMO分类算法时,ESUS在5个数据集上取得了最好的Macro-averaged AUC结果,其平均性能比排名第二的LPROS提升了3.1%;当使用KNN分类算法时,ESUS在4个数据集上取得了最好的Macro-averaged AUC结果,其平均性能比排名第二的MLSOL提升了1.3%. 由此可见,当使用Macro-averaged AUC作为性能评价指标时,ESUS方法在处理多标签不均衡数据集是比较有效的.

综上,当使用本文的四种性能评价指标和三种分

表5 Macro-averaged AUC结果

Classification	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
C4.5	ESUS	0.742 5	**0.684 2**	0.508 9	**0.585 9**	**0.795 3**	**0.581 1**
	MLUL	0.734 8	0.656 3	0.511 4	0.567 5	0.764 0	0.576 6
	MLSOL	0.747 6	0.643 4	0.505 2	0.574 8	0.778 1	0.571 9
	MLTL	0.591 9	0.647 3	**0.512 4**	0.572 2	0.762 9	0.576 1
	LPROS	0.727 2	0.605 9	0.505 3	0.571 7	0.756 2	0.569 3
	LPRUS	0.729 6	0.679 9	0.505 3	0.567 4	0.765 3	0.573 0
	MLROS-10	**0.759 5**	0.647 8	0.501 9	0.576 1	0.757 5	0.564 4
	MLROS-25	0.759 2	0.645 4	0.511 8	0.578 6	0.763 0	0.570 2
	MLRUS-10	0.733 4	0.683 1	0.503 8	0.565 6	0.770 0	0.569 0
	MLRUS-25	0.724 8	0.660 4	0.507 6	0.565 9	0.772 8	0.569 6
SMO	ESUS	**0.687 2**	**0.725 3**	0.505 2	**0.619 4**	**0.822 8**	**0.573 9**
	MLUL	0.644 7	0.661 2	0.504 1	0.594 4	0.795 0	0.562 0
	MLSOL	0.651 9	0.679 7	0.505 3	0.599 3	0.808 3	0.563 5
	MLTL	0.555 5	0.674 7	**0.507 1**	0.573 1	0.800 7	0.562 8
	LPROS	0.651 4	0.689 1	0.504 2	0.596 6	0.804 2	0.568 5
	LPRUS	0.641 6	0.678 1	0.504 2	0.595 8	0.797 8	0.564 0
	MLROS-10	0.650 7	0.685 9	0.504 6	0.599 2	0.798 9	0.562 1
	MLROS-25	0.650 6	0.685 3	0.505 5	0.599 1	0.799 5	0.561 6
	MLRUS-10	0.639 7	0.667 7	0.504 7	0.596 4	0.803 5	0.561 4
	MLRUS-25	0.631 1	0.643 1	0.504 6	0.583 9	0.800 1	0.561 4
KNN	ESUS	**0.665 0**	**0.839 8**	**0.525 7**	0.624 0	0.890 0	**0.664 9**
	MLUL	0.643 1	0.826 1	0.517 0	0.616 0	0.888 6	0.658 2
	MLSOL	0.643 7	0.820 7	0.516 2	0.621 6	0.891 7	0.659 0
	MLTL	0.564 7	0.747 4	0.521 9	0.584 1	0.891 1	0.658 5
	LPROS	0.628 4	0.772 4	0.516 0	**0.628 7**	0.887 0	0.640 7
	LPRUS	0.637 8	0.817 3	0.516 0	0.616 5	0.886 3	0.651 7
	MLROS-10	0.640 6	0.807 1	0.518 6	0.617 4	**0.892 7**	0.653 2
	MLROS-25	0.635 5	0.781 5	0.512 8	0.615 0	0.890 1	0.648 9
	MLRUS-10	0.641 1	0.809 4	0.518 3	0.610 6	0.889 2	0.655 7
	MLRUS-25	0.634 9	0.779 1	0.519 7	0.602 8	0.889 2	0.652 4

类算法时，ESUS方法和一些基线方法相比，在处理多标签不均衡数据集时是比较有效的. 为了验证ESUS方法的稳定性，图2~5给出了ESUS及对比方法在3种分类算法下的4个平均性能评价指标.

如图2所示，当使用Micro-averaged Recall指标时，ESUS方法在C4.5和SMO分类算法上都排名第1，在KNN分类算法上排名第3. 如图3所示，当使用Macro-averaged Recall指标时，ESUS方法在C4.5和SMO分类算法上都排名第1，在KNN分类算法上排名第3. 如图4所示，当使用Micro-averaged AUC指标时，ESUS方法在SMO分类算法排名第1，在C4.5分类算法上排名第3，而在KNN分类算法上排名一般. 如图5所示，当使用Macro-averaged AUC指标时，ESUS方法在三种分类算法上都排名第1，明显高于其他对比算法. 然而对于其他基线对比方法，例如LPROS方法，其在Recall指标上的排名还比较靠前，然而在AUC指标上排名就一般了；而LPRUS方法，其在AUC指标上的排名还比较靠前，然而在AUC指标上排名就比较一般了. 由此可见，和选用的基线方法相比，ESUS方法更稳定.

综上所述，在处理多标签不均衡数据集时，本文提出的ESUS方法不仅有效，且和现有的基线方法相比，其更稳定.

(2)安全欠采样方法的有效性如何?

本节对ESUS方法中安全欠采样方法的有效性进行了消融实验，表6~8分别给出了在三种分类算法下ESUS方法使用安全欠采样(sampling)和不使用安全欠采样(no sampling)的消融实验结果.

表6给出了使用C4.5分类算法时进行安全欠采样

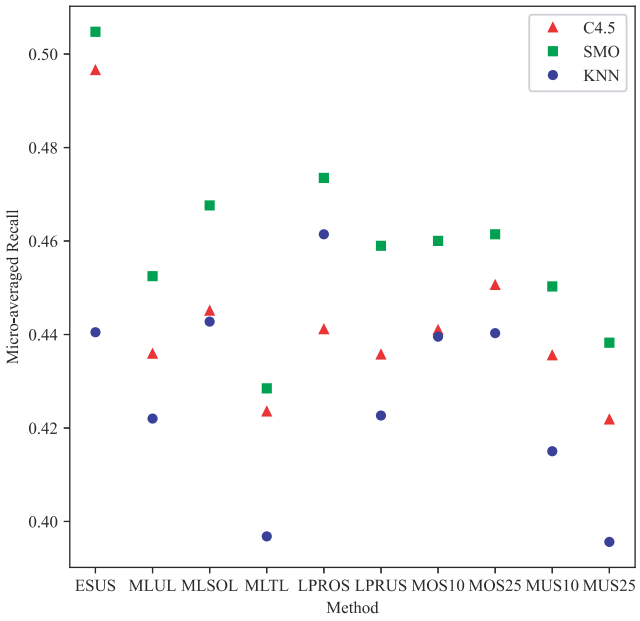

图2　Micro-averaged Recall均值比较图

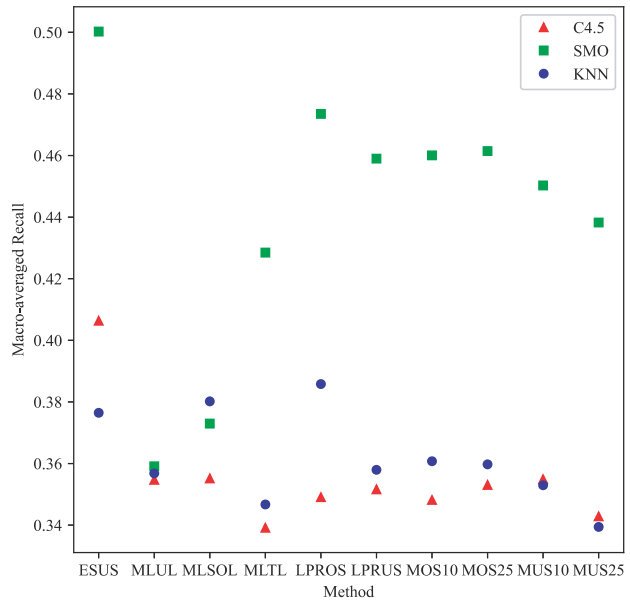

图3　Macro-averaged Recall均值比较图

和不进行安全欠采样的消融实验结果. 从表6可以看出,使用C4.5分类算法时,在宏观和微观Recall上安全欠采样在所有数据集上都取得了最好的结果,平均而言,在Micro-averaged Recall指标上安全欠采样和不进行安全欠采样平均相比提升了14.06%,在Macro-averaged Recall指标上平均提升了15.85%,平均提升比例还是比较大的. 此外,对于Macro-averaged AUC指标,安全欠采样在5个数据集上取得了最佳的结果,其平均性能0.649 7也略好于不进行安全欠采样的0.640 7;而对于Micro-averaged AUC指标来说,尽管仅在2个数据集上取得了最好的结果,其平均性能0.746 8略微次于

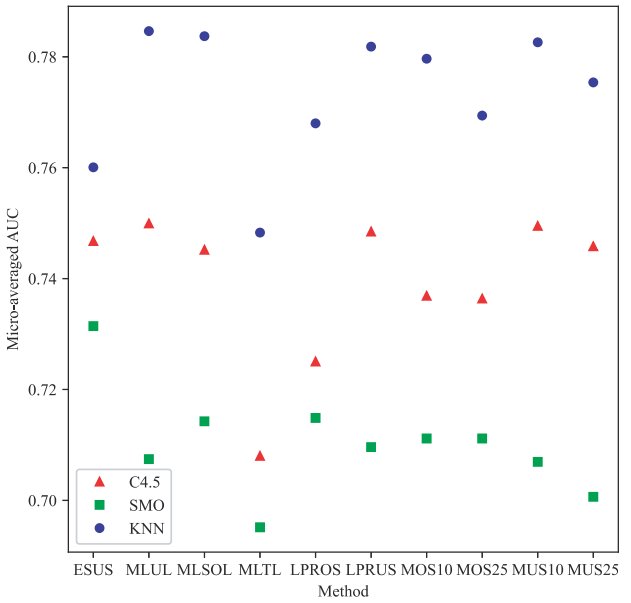

图4　Micro-averaged AUC均值比较图

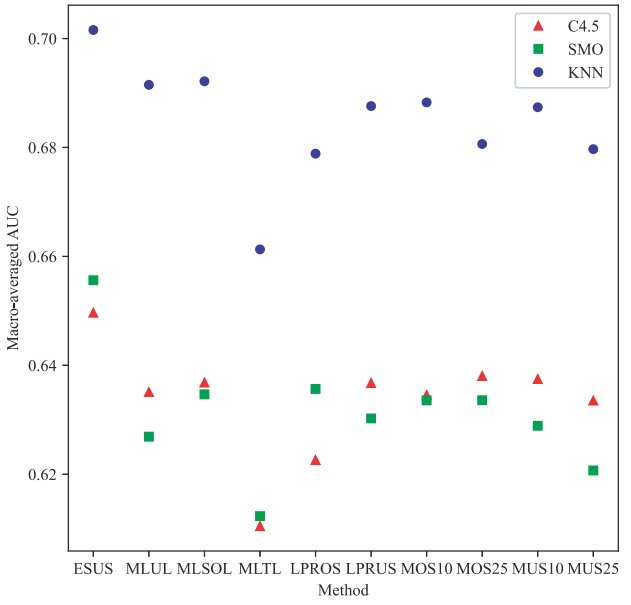

图5　Macro-averaged AUC均值比较图

不使用安全欠采样的0.754 1. 由此可见,当使用C4.5分类算法时,ESUS方法使用安全欠采样是有效的.

表7展示了使用SMO分类算法时采样方法的消融实验结果. 从表中可以看出,在四个评价指标上,安全欠采样方法在几乎所有的数据集上都取得了最优的结果. 只有在使用Micro-averaged AUC指标时,在Yeast这个数据集上有略微的差距,只相差了0.001 7. 因此当使用SMO分类算法时,ESUS方法使用安全欠采样是有效的.

表8提供了使用KNN分类算法时采样方法的消融实验结果. 从表8中可以看出,在Micro-averaged Recall, Macro-averaged Recall和Macro-averaged AUC三个指标

表6 使用C4.5分类算法时采样方法的消融实验结果

Evaluation	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
Micro-averaged Recall	Sampling	**0.425 3**	**0.457 8**	**0.340 1**	**0.528 7**	**0.709 9**	**0.517 6**
	No Sampling	0.333 2	0.378 4	0.277 7	0.484 1	0.622 4	0.516 6
Macro-averaged Recall	Sampling	**0.340 2**	**0.405 9**	**0.242 6**	**0.335 1**	**0.720 1**	**0.394 2**
	No Sampling	0.246 6	0.368 3	0.210 1	0.304 6	0.633 8	0.341 4
Micro-averaged AUC	Sampling	0.820 2	**0.751 5**	0.652 1	0.811 6	**0.786 4**	0.658 8
	No Sampling	**0.831 8**	0.743 5	**0.694 6**	**0.814 8**	0.750 9	**0.688 9**
Macro-averaged AUC	Sampling	0.742 5	**0.684 2**	**0.508 9**	**0.585 9**	**0.795 3**	**0.581 1**
	No Sampling	**0.745 6**	0.676 9	0.505 3	0.575 4	0.763 2	0.577 5

表7 使用SMO分类算法时采样方法的消融实验结果

Evaluation	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
Micro-averaged Recall	Sampling	**0.469 9**	**0.548 4**	**0.234 1**	**0.548 6**	**0.722 2**	**0.505 4**
	No Sampling	0.392 9	0.416 1	0.226 5	0.500 7	0.637 5	0.504 4
Macro-averaged Recall	Sampling	**0.386 0**	**0.479 5**	**0.180 7**	**0.369 1**	**0.733 4**	**0.285 9**
	No Sampling	0.306 7	0.403 2	0.178 1	0.322 5	0.649 8	0.267 9
Micro-averaged AUC	Sampling	**0.729 1**	**0.752 7**	**0.602 8**	**0.752 2**	**0.817 6**	0.734 2
	No Sampling	0.693 1	0.702 0	0.601 0	0.735 1	0.793 9	**0.735 9**
Macro-averaged AUC	Sampling	**0.687 2**	**0.725 3**	**0.505 2**	**0.619 4**	**0.822 8**	**0.573 9**
	No Sampling	0.650 0	0.664 8	0.504 2	0.597 3	0.799 8	0.562 9

表8 使用KNN分类算法时采样方法的消融实验结果

Evaluation	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
Micro-averaged Recall	Sampling	**0.172 6**	**0.560 1**	**0.327 9**	**0.350 4**	0.674 5	**0.557 4**
	No Sampling	0.157 3	0.498 5	0.304 9	0.328 5	**0.682 6**	0.540 0
Macro-averaged Recall	Sampling	**0.108 8**	**0.542 8**	**0.236 1**	**0.281 0**	0.686 2	**0.403 9**
	No Sampling	0.085 5	0.502 0	0.231 8	0.265 0	**0.691 6**	0.367 2
Micro-averaged AUC	Sampling	**0.697 3**	0.829 1	0.623 0	0.748 6	0.884 6	0.777 8
	No Sampling	0.689 7	**0.843 6**	**0.700 3**	**0.795 4**	**0.891 5**	**0.801 3**
Macro-averaged AUC	Sampling	**0.665 0**	**0.839 8**	**0.525 7**	**0.624 0**	0.890 0	**0.664 9**
	No Sampling	0.645 7	0.823 4	0.516 0	0.620 4	**0.891 4**	0.660 5

上,使用安全欠采样方法在5个数据集上都取得最好的结果,仅仅在Scene这个数据集上略微次于没有使用安全欠采样的性能. 例如使用Macro-averaged AUC指标时,安全欠采样的性能结果是0.89,不使用安全欠采样的结果是0.891 4,两者相差很小. 此外,对于Micro-averaged AUC指标而言,使用安全欠采样仅仅在1个数据集上取得了最佳的结果. 整体而言,当使用KNN分类算法时,ESUS方法使用安全欠采样是有效的.

综上所述,在处理多标签不均衡数据集时,ESUS方法中的安全欠采样是有效的.

(3)数据剪枝的有效性如何?

数据剪枝能一定程度上会影响分类性能,并可以减少时空复杂度. 因此对ESUS方法中的数据剪枝进行了消融实验. 消融实验的结果如表9~12所示. 表9~11分别为使用C4.5,SMO和KNN分类算法时数据剪枝的消融实验结果,这3个表中对性能发生变化且表现更好的结果进行加粗. 此外,表12为使用三种分类算法时的时间对比结果,表格中对于每个数据集上表现最好方法的结果进行了加粗.

表9展示了使用C4. 5分类算法时数据剪枝的消融实验结果. 从表中数据可以看出,尽管我们删除了一部分标签对数据集,但是对于Recall评价指标来说,数据剪枝甚至在一些数据集上取得了比不进行数据剪枝方法更好的结果,整体上变化不大,这恰好印证了有些标签对数据集对最终分类结果影响较小. 与此同时,AUC结果没有改变说明了删除的标签对数据集没有对最后的分类结果产生影响. 因此在使用C4.5分类算法时,ESUS方法使用数据剪枝是有效的.

表10为使用SMO分类算法时数据剪枝的消融实验结果. 从表中可以看出,在使用SMO分类算法时,在

表9　使用C4.5分类算法时数据剪枝的消融实验结果

Evaluation	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
Micro-averaged Recall	Data Pruning	**0.425 3**	**0.457 8**	**0.340 1**	0.528 7	**0.709 9**	0.517 6
	No Pruning	0.425 0	0.454 6	0.334 3	0.528 7	0.699 9	**0.525 0**
Macro-averaged Recall	Data Pruning	**0.340 2**	0.405 9	0.242 6	0.335 1	**0.720 1**	**0.394 2**
	No Pruning	0.339 8	**0.413 7**	**0.252 1**	**0.336 9**	0.709 6	0.356 2
Micro-averaged AUC	Data Pruning	0.820 2	0.751 5	0.652 1	0.811 6	0.786 4	0.658 8
	No Pruning	0.820 2	0.751 5	0.652 1	0.811 6	0.786 4	0.658 8
Macro-averaged AUC	Data Pruning	0.742 5	0.684 2	0.508 9	0.585 9	0.795 3	0.581 1
	No Pruning	0.742 5	0.684 2	0.508 9	0.585 9	0.795 3	0.581 1

表10　使用SMO分类算法时数据剪枝的消融实验结果

Evaluation	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
Micro-averaged Recall	Data Pruning	0.469 9	**0.548 4**	0.234 1	0.548 6	**0.722 2**	0.505 4
	No Pruning	0.469 9	0.545 1	0.234 1	0.548 6	0.720 3	**0.570 9**
Macro-averaged Recall	Data Pruning	0.386 0	0.479 5	0.180 7	0.369 1	**0.733 4**	0.285 9
	No Pruning	0.386 0	**0.483 0**	**0.181 3**	0.369 1	0.731 4	**0.339 6**
Micro-averaged AUC	Data Pruning	0.729 1	0.752 7	0.602 8	0.752 2	0.817 6	0.734 2
	No Pruning	0.729 1	0.752 7	0.602 8	0.752 2	0.817 6	0.734 2
Macro-averaged AUC	Data Pruning	0.687 2	0.725 3	0.505 2	0.619 4	0.822 8	0.573 9
	No Pruning	0.687 2	0.725 3	0.505 2	0.619 4	0.822 8	0.573 9

Recall指标上,除了Yeast数据集,在其他数据集上的性能变化不大;而在AUC指标上,性能没有任何变化,因此在使用SMO分类算法时,ESUS方法使用数据剪枝是有效的.

表11　使用KNN分类算法时数据剪枝的消融实验结果

Evaluation	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
Micro-averaged Recall	Data Pruning	0.172 6	**0.560 1**	**0.327 9**	0.350 4	0.674 5	0.557 4
	No Pruning	0.172 6	0.558 6	0.327 7	0.350 4	0.674 5	**0.617 1**
Macro-averaged Recall	Data Pruning	0.108 8	**0.542 8**	**0.236 1**	0.281 0	0.686 2	0.403 9
	No Pruning	0.108 8	0.541 1	0.235 2	0.281 0	0.686 2	**0.439 0**
Micro-averaged AUC	Data Pruning	0.697 3	0.829 1	0.623 0	0.748 6	0.884 6	0.777 8
	No Pruning	0.697 3	0.829 1	0.623 0	0.748 6	0.884 6	0.777 8
Macro-averaged AUC	Data Pruning	0.665 0	0.839 8	0.525 7	0.624 0	0.890 0	0.664 9
	No Pruning	0.665 0	0.839 8	0.525 7	0.624 0	0.890 0	0.664 9

表11给出了使用KNN分类算法时数据剪枝的消融实验结果. 在表11中,针对两个Recall指标,Yeast数据集有性能差异,而在其他数据集上基本没差异;针对两个AUC指标,剪枝前后性能不变,因此在KNN分类算法下,ESUS方法使用数据剪枝是有效的.

从表12可以看出进行数据剪枝后,算法运行时间都是减少得. 具体而言,使用C4.5分类算法时,对于CAL500和Enron这种不均衡程度很高的数据集,运行时间减少了600 s多;值得注意的是,对于Bibtex这种不均衡程度相对较大的大样本数据集,时间下降地非常多,下降了有7 500 s左右,这对于效率的提高是非常明显的,所以使用C4.5分类算法时数据剪枝的效果是很显著的. 使用SMO分类算法时,从表中可以看出时间也都是下降的,对于Bibtex数据集,节省了25 000 s,这是非常显著的提升,所以使用SMO分类算法时,数据剪枝对于大样本不均衡数据集性能的提升是巨大的;使用KNN分类算法时,Enron和Bibtex这种不均衡程度较高的大样本数据集时间下降的依旧很多,Enron节省了1 000 s多,而Bibtex则是节省了更多了时间,将近50 000 s的时间,所以数据剪枝的效果十分显著.

综上所述,ESUS方法使用数据剪枝和不进行数据剪枝相比,分类性能上整体没有太大的变化,但时间复杂度下降了许多,因此在ESUS方法中数据剪枝是有效的.

表 12 时间对比结果

Classification	Method	Bibtex	Birds	CAL500	Enron	Scene	Yeast
C4.5	Data Pruning	**90 065**	**75**	**426**	**2 156**	**317**	**274**
	No Pruning	97 547	87	1 023	2 837	344	355
SMO	Data Pruning	**85 243**	**67**	**174**	**1 776**	**299**	**186**
	No Pruning	333 063	74	368	2 155	304	198
KNN	Data Pruning	**93 192**	**70**	**116**	**1 699**	**304**	**183**
	No Pruning	140 817	85	658	2 738	362	349

5 总结

在不均衡多标签数据分类中,现有的采样方法不能同时解决标签内不均衡问题和标签间不均衡问题.本文提出了一种安全欠采样方法解决标签内不均衡问题,利用安全样本来训练模型,提高了模型的分类性能.利用数据剪枝和集成学习解决标签间不均衡问题,既保持分类性能又降低时空复杂度.实验结果表明,本文方法比七种对比方法更稳定有效.

然而,本文方法也存在一些不足之处.在数据剪枝中,不同多标签不均衡数据集的剪枝阈值可能不同,因此计划探索根据数据集的分布情况实现自动化设置阈值.此外,本文方法在KNN分类算法下性能一般,这可能与选择的k值有关,未来考虑研究对KNN分类算法进行超参数优化,进一步提升多标签分类性能.

参考文献

[1] BHATTACHARYA S, RAJAN V, SHRIVASTAVA H. ICU mortality prediction: A classification algorithm for imbalanced datasets[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI, 2017: 1288-1294.

[2] ZHONG W C, RAAHEMI B, LIU J. Classifying peer-to-peer applications using imbalanced concept-adapting very fast decision tree on IP data stream[J]. Peer-to-Peer Networking and Applications, 2013, 6(3): 233-246.

[3] ZAKARYAZAD A, DUMAN E. A profit-driven artificial neural network (ANN) with applications to fraud detection and direct marketing[J]. Neurocomputing, 2016, 175: 121-131.

[4] ZHU Y, KWOK J T, ZHOU Z H. Multi-label learning with global and local label correlation[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(6): 1081-1094.

[5] ZHANG M L, ZHOU Z H. A review on multi-label learning algorithms[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(8): 1819-1837.

[6] TSOUMAKAS G, VLAHAVAS I. Random k-labelsets: an ensemble method for multilabel classification[C]//European Conference on Machine Learning. Berlin: Springer, 2007: 406-417.

[7] ALMEIDA T B, BORGES H B. An Adaptation of the ML-kNN algorithm to predict the number of classes in hierarchical multi-label classification[M]//Modeling Decisions for Artificial Intelligence. Cham: Springer International Publishing, 2017: 77-88.

[8] CHEN L J, FU Y G, CHEN N N, et al. Rule reduction for ebrb classification based on clustering[C]//International Conference on Web Information Systems and Applications. Berlin: Springer, 2021: 442-454.

[9] ELISSEEFF A, WESTON J. A kernel method for multi-labelled classification[C]//Advances in Neural Information Processing Systems. British Columbia: MIT Press, 2001: 681-688.

[10] ZHANG M L, ZHOU Z H. A k-nearest neighbor based algorithm for multi-label classification[C]//IEEE International Conference on Granular Computing. Piscataway: IEEE, 2005: 718-721.

[11] BOUTELL M R, LUO J B, SHEN X P, et al. Learning multi-label scene classification[J]. Pattern Recognition, 2004, 37(9): 1757-1771.

[12] TSOUMAKAS G, KATAKIS I, VLAHAVAS I. Random k-labelsets for multilabel classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(7): 1079-1089.

[13] READ J, PFAHRINGER B, HOLMES G, et al. Classifier chains for multi-label classification[J]. Machine Learning, 2011, 85(3): 333-359.

[14] YU G X, DOMENICONI C, RANGWALA H, et al. Transductive multi-label ensemble classification for protein function prediction[C]//Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2012: 1077-1085.

[15] ZHANG W B, PINCUS Z. Predicting all-cause mortality from basic physiology in the Framingham Heart Study[J]. Aging Cell, 2016, 15(1): 39-48.

[16] 胡峰, 王蕾, 周耀. 基于三支决策的不平衡数据过采样方法[J]. 电子学报, 2018, 46(1): 135-144.
HU F, WANG L, ZHOU Y. An oversampling method for imbalance data based on three-way decision model[J]. Acta Electronica Sinica, 2018, 46(1): 135-144. (in Chinese)

[17] 张艳梅, 植胜林, 姜淑娟, 等. 类不平衡对软件缺陷预测模型稳定性和预测性能的影响分析方法[J]. 电子学报, 2023, 51(8): 2076-2087.
ZHANG Y M, ZHI S L, JIANG S J, et al. Influence analysis method of class imbalance on software defect prediction model stability and prediction performance[J]. Acta Electronica Sinica, 2023, 51(8): 2076-2087. (in Chinese)

[18] GUZMÁN-PONCE A, VALDOVINOS R M, SÁNCHEZ J S, et al. A new under-sampling method to face class overlap and imbalance[J]. Applied Sciences, 2020, 10 (15): 5164.

[19] TAREKEGN A N, GIACOBINI M, MICHALAK K. A review of methods for imbalanced multi-label classification[J]. Pattern Recognition, 2021, 118: 107965.

[20] CHARTE F, RIVERA A, DEL JESUS M J, et al. Resampling multilabel datasets by decoupling highly imbalanced labels[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015: 489-501.

[21] CHARTE F, RIVERA A J, DEL JESUS M J, et al. MLeNN: A first approach to heuristic multilabel undersampling[M]//Intelligent Data Engineering and Automated Learning —IDEAL 2014. Cham: Springer International Publishing, 2014: 1-9.

[22] CHARTE F, RIVERA A J, DEL JESUS M J, et al. Addressing imbalance in multilabel classification: Measures and random resampling algorithms[J]. Neurocomputing, 2015, 163: 3-16.

[23] GUO H X, LI Y J, SHANG J, et al. Learning from class-imbalanced data: Review of methods and applications[J]. Expert Systems with Applications, 2017, 73: 220-239.

[24] CHARTE F, RIVERA A, DEL JESUS M J, et al. A first approach to deal with imbalance in multi-label datasets [C]//International Conference on Hybrid Artificial Intelligence Systems. Salamanca: Springer, 2013: 150-160.

[25] PEREIRA R M, COSTA Y M G, SILLA C N Jr. MLTL: A multi-label approach for the Tomek link undersampling algorithm[J]. Neurocomputing, 2020, 383: 95-105.

[26] BATISTA G E A P A, PRATI R C, MONARD M C. A study of the behavior of several methods for balancing machine learning training data[J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 20-29.

[27] WILSON D L. Asymptotic properties of nearest neighbor rules using edited data[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1972, SMC-2(3): 408-421.

[28] 陈旭, 刘鹏鹤, 孙毓忠, 等. 面向不均衡医学数据集的疾病预测模型研究[J]. 计算机学报, 2019, 42(3): 596-609.
CHEN X, LIU P H, SUN Y Z, et al. Research on disease prediction models based on imbalanced medical data sets [J]. Chinese Journal of Computers, 2019, 42(3): 596-609. (in Chinese)

[29] LIU B, BLEKAS K, TSOUMAKAS G. Multi-label sampling based on local label imbalance[J]. Pattern Recognition, 2022, 122: 108294.

[30] CHAWLA N V, BOWYER K W, HALL L O, et al. SMOTE: Synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research, 2002, 16: 321-357.

[31] CHARTE F, RIVERA A J, DEL JESUS M J, et al. MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation[J]. Knowledge-Based Systems, 2015, 89: 385-397.

[32] MISHRA N K, SINGH P K. Feature construction and smote-based imbalance handling for multi-label learning [J]. Information Sciences, 2021, 563: 342-357.

[33] SADHUKHAN P, PALIT S. Reverse-nearest neighborhood based oversampling for imbalanced, multi-label datasets[J]. Pattern Recognition Letters, 2019, 125: 813-820.

[34] JO W, KIM D. OBGAN: Minority oversampling near borderline with generative adversarial networks[J]. Expert Systems with Applications, 2022, 197: 116694.

[35] ZHANG K, MAO Z Y, CAO P, et al. Label correlation guided borderline oversampling for imbalanced multi-label data learning[J]. Knowledge-Based Systems, 2023, 279: 110938.

[36] ZHU T, LIU X, ZHU E. Oversampling with reliably expanding minority class regions for imbalanced data learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(6): 6167-6181.

[37] ARTHUR D, VASSILVITSKII S. k-means++: The advantages of careful seeding[C]//Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. New Orleans: SIAM, 2007: 1027-1035.

[38] BOATENG E Y, OTOO J, ABAYE D A. Basic tenets of classification algorithms k-nearest-neighbor, support vector machine, random forest and neural network: A review [J]. Journal of Data Analysis and Information Processing, 2020, 8(4): 341-357. .

作者简介

孙中彬 男,1990 年 9 月出生于山东省枣庄市. 现为中国矿业大学计算机科学与技术学院准聘副教授、硕士生导师. 主要研究方向为不均衡数据挖掘、目标检测、工业视觉异常检测. 在国内外发表学术论文 10 余篇.

E-mail: zhongbin@cumt.edu.cn

刁宇轩 男,1998 年 3 月出生于江苏省宿迁市. 现为中国矿业大学计算机科学与技术学院研究生. 主要研究方向为多标签不均衡数据挖掘.

E-mail: TS21170060P31@cumt.edu.cn

马苏洋 男,2004 年 9 月出生于贵州省贵阳市. 现为中国矿业大学计算机科学与技术学院本科生. 主要研究方向为不均衡数据挖掘.

E-mail: 08222744@cumt.edu.cn