计算机科学
COMPUTER SCIENCE

基于构造性神经网络与全局密度信息的不平衡数据欠采样方法

严远亭，马迎澳，任艳平，张燕平

引用本文

严远亭，马迎澳，任艳平，张燕平. 基于构造性神经网络与全局密度信息的不平衡数据欠采样方法[J]. 计算机科学，2023，50(10)：48-58.

YAN Yuanting, MA Yingao, REN Yanping, ZHANG Yanping. Imbalanced Undersampling Based on Constructive Neural Network and Global Density Information [J]. Computer Science, 2023, 50(10): 48-58.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230600022

基于构造性神经网络与全局密度信息的不平衡数据欠采样方法

严远亭　马迎澳　任艳平　张燕平
安徽大学计算机科学与技术学院　合肥 230601

摘　要　多数类欠采样是当前数据层面解决不平衡数据学习的主流技术之一，近年来，研究者们提出了一系列的欠采样方法，但大多都将重点放在如何选择代表性的样本，从而降低信息损失。然而，如何在欠采样过程中保持多数类内部的结构信息，仍然是欠采样面临的主要挑战。针对该挑战，提出了一种基于构造性神经网络和全局分布密度的不平衡数据集欠采样方法。该方法首先基于构造性神经网络，设计了一种多数类局部模式的学习方法；然后基于多数类局部模式，设计了两种具有结构保持特性的样本选择策略；最后针对局部模式学习的随机性可能导致的采样结果非优的问题，进一步引入了 bagging 集成策略，提升了方法的性能。在 59 个数据集上与 13 种对比方法进行了对比实验，验证了所提方法在 G-mean，AUC 和 F1-score 这 3 个常用指标上的有效性。

关键词：欠采样；不平衡数据；分布密度；构造性神经网络；集成学习
中图法分类号　TP311

Imbalanced Undersampling Based on Constructive Neural Network and Global Density Information

YAN Yuanting，MA Yingao，REN Yanping and ZHANG Yanping
College of Computer Science and Technology，Anhui University，Hefei 230601，China

Abstract　Undersampling is one of the mainstream data-level technologies to deal with imbalanced data. In recent years，researchers have proposed numerous undersampling methods，but most of them focus on how to select representative majority class samples to avoid the loss of informative data. However，how to maintain the structures of the original majority class in the process of undersampling is still an open challenge. To this end，an undersampling method for imbalanced data classification is proposed based on constructive neural network and data density. Firstly，it detects the majority local patterns with a simplified constructive process. Then，two sample selection strategies are designed to maintain the structure of the selected groups according to the original majority distribution information. Finally，to solve the problem that the randomness of local pattern learning may lead to non-optimal sampling results，the bagging technique is introduced to further improve the learning performance. Comparative experiments with 13 comparison methodson 59 datasets verify the effectiveness of the proposed method in terms of three metrics G-mean，AUC and F1-score.

Keywords　Undersampling，Imbalanced data，Distribution density，Constructive neural network，Ensemble learning

1　引言

不平衡学习是机器学习、数据挖掘等领域的重要研究内容。不平衡数据的存在给许多经典分类方法带来了挑战，因为传统的分类方法大多是针对平衡数据而设计的，以优化整体分类误差为学习目标。但是，由于不平衡数据中多数类样本与少数类样本数量差异较大，传统分类方法会过分强调多数类，从而出现学习偏置问题。另一方面，在许多实际应用场景中少数类往往更值得关注，例如在医疗诊断[1]、故障诊断[2]、欺诈检测[3]等领域，少数类样本具有更高的误分类代价。

为应对数据不平衡带来的挑战，研究者们从不同视角提出了大量的方法，这些方法大致可分为两类：数据层面方法和算法层面方法[4]。数据层面方法通过对不平衡数据进行重采样，来得到一个平衡的数据分布，并将其直接应用于多数传统分类器；算法层面方法通过修改现有分类器或设计新的分类器，来提升对少数类的识别能力。数据层面方法因其简单有效且独立于后续分类模型，已逐渐成为不平衡学习的主流方法[5]。数据层面方法主要包含过采样方法和欠采样方法[6]，前者通过合成少数类样本的方式来平衡数据集，后者则通过

到稿日期：2023-06-02　返修日期：2023-08-08
基金项目：国家自然科学基金(61806002)
This work was supported by the National Natural Science Foundation of China(61806002).
通信作者：严远亭(ytyan@ahu.edu.cn)

选择代表性的多数类子集的方式来平衡数据分布。然而，这两种方法在实际应用中均存在一些缺陷[7]，例如，过采样方法在大量合成样本的过程中，不仅会耗费更多的计算资源，还会存在过拟合的风险。虽然欠采样方法的训练时间更短，但其仍存在丢失高信息量样本的风险。

文献[8]的研究表明，相比过采样方法，欠采样方法往往有较好的性能。此外，欠采样方法用于训练分类模型的数据均来自于原始数据集，数据的真实性提升了分类模型的可靠性。但是欠采样平衡数据集的方式可能会忽略掉一些重要的多数类样本，进而导致后续分类性能不佳。近年来，出现了许多集成学习与欠采样方法相结合的方法，此类方法已被证明是一种能够有效提升分类模型性能的方法[9]。然而，如何在欠采样时保持多数类的类内结构，仍然是不平衡数据欠采样面临的一大挑战。

针对这一问题，本文提出了一种基于构造性神经网络与全局密度的不平衡数据欠采样方法（Constructive Neural Network and Global Density Based Under Sampling，CDUS）。CDUS 利用构造性神经网络学习样本的局部分布信息，结合样本全局密度信息，针对欠采样的类内结构保持问题，设计了两种不同的样本采样策略 CDUS1 和 CDUS2。CDUS1 考虑到内部样本分布的非均匀性，试图为局部邻域内的每个区域提供相似的优先级。CDUS2 则进一步对样本局部模式进行细分，得到 4 个不同的子区域，通过保证子区域间样本选择的概率，来保持多数类的类内结构。本文的主要贡献如下：

1）提出了基于构造性神经网络和全局密度信息的不平衡数据欠采样方法。

2）提出了两种针对欠采样过程中样本类内结构的保持问题的样本采样策略。

3）在 59 个不平衡数据集上与 13 种方法进行对比，验证了所提方法的有效性。

2 相关工作

近年来，大量数据层面的方法被提出，用于解决不平衡学习的问题。其中最具代表性的是 SMOTE[10]，该方法以线性插值的方式在少数类样本及其近邻样本间合成样本来平衡数据分布，但其合成样本的盲目性可能会引入噪声样本等异常样本，影响后续的分类性能。

为了弥补 SMOTE 的不足，研究者们提出了许多基于 SMOTE 的改进方法。Borderline-SMOTE[11] 注重增强边界区域的少数类样本，其通过 k 近邻挖掘边界样本，并通过在边界附近合成新的样本，来强化分类器对分类边界的学习能力。ADASYN[12] 主要通过自适应的样本困难度度量和采样权重分配，强调对较难学习的少数类样本的学习能力。MWMOTE[13] 利用聚类方法，挖掘样本的分布信息，并基于聚类提取难学习样本，通过对难学习样本进行过采样，并将合成样本控制在聚类内部的方法，提升了模型对易被误分类的少数类样本的学习能力。Safe-Level-Smote[14] 着重强调在安全区域进行样本的合成，避免引入异常的难学习样本，从而提升分类性能。针对 SMOTE 合成方式的不足，GDO[4] 以密度和距离信息对少数类样本进行加权，并利用高斯分布模型提出了一种新的样本合成策略，通过高斯模型实现对合成样本位置的控制。

在欠采样方面，RBU[15] 基于高斯径向基函数设计了 mutual potential 的概念，用于度量样本的分布特征，并依据样本的 mutual potential 值来实现多数类样本的欠采样。此外，基于聚类的欠采样方法也是一类典型代表，文献[16]首先对多数类样本进行 K-Means 聚类，并指定聚类个数为少数类样本的数量，同时将各个簇的聚类中心作为多数类样本与少数类样本组成平衡数据集（CU1）。文献[5]则在多数类样本集中选择各个簇中距离聚类中心最近的多数类样本，实现多数类的欠采样（CU2）。研究表明，在欠采样中引入集成学习技术，能够很好地提升学习性能。经典的方法如下：1）基于 bagging 的欠采样，例如 UnderBagging[17] 通过多次从多数类样本中随机选取（有放回）与少数类样本数量相等的样本，并分别与少数类样本组合成平衡数据集，来训练得到多个分类模型；2）基于 boosting 的欠采样集成方法，例如 Liu 等提出的 EasyEnsemble 和 BalancedCasecade[18]，前者将多数类样本随机划分为几个具有相同规模的不重叠子集，并将这些子集分别与少数类结合，再使用 Adaboost 训练基分类器，后者以随机选取多数类子集的方式训练 Adaboost 基分类器，然后通过调整阈值来保留分类错误的样本并移除分类正确的样本。AdaC2[19] 通过优化 Adaboost 迭代更新中的权值更新方法，来提升集成模型的性能。

除上述将欠采样方法和集成学习结合的方法外，Zhu 等提出了 GSE[20] 的集成方法。具体来说，该方法属于算法层面的方法，其基于欧氏距离，通过学习仅包含多数类样本的超球面和其对应的分类面来构建一个分类器，然后通过删除超球面内的样本并迭代地进行上述过程，得到一组分类器。针对重叠区域，GSE 设计了两种松弛技术来提升模型的泛化能力。GSE 从空间几何结构的视角引入集成策略，且其每个基分类器均可识别所有少数类样本和部分多数类样本的特性，为集成分类模型的可靠性提供了保障。

3 本文方法

本节主要介绍了本文提出的 CDUS 方法。CDUS 主要包括 3 个步骤：首先，通过一个有监督的构造过程来学习输入数据空间的一组多数类超球面邻域；然后，融合密度信息选择具有代表性的多数类样本子集（两种选择策略）；最后，针对邻域学习的随机性问题，通过 bagging 的集成策略，训练一组集成分类器。本文算法的框架如图 1 所示。

为了更好地描述本文的后续内容，本文给出以下符号定义：$D=\{x_1,x_2,\cdots,x_N\}$ 表示不平衡数据集，N 为数据集总数，$\forall x_i\in D(1\leqslant i\leqslant N)$ 表示 m 维的样本；D_{maj} 表示多数类样本集合，$D_{\min}$ 表示少数类样本集合，即 $D=D_{\text{maj}}\cup D_{\min}$；$N_{\text{maj}}$ 为多数类样本数量，$N_{\min}$ 为少数类样本数量，即 $N=N_{\text{maj}}+N_{\min}$；$S=\{s_1,s_2,\cdots,s_v\}$，$S$ 表示多数类局部邻域集合，$\forall s_u\in S(1\leqslant u\leqslant v)$ 表示一个局部模式（局部邻域）。

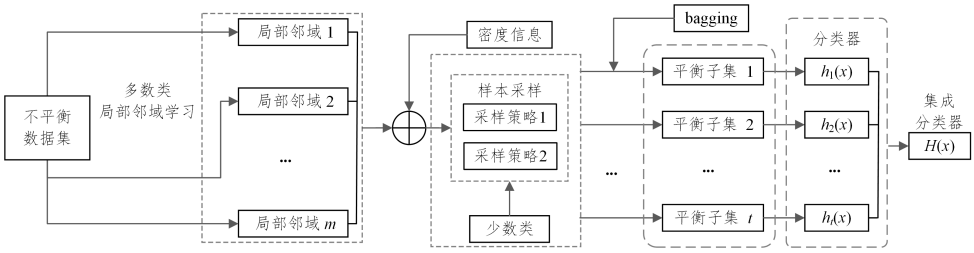


图 1 CDUS 算法的框架

Fig. 1 Framework of the proposed CDUS

3.1 局部邻域的检测

构造性神经网络的特点是网络的结构不需要事先指定，而是通过学习过程自动确定。CCA 是构造性神经网络的一种，其通过不断构造样本的超球面，来得到一组神经元[21]。本文与 CCA 不同，CCA 为了得到模型的最佳性能，需要设计合适的参数，而本文侧重于利用其构造局部邻域（超球）学习当前数据局部分布信息的能力，挖掘不平衡数据中多数类的局部信息，并基于局部信息设计结构保持的多数类采样策略。换句话说，本文关注的是多数类中的哪些样本构成了一个局部邻域。

局部邻域 $S=\{s_1,s_2,\cdots,s_v\}$ 的构造过程大致可分为以下 3 个步骤：

Step1 随机选取一个未被标记的多数类样本 x_k 作为邻域中心，并计算邻域半径（$i,k\in\{1,2,\cdots,N\}$）。

$$\| x_k-x_i \|_2=\sqrt{\sum_{J=1}^{m}|x_k^J-x_i^J|^2},i\neq k \tag{1}$$

$$d_1=\min\{\| x_k-x_i \|_2\},y_k\neq y_i \tag{2}$$

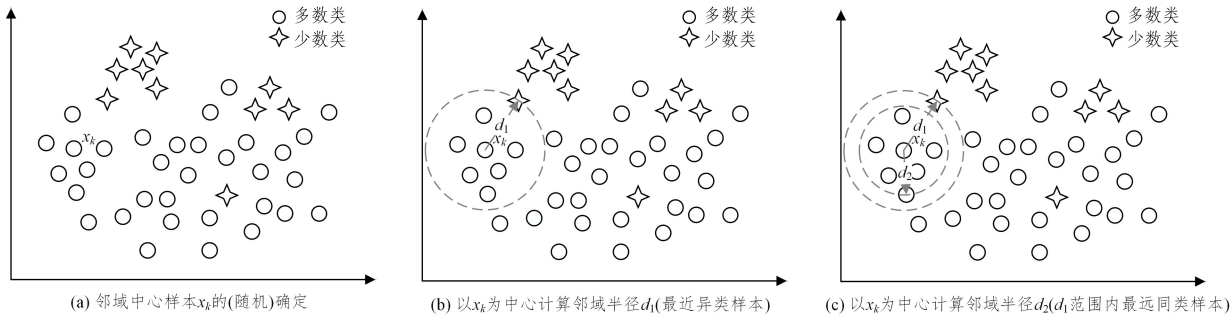
$$d_2=\max\{\| x_k-x_i \|_2 \mid \| x_k-x_i \|_2<d_1\},y_k=y_i \tag{3}$$

其中，y 表示样本类别；$\| x_k-x_i \|_2$ 表示样本 x_k 与 x_i 之间的欧氏距离；d_1 表示 x_k 与异类样本 x_i 间的最小距离，d_2 表示在 d_1 的约束下，x_k 与同类样本 x_i 间的最大距离，在 x_k 为中心、d_2 为半径的范围内，均为多数类样本。

Step2 以 x_k 为中心、d_2 为半径，构造一个局部邻域 s_k，由式(1)将与样本 x_k 距离小于或等于 d_2 的同类样本放入 s_k 中，并将该邻域内的样本标记为“已学习”。

Step3 若数据集中存在未被标记的样本，则重复步骤 1 和步骤 2，继续构造局部邻域直到所有多数类样本都被标记为“已学习”。最终得到邻域集合 $S=\{s_1,s_2,\cdots,s_v\}$。

局部邻域构造过程如图 2(a)－图 2(c)所示。需要指出的是，若 x_k 为邻域中心样本且其最近样本为异类样本，则以 $d_1/2$ 为球面邻域 s_k 的半径。



(a) 邻域中心样本 x_k 的(随机)确定　(b) 以 x_k 为中心计算邻域半径 d_1(最近异类样本)　(c) 以 x_k 为中心计算邻域半径 d_2(d_1 范围内最远同类样本)

图 2 局部邻域学习示意图

Fig. 2 Sketch map of the construction process of local neighborhood

3.2 多数类欠采样

通过上述局部邻域检测方法可以构建局部邻域来获取多数类的局部分布特征。样本邻域的具体信息如图 3 所示。

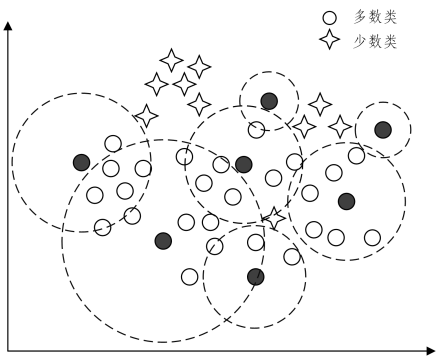


图 3 多数类邻域示意图

Fig. 3 Sketch map of majority class neighborhood

图 3 中实心圆为邻域中心样本，每一个邻域表示多数类的一个局部模式，考虑到邻域学习的先后关系，邻域重叠区域的样本属于先学习到的邻域。

假设多数类局部模式集合为 $S=\{s_1,s_2,\cdots,s_v\}$，对 $\forall s_u\in S(1\leqslant u\leqslant v))$，$s_u$ 内选择的样本数量 Num_{s_u} 由式(4)计算得到。

$$Num\,s_u=\frac{|s_u|}{N_{\text{maj}}}*N_{\min} \tag{4}$$

其中，$|s_u|$ 表示局部邻域 s_u 内的样本数量。为了保持多数类样本的类内结构，我们选择按比例大小从多数类邻域中进行采样。但由于多数类局部模式学习的初始化具有随机性，学习到的邻域内不同区域的样本密度可能差异较大。因此，在选择样本之前，本文引入多数类样本的全局密度来进一步评估样本的选择权重，兼顾多数类样本的全局分布信息。多数类样本全局密度由式(5)计算得到。

$$GD_i=\sum_{x_j\in D_{\text{maj}}}\exp(-\|x_i-x_j\|_2) \tag{5}$$

依据上述公式，对 $\forall x_i\in D_{\text{maj}}(1\leqslant i\leqslant N_{\text{maj}})$，计算样本 x_i 与同类样本 x_j 之间的欧氏距离 $\|x_i-x_j\|_2$，然后由式(5)计算每个多数类样本 x_i 的全局密度 GD_i。

3.3 采样策略 1(CDUS1)

本节主要介绍第一种保持多数类类内结构信息的采样策略。由于局部模式中样本分布可能不均匀，CDUS1 试图为局部模式中样本基数较小的区域提供更高的选择优先级。因此，CDUS1 方法引入多数类的全局分布密度来描述每个局部模式中样本的分布情况。样本的密度越小，说明该样本周围的样本越少；相反，样本密度越大，说明其周围的样本越多。因此，我们为密度小的样本赋予一个更高的选择权重，以此来提升局部模式内稀疏区域中样本被选中的概率。

具体地，对任意一个局部模式 s_u 中的样本，其采样权重 ω_i 由式(6)和式(7)计算得到。

$$\rho_i=\frac{1}{GD_i} \tag{6}$$

$$\omega_i=\frac{\rho_i}{\sum_j^{|s_u|}\rho_i} \tag{7}$$

依据样本的采样权重，利用随机加权采样方法[22]，从 s_u 中抽取 Num_{s_u} 个样本。

CDUS1 方法的本质是试图平等地选择 s_u 中所有区域的样本。一般情况下，位于 s_u 内较低密度区域的样本应被赋予比较高密度区域的样本更大的权重。如图 4 所示，样本 A 和样本 B 相比邻域中的其他样本所在区域的密度更小，而样本 C 所在区域的密度较大，因此样本 A 和样本 B 比样本 C 被选中的概率更大。但由于右半圆区域有更多的样本(该区域的累积概率较高)，因此在进行加权随机抽样时，该方法可使得左半圆区域和右半圆区域均有较高的样本选择概率。

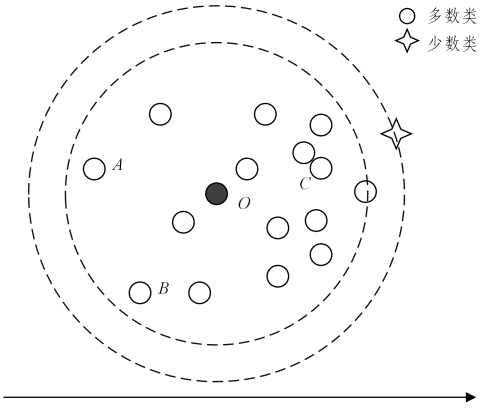


图 4 CDUS1 样本选择策略示意图

Fig. 4 Sketch map of CDUS1 sample selection strategy

数据中可能存在噪声、重叠和离群样本，而这些样本可能会对模型训练产生不利影响。在局部邻域检测过程中，它们受周围少数类样本的约束，会被分割成基数较小的局部模式(孤立点或覆盖内样本数量非常少)；同时，这些噪声样本、重叠样本和间断点相对其他样本来说，距离多数类聚集区域较远，密度相对较小。为了降低这些样本对分类的影响，我们合并所有符合条件的小基数局部模式，并设置样本密度越大得到的权重越大，密度越小得到的权重越小。具体地，对任意 $Num_{s_u}<1$ 的局部模式 s_u，将其所有样本放入集合 S_{rest} 中，然后利用式(8)计算 S_{rest} 中样本的权重，最后按采样权重对 S_{rest} 中的样本进行加权随机采样。

$$\omega_i=\frac{GD_i}{\sum_j^{|S_{\text{rest}}|}GD_j} \tag{8}$$

通过上述步骤，可以从多数类中选择 $N_{\min}$ 个样本。最后将选择的多数类样本与所有原始的少数类样本相结合，就可以得到一个平衡的数据集。

3.4 采样策略 2(CDUS2)

本节将介绍第二种样本选择策略(CDUS2)。CDUS2 考虑进一步将 s_u 划分成多个子区域，然后从多个子区域中选择样本，以保持多数类的类内结构。CDUS2 的主要思想是通过余弦相似度(Cosine Similarity，C_S)来度量样本间的相似性，并依据 C_S 将 s_u 划分成 4 个相似度不同的区域；然后根据 s_u 区域的样本规模依次循环加权随机抽样。局部模式的分区过程可以分解为以下两个步骤。

1)根据 C_S 将 s_u 中的样本分为 s_u^1 和 s_u^2 两个区域。首先确定 s_u 的基准向量，以局部模式 s_u 的中心样本 x_u 以及该局部模式中除样本 x_u 外全局密度值最小的样本 x_w 作为 s_u 的基准向量 $\overrightarrow{x_ux_w}$。然后，对任意样本 $x_j\in s_u$，计算 $\overrightarrow{x_ux_w}$ 与 $\overrightarrow{x_ux_j}$ 的 C_S，即 $\cos(\alpha)$(α 为 $\overrightarrow{x_ux_w}$ 和 $\overrightarrow{x_ux_j}$ 之间的夹角)。最后，根据 $\cos(\alpha)$ 的大小将 s_u 中的样本划分为相似度不同的区域。当 $\cos(\alpha)\geqslant 0$ 时，将样本划分到 s_u^1 区域；当 $\cos(\alpha)<0$ 时，将样本划分到 s_u^2 区域。s_u^1 和 s_u^2 的定义如下：

$$s_u^1=\{x_j\mid\cos(\alpha)\geqslant 0\}$$

$$s_u^2=\{x_j\mid\cos(\alpha)<0\}$$

2)根据样本与基向量之间的弧度将 s_u^1 划分成 s_u^{11} 和 s_u^{12}，将 s_u^2 划分成 s_u^{21} 和 s_u^{22}。s_u^1 中包含了所有 C_S 大于 0 的样本，根据基向量 $\overrightarrow{x_ux_w}$ 可以找到一个向量 $\overrightarrow{x_ux_a}$($x_a\in s_u^1$)使得到的 C_S 值最小，即 $\min\{\cos(\overrightarrow{x_ux_w},\overrightarrow{x_ux_a})\}$。同理，$s_u^2$ 中可以找到一个向量 $\overrightarrow{x_ux_b}$($x_b\in s_u^2$)使得到的 C_S 值最大，即 $\max\{\cos(\overrightarrow{x_ux_w},\overrightarrow{x_ux_b})\}$。

为了更好地描述子区域划分过程，令 O 表示邻域 s_u 的中心样本，W 表示样本 x_w，A 表示样本 x_a(s_u^1 区域中 C_S 值最小的样本)，B 表示样本 x_b(s_u^2 区域中 C_S 值最大的样本)，相应的基准向量为 $\overrightarrow{OW}$。

然后，通过样本与基准向量之间的弧度值将 s_u^1 和 s_u^2 划分为 4 个区域，其中以 $\overset{\frown}{WOA}$(向量 $\overrightarrow{OW}$ 与 $\overrightarrow{OA}$ 之间的弧度值)为 s_u^1 的基准弧度，以 $\pi-\overset{\frown}{WOB}$(向量 $\overrightarrow{OW}$ 与 $\overrightarrow{OB}$ 之间的弧度值)为 s_u^2 的基准弧度，弧度计算式如式(9)所示：

$$\overset{\frown}{xoy}=\arccos\frac{\overrightarrow{ox}\cdot\overrightarrow{oy}}{|\overrightarrow{ox}||\overrightarrow{oy}|} \tag{9}$$

因 s_u^1 和 s_u^2 的域划分规则类似，为简便起见，以 s_u^1 为例，对任意 $x_j\in s_u^1$(令 J 表示样本 x_j)，计算 $\overrightarrow{OJ}$ 与 $\overrightarrow{OA}$ 之间的弧度 $\overset{\frown}{AOJ}$，比较 $\overset{\frown}{AOJ}$ 与 $\overset{\frown}{WOA}$ 的大小。若 $\overset{\frown}{AOJ}\geqslant\overset{\frown}{WOA}$，则将样本 x_j 划分到 s_u^{11} 区域，否则将样本 x_j 划分到 s_u^{12} 区域。同理

可以将s_u^2划分成s_u^{21}和s_u^{22}。

CDUS2通过上述划分过程可以将s_u划分成4个等体积的空间区域s_u^{11},s_u^{12},s_u^{21}和s_u^{22}。它们的定义分别为:

$$s_u^{11}=\{x_j \mid \overset{\frown}{AOJ} \geqslant \overset{\frown}{WOA}, x_j \in s_u^1\}$$

$$s_u^{12}=\{x_j \mid \overset{\frown}{AOJ} < \overset{\frown}{WOA}, x_j \in s_u^1\}$$

$$s_u^{21}=\{x_j \mid \overset{\frown}{BOJ} \geqslant \pi-\overset{\frown}{WOB}, x_j \in s_u^2\}$$

$$s_u^{22}=\{x_j \mid \overset{\frown}{BOJ} < \pi-\overset{\frown}{WOB}, x_j \in s_u^2\}$$

当样本分布不均匀时,这4个区域所包含的样本数量会有所不同。因此,我们规定如下的采样顺序:首先,选取s_u的中心样本O,然后按照子区域样本规模对s_u^{11},s_u^{12},s_u^{21}和s_u^{22}进行降序排列。以轮转的方式依次(每次选择一个样本)从上述4个子区域中选择样本剩余的$Num_{s_u}-1$个样本。在选择过程中,对每个子区域中的样本仍然采取加权随机采样,如果某个区域内所有样本都已被选中,则不会进入下一轮的选择。

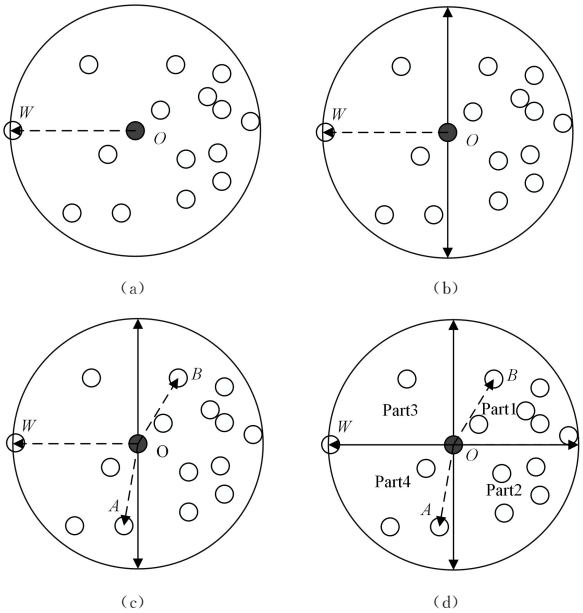


图5 CDUS2样本选择策略示意图

Fig. 5 Sketch of CDUS2 sample selection strategy

为了更好地理解CDUS2的工作原理,图5(a)—图5(d)给出了局部邻域s_u划分为Part1—Part4这4个区域的过程。首先找到当前邻域内除中心样本O以外全局密度值最小的样本W(见图5(a)),以$\overrightarrow{OW}$为邻域的基向量,对于邻域内的任一样本X,其与中心样本O的向量为$\overrightarrow{OX}$,计算$\overrightarrow{OX}$和基向量$\overrightarrow{OW}$的余弦相似度,并根据余弦值的正负性将样本划分为左右两个区域,如图5(b)所示。然后依据样本和基向量$\overrightarrow{OW}$的余弦相似度值,可以找到左半部分的余弦值最小的样本A和右半部分余弦值最大的样本B,如图5(c)所示。

我们以左半区域为例,对于任一属于左半区域的样本X,比较$\overset{\frown}{AOX}$和$\overset{\frown}{WOA}$的弧度大小即可将该区域进一步划分成两个区域($\overset{\frown}{AOX} \geqslant \overset{\frown}{WOA}$或$\overset{\frown}{AOX} < \overset{\frown}{WOA}$),由此可将左半区域划分为两个子区域,类似地,可以将右半区域也划分成两个子区域,如图5(d)所示。最后按4个区域中的样本数量大小降序排列,其中Part1,Part2,Part4,Part3中包含的样本数量分别为6,4,3,2。如果只选择一个样本,我们选择中心样本O,否则剩余的样本将按照Part1-Part2-Part4-Part3的顺序依次对各个区域内的样本进行加权随机采样。

此外,为了尽可能避免选择到重叠区域中的多数类样本,我们采取与式(8)同样的方式对数据规模较小的局部模式进行采样。

3.5 CDUS集成框架

多数类局部模式的学习涉及随机初始化过程,不同的初始化对应不同的局部模式,使得后续采样的结果具有多样性。

如图6(a)和图6(b)所示,同一数据集经过两次不同的随机初始化过程,得到了不同的局部模式,因此一次局部邻域的学习和采样可能无法达到最优性能。为了解决这一问题,本文引入基于bagging的集成策略,通过多次的局部模式学习和基于局部模式的采样,得到一组不同的平衡数据集,训练得到一组基分类器,通过融合一组分类器进一步提高CDUS方法的性能。CDUS的伪代码如算法1所示。

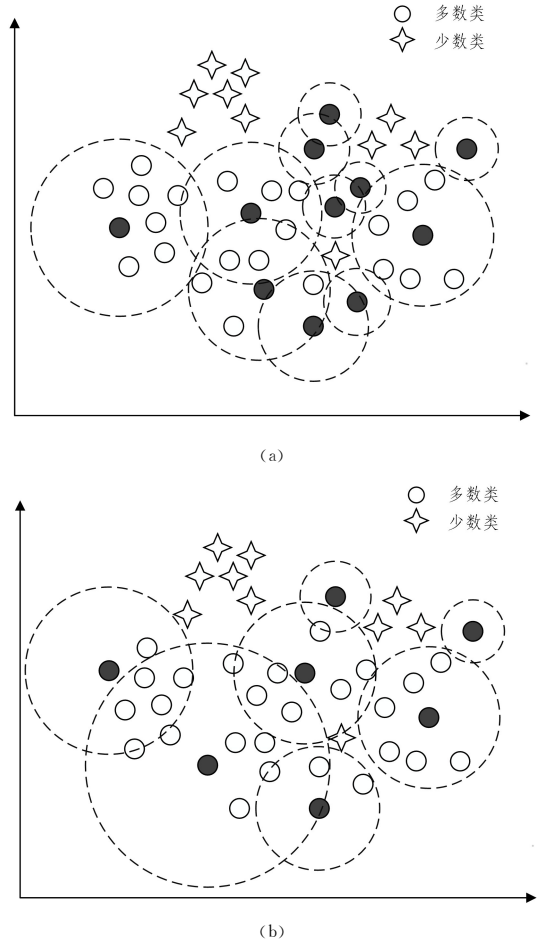


图6 局部模式检测的多样性示意图

Fig. 6 Sketch map of the diversity of local pattern detection

算法1 CDUS算法

输入:不平衡训练集D,集成次数T

[1] http://www.keel.es

输出：集成分类器 H

1. 划分 D 为多数类集合D_{maj}和少数类集合D_{min}；
2. t=0；
3. repeat
4. $S_{rest}=\{\}$，$S_{ord}=\{\}$，t=t+1；
5. Step1：局部邻域的挖掘
6. 构建多数类局部模式集合 $S=\{s_1,s_2,\cdots,s_v\}$；
7. 根据式(4)计算局部模式s_u的采样数量Num_{s_u}；
8. 根据式(5)计算多数类样本x_i的全局密度GD_i；
9. for s_u in S：
10. 若Num_{s_u}小于 1，则$S_{rest}\leftarrow s_u$，否则$S_{ord}\leftarrow s_u$；
11. end for
12. Step2：多数类欠采样
13. for s_u in S_{ord}：
14. 根据式(6)和式(7)计算s_u中样本的采样权重；
15. 采样策略 1：用 CDUS1 选择多数类样本子集D_{m1}；
16. 采样策略 2：用 CDUS2 选择多数类样本子集D_{m1}；
17. end for
18. 根据式(8)计算S_{rest}中样本的采样权重；
19. 对S_{rest}进行加权随机采样，选择多数类样本子集D_{m2}，使得$|D_{m1}|+|D_{m2}|=N_{min}$；
20. Step3：CDUS 集成
21. $D_{balance}=D_{m1}\cup D_{m2}\cup D_{min}$；
22. 使用$D_{balance}$训练基分类器 H[t]；
23. 构建集成分类器 H=H+H[t]；
24. until t=T.

4 实验设计与分析

为了验证本文提出的 CDUS 算法的有效性，我们采用 KEEL 数据库[1)]中的不平衡数据集进行实验。

4.1 数据集

本文从 KEEL 数据库中选取了 59 个不平衡数据集进行数值实验。数据集的不平衡率范围为 1.82～129.44，类重叠率范围为 0～0.99，样本数量范围为 129～5 472，属性范围为 2～40。这 59 个不平衡数据集的详细信息如表 1 所列，数据集按类不平衡率从小到大依次排列。表 1 中，IR 表示数据集的不平衡率，OR 表示类间的重叠程度[23]，Attr 表示样本的属性个数，Sample 表示样本的数量，Major 表示多数类样本的数量，Minor 表示少数类样本的数量。

表 1 KEEL 数据集的详细信息

Table 1 Detail information of KEEL dataset

数据集	IR	OR	Attr	Sample	Major	Minor	数据集	IR	OR	Attr	Sample	Major	Minor
glas1	1.82	0.26	8	214	138	76	shut0	13.87	0.01	8	1829	1706	123
ecol0	1.86	0.03	6	220	143	77	yeas17	14.30	0.84	6	459	429	30
wisco	1.86	0.03	8	683	444	239	glas4	15.46	0.65	8	214	201	13
pima	1.87	0.34	7	768	500	268	ecol4	15.80	0.19	6	336	316	20
iris0	2.00	0.00	3	150	100	50	abal9	16.40	0.92	7	731	689	42
glas0	2.06	0.19	8	214	144	70	derm6	16.90	0.00	33	358	338	20
yeas1	2.46	0.43	7	1484	1055	429	shut2	20.50	0.95	8	129	123	6
haber	2.78	0.65	2	306	225	81	shut6	22.00	0.48	8	230	220	10
vehi1	2.90	0.42	17	846	629	217	yeas1458	22.10	0.96	7	693	663	30
vehi3	2.99	0.48	17	846	634	212	yeas28	23.10	0.43	7	482	462	20
glas0123	3.20	0.19	8	214	163	51	lymph	23.67	0.96	17	148	142	6
ecol1	3.36	0.20	6	336	259	77	flare	23.79	0.80	10	1066	1023	43
newt1	5.14	0.17	4	215	180	35	krone	27.77	0.07	5	2244	2166	78
newt2	5.14	0.17	4	215	180	35	yeas4	28.10	0.83	7	1484	1433	51
glas6	6.38	0.27	8	214	185	29	wine4	29.17	0.97	10	1599	1546	53
yeas3	8.10	0.27	7	1484	1321	163	poker	29.50	0.85	9	244	236	8
ecol3	8.60	0.36	6	336	301	35	kddcg	29.98	0.02	40	1642	1589	53
page0	8.79	0.28	9	5472	4913	559	yeas1289	30.57	0.94	7	947	917	30
ecoll0234	9.10	0.23	6	202	182	20	abal3	32.47	0.19	7	502	487	15
glas015	9.12	0.90	8	172	155	17	wine9	32.60	0.97	10	168	163	5
yeas02579	9.14	0.17	7	1004	905	99	yeas5	32.73	0.33	7	1484	1440	44
yeas0256	9.14	0.41	7	1004	905	99	wine8	35.44	0.97	10	656	638	18
ecol046	9.15	0.23	5	203	183	20	ecol0137	39.14	0.28	6	281	274	7
ecol01	9.17	0.34	6	244	220	24	abal17	39.31	0.94	7	2338	2280	58
ecol0346	9.25	0.23	6	205	185	20	yeas6	41.40	0.42	7	1484	1449	35
yeas05679	9.35	0.59	7	528	477	51	wine39	58.28	0.98	10	1482	1457	25
glas016	10.29	0.91	8	192	175	17	shut25	66.67	0.00	8	3316	3267	49
ecol015	11.00	0.23	5	240	220	20	kddcl	75.67	0.00	40	1610	1589	21
glass0146	11.06	0.92	8	205	188	17	abal19	129.44	0.99	7	4174	4142	32
glas2	11.59	0.92	8	214	197	17							

4.2 评价指标

目前常用于评估分类器性能的指标主要有准确率(Accuracy)、精准率(Precision)、召回率(Recall)、G-mean、AUC 和 F1-score。Accuracy 表示预测正确的样本数量占总体样本数量的比例，该指标不适用于评估不平衡数据的学习性能。Precision 也被称为查准率，该指标表示预测为少数类样本中真实标签为少数类样本的比例。Recall 也被称为查全率，该指标表示真实标签为少数类样本中预测正确的少数类样本所占比例。在不平衡学习中，G-mean，AUC 和 F1-score 是最常用的评价指标，G-mean 综合考虑 Precision 和 Recall，该指标值越高，说明性能越好。AUC 是 ROC 曲线下的面积，数值越大，对应的分类器越好。F1-score 是 Precision 和 Recall 的调和平均

数，主要衡量分类器预测少数类样本的能力。

4.3 参数分析

本节主要分析基分类器独立训练次数 T 与 CDUS 性能之间的关系，集成次数 T 设置为 1，5，10，20 和 30，每组重复 10 次。使用随机森林（RF）分类器，CDUS1 和 CDUS2 在 10 个数据集上的 G-mean，AUC 和 F1-score 结果分别如图 7 和图 8 所示。其中，横坐标表示基分类器独立训练次数 T，纵坐标表示 CDUS 方法的性能，每个子图中包含了 CDUS 在 3 个评估指标上与训练次数 T 的关系。由图 7 和图 8 可以观察到，当 T 从 1 增加到 5 时，CDUS 性能有明显提升。除 glas016 和 lymph 外，算法在 $T=10$ 后的表现都相对稳定。

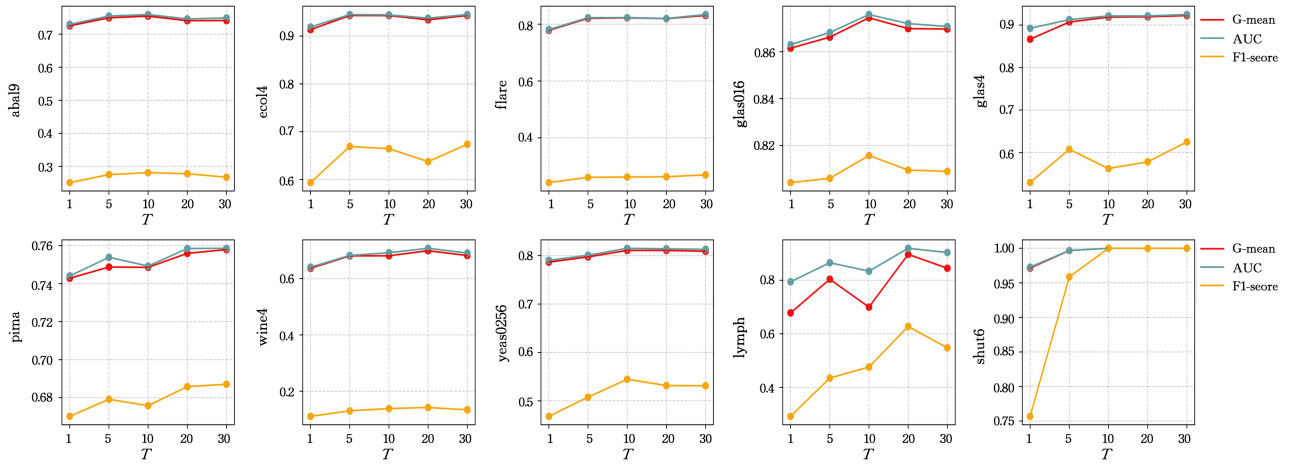


图 7 T 对 CDUS1 性能的影响

Fig. 7 Influence of T on CDUS1 performance

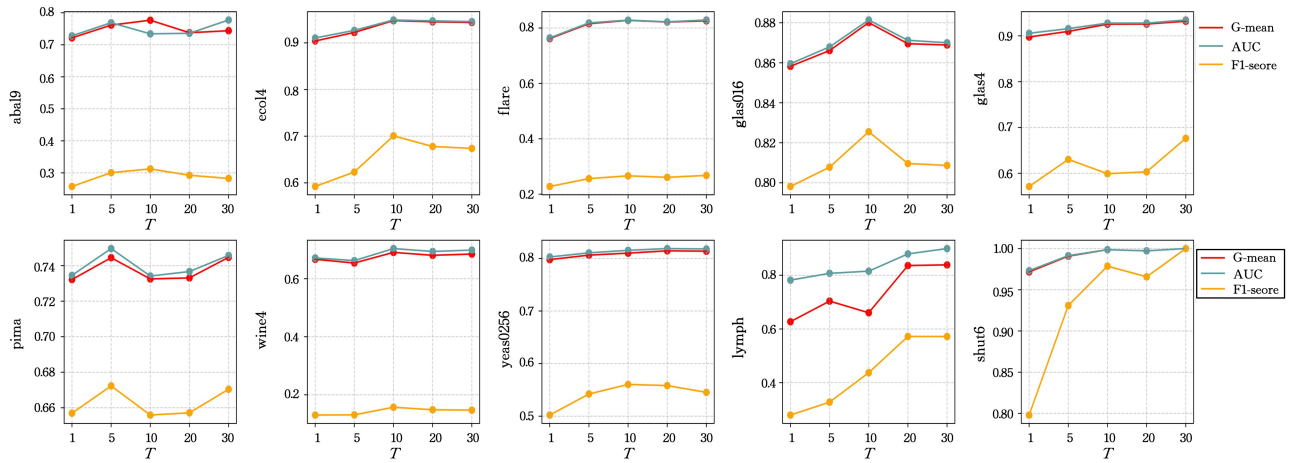


图 8 T 对 CDUS2 性能的影响

Fig. 8 Influence of T on CDUS2 performance

4.4 结果分析

为了验证本文方法的有效性，我们将 CDUS 方法与 13 种方法进行对比，包括 7 种集成方法，分别为 RUSB[24]，EUSB[25]，EasyEnsemble（EE），BalanceCascade（BC），UnderBagging（UB），AdaC2 和 IIVotes（IIV）[26]；2 种过采样方法，分别为 MWMOTE（MWM）和 GDO；3 种欠采样方法，分别为 CU1，CU2 和 RBU；1 种算法层面的方法，即 GSE。在上述方法中，EE，BC，RUSB，AdaC2，UB，EUSB 和 IIV 是针对特定的分类器而设计的。例如，EE 和 BC 使用分类回归树（CART）算法训练弱分类器，子分类器数为 4，每个 AdaBoost 的训练次数为 10。本文对这些方法均采用原文的推荐设置。为了保证实验结果的稳定性和准确度，我们对每个数据集进行了 10 次 5 折交叉验证，最终结果为 10 次结果的平均值。

表 2 列出了对比方法在 AUC 上的对比结果。由表 2 可知，CDUS1 和 CDUS2 在 59 个数据集上的均值分别为 0.8814 和 0.8808，且分别比排名第三的 EE（0.8514）高出 3.52%和 3.45%。此外，CDUS1 和 CDUS2 均在 33 个数据集上取得了最佳表现，对比方法中仅有 MWMOTE 在 9 个数据集上达到最优，可以看出 CDUS1 和 CDUS2 有较大的优势。

表 3 列出了本文算法与上述 13 种对比方法在 59 个数据集上 G-mean，AUC 和 F1-score 的均值以及每种方法在实验数据集上的平均排名，其中加粗数据表示性能排名前三的方法。具体而言，在 G-mean 上排名前三的方法分别为 CDUS1，CDUS2 和 UB，在 AUC 上排名前三的方法分别为 CDUS2，CDUS1 和 CU2，在 F1-score 上排名前三的方法分别为 CDUS2，MWM 和 CDUS1，从平均排名以及不同方法之间的性能差距可以看出本文方法取得了较好的性能。

表 2 15 种方法在 RF 分类器上的 AUC 实验结果

Table 2 Results of AUC of 15 methods on RF classifier

	CDUS1	CDUS2	RBU	GSE	CU1	CU2	BC	EE	RUSB	UB	MWM	GDO	AdaC2	IIV	EUSB
abal17	0.8492	**0.8534**	0.6994	0.8403	0.8525	0.8069	0.7964	0.8337	0.6907	0.8469	0.7302	0.7265	0.6905	0.6004	0.8145
abal3	0.9944	0.9942	**1.0000**	0.9959	0.9949	0.9949	0.9990	0.9947	0.9948	0.9949	**1.0000**	0.9994	0.9656	0.9656	0.9656
abal19	0.7722	**0.7774**	0.5304	0.7619	0.6997	0.7245	0.5561	0.7730	0.6217	0.7476	0.5225	0.4973	0.5093	0.4989	0.5904
abal9	0.7594	**0.7789**	0.6913	0.7433	0.7438	0.7443	0.6982	0.7405	0.6571	0.7230	0.6947	0.7032	0.6498	0.6348	0.6951
derm6	**1.0000**	**1.0000**	**1.0000**	0.9441	0.9719	0.9997	0.9985	0.9979	0.9829	**1.0000**	0.9950	0.9997	0.9750	0.9500	0.9956
ecol0137	0.8922	**0.9011**	0.8303	0.8619	0.8785	0.8624	0.8045	0.8743	0.8575	0.7614	0.6389	0.8963	0.8191	0.7390	0.6387
ecol01	0.9155	**0.9182**	0.9081	0.8291	0.8752	0.9059	0.8981	0.8175	0.8153	0.8555	0.8331	0.8717	0.8591	0.7632	0.8859
ecol015	**0.9705**	0.9523	0.9540	0.9182	0.9423	0.9455	0.9523	0.8841	0.8877	0.9177	0.8982	0.8618	0.8364	0.8591	0.8955
ecol0234	0.9537	**0.9542**	0.9459	0.8450	0.9379	0.9041	0.9020	0.8711	0.9156	0.8979	0.8940	0.8640	0.9032	0.9141	0.8143
ecol0346	0.9469	**0.9491**	0.9207	0.8446	0.9342	0.9455	0.8958	0.8965	0.8976	0.8897	0.8826	0.8665	0.8541	0.8534	0.8818
ecol046	0.9473	**0.9517**	0.9203	0.8370	0.9366	0.9288	0.8781	0.8701	0.9014	0.8855	0.8897	0.8908	0.8038	0.8505	0.8928
ecol0	0.9868	**0.9875**	0.9822	0.9526	0.9852	0.9864	0.9772	0.9758	0.9843	0.9783	0.9822	0.9806	0.9836	0.9696	0.9801
ecol1	**0.9121**	0.9061	0.8592	0.8363	0.8945	0.9012	0.8922	0.8609	0.8282	0.8967	0.8734	0.8816	0.8517	0.8656	0.8918
eco3	0.8972	**0.8993**	0.8241	0.8950	0.8980	0.8915	0.8696	0.8662	0.7917	0.8845	0.8263	0.7838	0.8830	0.8239	0.8763
ecol4	0.9450	**0.9491**	0.9172	0.8909	0.9453	0.9348	0.9152	0.9487	0.9233	0.9141	0.9043	0.9196	0.8889	0.7639	0.9450
flare	**0.8339**	0.8291	0.7253	0.6954	0.6475	0.8321	0.7355	0.8043	0.7918	0.8226	0.5978	0.5282	0.5000	0.6444	0.8087
glas0123	0.9603	**0.9628**	0.9591	0.8853	0.9207	0.9219	0.9389	0.9591	0.8991	0.9307	0.9273	0.9209	0.8960	0.8978	0.9244
glas0146	**0.7975**	0.7924	0.5968	0.7716	0.6548	0.6967	0.6747	0.7970	0.6290	0.7694	0.6324	0.5521	0.5256	0.6657	0.7387
glas015	**0.7762**	0.7504	0.6400	0.7497	0.5915	0.6448	0.7266	0.6641	0.6339	0.5848	0.6361	0.5933	0.5441	0.4669	0.7608
glas016	**0.7948**	0.7910	0.6037	0.7795	0.5550	0.6657	0.7200	0.7738	0.5953	0.6573	0.6370	0.5575	0.4831	0.6740	0.7064
glas0	0.8759	**0.8814**	0.8795	0.8393	0.8203	0.8289	0.8312	0.7979	0.7505	0.8519	**0.8485**	0.8350	0.8028	0.7718	0.8419
glas1	**0.8164**	**0.8164**	0.7773	0.7379	0.7504	0.7598	0.7637	0.7607	0.7180	0.7537	0.8160	0.8163	0.7294	0.7064	0.7780
glas2	0.7537	0.7696	0.6570	**0.7773**	0.6723	0.7411	0.7181	0.7436	0.6346	0.6980	0.6654	0.5891	0.6391	0.5993	0.6414
glas4	0.9247	**0.9347**	0.8625	0.8692	0.7908	0.9015	0.9312	0.9259	0.8756	0.9092	0.8503	0.8547	0.8258	0.8109	0.9143
glas6	0.9449	**0.9476**	0.9098	0.8860	0.9419	0.9302	0.9213	0.9197	0.9028	0.9376	0.9274	0.9462	0.9234	0.9005	0.9207
haber	**0.6475**	0.6302	0.6465	0.5561	0.5599	0.5750	0.5930	0.5774	0.5553	0.6136	0.6062	0.5928	0.5537	0.6432	0.5878
iris0	**1.0000**	**1.0000**	**1.0000**	0.9950	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9900	0.9800	0.9900
kddcg	**1.0000**	**1.0000**	**1.0000**	0.9991	**1.0000**	**1.0000**	0.9998	0.9999	0.9993	0.9998	**1.0000**	**1.0000**	0.9991	0.9909	0.9943
kddcl	**1.0000**	**1.0000**	0.9853	0.9956	0.9994	0.9972	**1.0000**	0.9947	0.9881	0.9994	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
krone	**1.0000**	1.0000	0.9986	0.9788	0.9690	0.9902	0.9999	0.9998	**1.0000**	0.9987	**1.0000**	**1.0000**	0.9614	0.9739	0.9965
lymph	**0.9195**	0.8996	0.7706	0.7635	0.8201	0.8147	0.7566	0.8580	0.6670	0.6666	0.6993	0.8500	0.7394	0.7929	0.6544
newt1	**0.9922**	0.9894	0.9887	0.9639	0.9539	0.9672	0.9574	0.9663	0.9734	0.9832	0.9612	0.9534	0.9381	0.9286	0.9294
newt2	**0.9922**	0.9906	0.9853	0.9639	0.9583	0.9553	0.9722	0.9517	0.9706	0.9746	0.9527	0.9385	0.9492	0.9401	0.9579
page0	0.9687	**0.9718**	0.9598	0.8885	0.8742	0.8905	0.9562	0.9555	0.8693	0.9601	0.9487	0.9339	0.8937	0.9533	0.9515
pima	**0.7586**	0.7496	0.7373	0.6943	0.7463	0.7445	0.7217	0.7580	0.7086	0.7420	0.7457	0.7504	0.7259	0.7209	0.7252
poker	**0.8729**	0.8400	0.7586	0.7469	0.6105	0.7896	0.7559	0.7154	0.5211	0.7071	0.5700	0.5800	0.6122	0.6416	0.6482
shut25	**1.0000**	**1.0000**	**1.0000**	0.9972	**1.0000**	0.9990	**1.0000**	**1.0000**	0.9975	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
shut6	**1.0000**	**1.0000**	0.9895	0.9932	0.9777	0.9573	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9955	0.9955	0.9750
shut0	**1.0000**	**1.0000**	**1.0000**	0.9974	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9997	0.9994	**1.0000**
shut2	**1.0000**	**1.0000**	**1.0000**	0.9713	0.9900	**1.0000**	**1.0000**	**1.0000**	0.9480	0.8975	**1.0000**	**1.0000**	0.9500	**1.0000**	0.9542
vehi**1**	**0.8023**	0.7984	0.7973	0.6953	0.7749	0.7673	0.7727	0.7620	0.6865	0.7772	0.7646	0.7504	0.7415	0.7268	0.7929
vehi3	**0.8013**	0.7982	0.7835	0.6697	0.7814	0.7692	0.7564	0.7654	0.7086	0.7718	0.7376	0.7331	0.7410	0.7348	0.7588
wine4	**0.7077**	0.7039	0.6982	0.5360	0.6127	0.6555	0.6356	0.6998	0.5436	0.6642	0.5503	0.5058	0.5701	0.5151	0.5969
wine8	**0.8235**	0.8224	0.7602	0.6323	0.7704	0.7749	0.7492	0.7754	0.7190	0.8219	0.5669	0.5753	0.5819	0.6494	0.7760
wine39	**0.7180**	0.7142	0.6576	0.5877	0.6111	0.6715	0.6567	0.6766	0.5822	0.6406	0.5064	0.5405	0.6712	0.5573	0.6333
wine9	0.8228	**0.8509**	0.8265	0.8167	0.7955	0.7795	0.8198	0.8127	0.7618	0.6710	0.6951	0.7739	0.5781	0.6536	0.8492
wisco	0.9773	**0.9779**	0.9699	0.9643	0.9735	0.9704	0.9568	0.9736	0.9467	0.9650	0.9734	0.9764	0.9611	0.9577	0.9694
yeas0256	0.8151	**0.8188**	0.7735	0.6891	0.6802	0.7734	0.8019	0.7947	0.6701	0.8014	0.7807	0.7543	0.6941	0.7496	0.7910
yeas02579	**0.9131**	0.9101	0.8939	0.8319	0.8083	0.8960	0.9099	0.8906	0.8204	0.9123	0.9040	0.8995	0.8836	0.8682	0.8899
yeas05679	0.8264	**0.8320**	0.7087	0.8087	0.7392	0.8292	0.8200	0.7960	0.6826	0.8022	0.7577	0.6798	0.7179	0.7515	0.7784
yeas1289	**0.7531**	0.7470	0.5976	0.6492	0.6368	0.6826	0.6130	0.6794	0.5677	0.7030	0.5766	0.5777	0.6419	0.5935	0.6839
yeas1458	0.6850	**0.7029**	0.5587	0.6552	0.5949	0.6548	0.6230	0.6313	0.5615	0.6929	0.5440	0.4980	0.4927	0.5379	0.5978
yeas17	0.7993	**0.8076**	0.6807	0.7252	0.6872	0.6967	0.7495	0.7655	0.6267	0.7756	0.6317	0.6703	0.6827	0.6050	0.8060
yeas28	**0.8109**	0.7951	0.7330	0.7753	0.6985	0.7535	0.7687	0.7974	0.6245	0.7476	0.7781	0.7739	0.5457	0.5859	0.7667
yeas1	**0.7392**	0.7383	0.7150	0.6628	0.7019	0.7084	0.6993	0.7237	0.6947	0.7287	0.7311	0.6970	0.6427	0.7098	0.7084
yeas3	**0.9379**	0.9367	0.8918	0.8891	0.9174	0.9218	0.9078	0.9192	0.8575	0.9262	0.8818	0.8576	0.8762	0.8952	0.9251
yeas4	0.8491	**0.8505**	0.6773	0.8476	0.7536	0.8256	0.8207	0.8225	0.7076	0.8417	0.7351	0.6123	0.7329	0.6938	0.8204
yeas5	**0.9669**	0.9636	0.8882	0.9566	0.9633	0.9603	0.9359	0.9567	0.9383	0.9667	0.8949	0.8644	0.8875	0.8677	0.9569
yeas6	0.8791	**0.8818**	0.7738	0.8542	0.7841	0.8485	0.8318	0.8594	0.7996	0.8611	0.8063	0.6712	0.7805	0.7506	0.8400
aver	**0.8814**	0.8808	0.8271	0.8261	0.8234	0.8478	0.8396	0.8514	0.7945	0.8428	0.8034	0.7957	0.7809	0.7823	0.8323

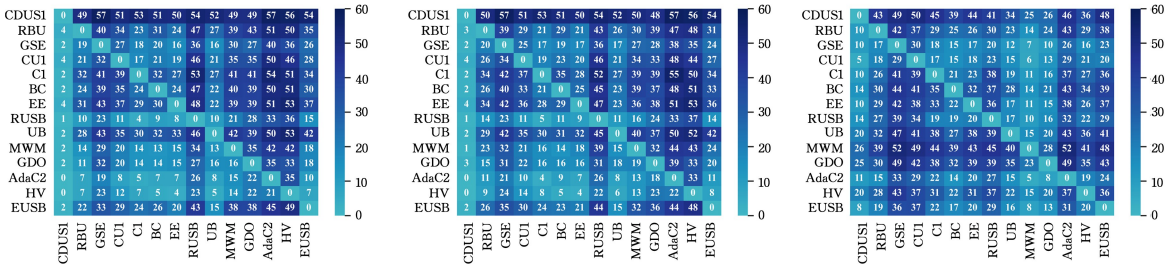
表 3 15 种方法在 RF 分类器上的总体性能和排名对比

Table 3 Comparison of average performance and rank of 15 methods on RF classifier

	G-mean	Rank	AUC	Rank	F1-score	Rank
CDUS1	0.8763	**2.46**	0.8814	**2.43**	0.6089	**5.31**
CDUS2	0.8749	**2.53**	0.8808	**2.40**	0.6157	**4.40**
RBU	0.8229	6.90	0.8271	7.64	0.5632	8.17
GSE	0.8093	9.68	0.8261	9.80	0.5270	11.10
CU1	0.8024	8.29	0.8234	8.24	0.5155	10.71
CU2	0.8373	6.62	0.8478	**6.64**	0.5606	8.75
BC	0.8251	7.28	0.8396	7.34	0.5932	7.32
EE	0.8430	6.67	0.8514	6.79	0.5727	8.52
RUSB	0.7508	11.16	0.7946	11.16	0.5586	9.53
UB	0.8263	**6.52**	0.8428	6.67	0.5841	6.98
MWM	0.7272	9.47	0.8034	8.97	0.6297	**4.81**
GDO	0.7006	10.10	0.7957	9.69	0.6307	5.87
AdaC2	0.7012	11.96	0.7809	11.92	0.5403	10.55
HV	0.7001	12.19	0.7823	12.06	0.5940	7.99
EUSB	0.8086	8.17	0.8323	8.27	0.5527	9.98

为了进一步研究 CDUS1 和 CDUS2 分别与 13 种对比方法之间的性能优劣，我们对 15 种方法两两分组进行进一步对比。图 9 和图 10 分别给出了 CDUS1 和 CDUS2 与 13 种对比方法在 59 个数据集上的性能值对比结果。图中每一行表示该行对应的方法与其他对比方法相比取得更好性能表现的数据集的个数。

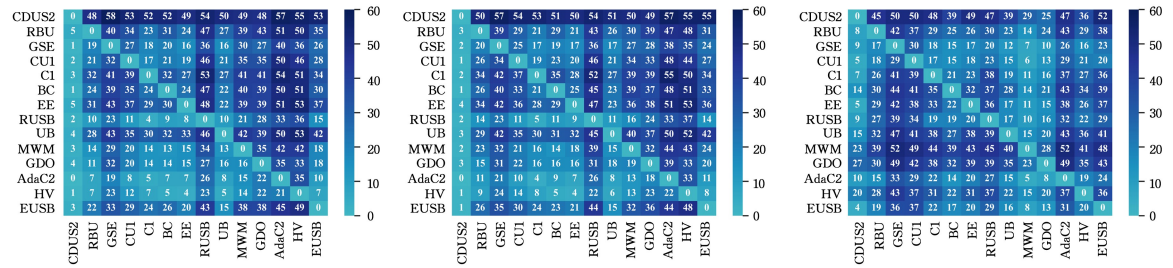
以图 9 中的 G-mean 为例，纵坐标上的 CDUS1 算法与横坐标上的方法相比，59 个数据集中 CDUS1 在 49 个数据集上的性能优于 RBU，在 57 个数据集上的性能优于 GSE，在 54 个数据集上的性能优于 EUSB。类似的可以得到所有算法之间的性能对比情况。除了从数值上观察这些算法之间的关系，还可以通过图中每个单元格的颜色深浅来判断，颜色越深表示该方法的分类效果越好。从图中可以看出，CDUS 算法在 3 个指标上得到的性能明显优于其他算法，尤其在 G-mean 和 AUC 上表现优异。



(a) G-mean (b) AUC (c) F1-score

图 9 CDUS1 与 13 种方法性能对比关系示意图

Fig. 9 Schematic diagram of one-to-one comparison between CDUS1 and 13 methods



(a) G-mean (b) AUC (c) F1-score

图 10 CDUS2 与 13 种方法性能对比关系示意图

Fig. 10 Schematic diagram of one-to-one comparison between CDUS2 and 13 methods

表 4 15 种方法在 RF 分类器上的 Friedman 检验统计值

Table 4 Friedman test of 15 methods on RF classifier

	评价指标	χ_F^2	F_F	p
CDUS1	G-mean	289.23	33.34	4.30×10^{-54}
	AUC	270.10	30.00	4.22×10^{-50}
	F1-score	166.68	15.45	8.71×10^{-29}
CDUS2	G-mean	285.66	32.71	2.39×10^{-53}
	AUC	270.88	30.14	2.90×10^{-50}
	F1-score	177.99	16.81	4.36×10^{-31}

同时，我们将 CDUS 算法和 13 种方法进行了无参的显著性统计假设检验，在 RF 分类器上的 Friedman 检验统计量结果如表 4 所列，其显著性水平 $\alpha=0.05$，其对应的卡方统计量和 F 分布统计量如下：$\chi_F^2(13)$ 为 22.36，$F_F(13,754)$ 为 1.73。从表中可以看出，CDUS1 算法与其他 13 种方法的 G-mean，AUC 和 F1-score 统计值 χ_F^2 分别为 289.23，270.10，166.68，统计值 F_F 分别为 33.34，30.00，15.45，这两个统计值远大于 $\chi_F^2(13)$ 和 $F_F(13,754)$ 统计量，且 p 值均小于 0.05。由此可得出结论：在 0.05 的水平上应该拒绝原假设，这 14 种方法的实验效果有显著性差异。同理，在 CDUS2 上可以得到相同的结论。

为了更好地比较这些方法的优劣，表 5 和表 6 分别列出了 CDUS1 和 CDUS2 与其他方法在 3 个评价指标上的统计检验秩排序。从秩排序结果可以看出，CDUS1 算法和 CDUS2 算法在 G-mean，AUC 和 F1-score 上的排序相比其他算法均最小，且具有明显优势，即分类性能最好。

综上所述，在 59 个不平衡数据上，本文提出的 CDUS 算法在 G-mean，AUC 和 F1-score 这 3 个指标上表现优异，上述对比实验结果证明了 CDUS 在保持原数据分布前提下对多数类样本进行采样的有效性和合理性。

表 5 CDUS1 和 13 种方法的 rank 对比

Table 5 Ranking comparison between CDUS1 and 13 methods

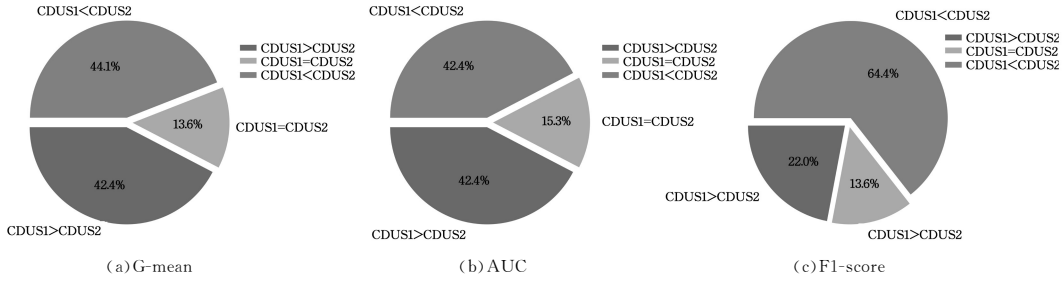
评价指标	CDUS1	RBU	GSE	CU1	CU2	BC	EE	RUSB	UB	MWM	GDO	AdaC2	IIV	EUSB
G-mean	**1.95**	6.03	8.69	7.36	5.70	6.35	5.80	10.22	5.63	8.58	9.23	10.97	11.24	7.25
AUC	**1.93**	6.75	8.83	7.29	5.70	6.42	5.90	10.22	5.76	8.06	8.80	10.93	11.10	7.31
F1-score	**4.59**	7.36	10.25	9.83	7.90	6.61	7.64	8.71	6.28	4.26	5.39	9.74	7.36	9.08
aver	**2.82**	6.71	9.26	8.16	6.44	6.46	6.45	9.72	5.89	6.97	7.81	10.55	9.90	7.88

表 6 CDUS2 和 13 种方法的 rank 对比

Table 6 Ranking comparison between CDUS2 and 13 methods

评价指标	CDUS2	RBU	GSE	CU1	CU2	BC	EE	RUSB	UB	MWM	GDO	AdaC2	IIV	EUSB
G-mean	**2.04**	6.02	8.71	7.39	5.69	6.36	5.78	10.21	5.59	8.58	9.20	10.97	11.22	7.23
AUC	**1.90**	6.75	8.83	7.34	5.70	6.42	5.90	10.21	5.75	8.05	8.81	10.93	11.08	7.33
F1-score	**4.11**	7.39	10.27	9.83	7.95	6.61	7.73	8.81	6.36	4.32	5.36	9.75	7.36	9.14
aver	**2.68**	6.72	9.27	8.19	6.45	6.47	6.47	9.74	5.90	6.98	7.79	10.55	9.89	7.90

另一方面，从上述的统计结果数据可以发现，CDUS1 和 CDUS2 在学习性能上没有显著差异，虽然这两种样本选择方法的设计角度不同，但 CDUS2 的平均排名优于 CDUS1。我们在所有测试数据集上对这两种算法进行一对一的比较，以进一步分析这两种方法。图 11 给出了 CDUS1 和 CDUS2 在 59 个数据集上的对比结果。图例中“CDUS1＞CDUS2”表示 CDUS1 性能优于 CDUS2 的数据集占比，“CDUS1＝CDUS2”表示 CDUS1 与 CDUS2 性能值相等的数据集占比，“CDUS1＜CDUS2”表示 CDUS2 性能优于 CDUS1 的数据集占比。从图中可以观察到，在 G-mean 和 AUC 上 CDUS1 与 CDUS2 的性能相差不大，在 F1-score 上 CDUS2 明显优于 CDUS1。



(a)G-mean (b)AUC (c)F1-score

图 11 CDUS1 与 CDUS2 的对比结果

Fig. 11 Comparison results between CDUS1 and CDUS2

结束语 本文提出了一种基于构造性神经网络与全局分布密度的不平衡数据欠采样方法。该方法主要提出了一种构造性的局部模式学习方法，并基于局部模式提出了两种保持多数类类内分布结构的采样策略。最后针对局部信息学习过程的随机初始化导致的结果非优问题，引入 bagging 集成策略来提升算法的性能。在 59 个数据集上与 13 种方法进行对比实验，验证了 CDUS 方法的有效性。本文研究关注二分类场景下的不平衡数据学习，如何将本文方法扩展到多分类场景是下一步研究的重点。

参 考 文 献

[1] CHAMSEDDINE E, MANSOURI N, SOUI M, et al. Handling class imbalance in COVID-19 chest X-ray images classification: Using SMOTE and weighted loss[J]. Applied Soft Computing, 2022, 129: 109588.

[2] GUO J F, WANG M S, SUN L, et al. New method of fault diagnosis for rolling bearing imbalance data set based on generative adversarial network[J]. Computer Integrated Manufacturing Systems, 2022, 28(9): 2825-2835.

[3] CHEN Z, ZHU M, DU J W. Multi-view graph neural network for fraud detection algorithm[J]. Journal on Communications, 2022, 43(11): 225-232.

[4] XIE Y X, QIU M, ZHANG H B, et al. Gaussian distribution based oversampling for imbalanced data classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(2): 667-679.

[5] LIN W C, TSAI C F, HU Y H, et al. Clustering-based undersampling in class-imbalanced data[J]. Information Sciences, 2017, 409: 17-26.

[6] ZHANG Y Q, LU R Z, QIAO S J, et al. A Sampling Method of Imbalanced Data Based on Sample Space[J]. Acta Automatica Sinica, 2022, 48(10): 2549-2563.

[7] DONG H C, WEN Z Y, WAN Y H, et al. An imbalanced data classification algorithm based on DPC clustering resampling combined with ELM[J]. Computer Engineering & Science, 2021, 43(10): 1856-1863.

[8] DRUMMOND C, HOLTE R C. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling[C]// Workshop on learning from imbalanceddatasets II. 2003: 1-8.

[9] WANG S, MINKU L L, YAO X. Resampling-based ensemble methods for online class imbalance learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 27(5): 1356-1368.

[10] CHAWLA N V, BOWYER K W, HALL L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of Arti-

ficial Intelligence Research, 2002, 16: 321-357.
[11] HAN H, WANG W Y, MAO B H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning[C]// International Conference on Intelligent Computing. Berlin: Springer, 2005: 878-887.
[12] HE H B, BAI Y, GARCIA E A, et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning[C]// 2008 IEEE International Joint Conference on Neural Networks (IEEE world Congress on Computational Intelligence). IEEE, 2008: 1322-1328.
[13] BARUA S, ISLAM M M, YAO X, et al. MWMOTE--majority weighted minority oversampling technique for imbalanced data set learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 26(2): 405-425.
[14] BUNKHUMPORNPAT C, SINAPIROMSARAN K, LURSINSAP C. Safe-level-smote: Safe-level-synthetic minority oversampling technique for handling the class imbalanced problem[C]// Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference. Berlin: Springer, 2009: 475-482.
[15] KOZIARSKI M. Radial-based undersampling for imbalanced data classification[J]. Pattern Recognition, 2020, 102: 107262.
[16] ZHANG Y P, ZHANG L, WANG Y C. Cluster-based majority under-sampling approaches for class imbalance learning[C]// 2010 2nd IEEE International Conference on Information and Financial Engineering. IEEE, 2010: 400-404.
[17] BARANDELA R, VALDOVINOS R M, SÁNCHEZ J S. New applications of ensembles of classifiers[J]. Pattern Analysis & Applications, 2003, 6: 245-256.
[18] LIU X Y, WU J X, ZHOU Z H. Exploratory undersampling for class-imbalance learning[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2008, 39(2): 539-550.
[19] SUN Y M, KAMEL M S, WONG A K C, et al. Cost-sensitive boosting for classification of imbalanced data[J]. Pattern Recognition, 2007, 40(12): 3358-3378.
[20] ZHU Z H, WANG Z, LI D D, et al. Geometric structural ensemble learning for imbalanced problems[J]. IEEE Transactions on cybernetics, 2018, 50(4): 1617-1629.
[21] ZHANG L, ZHANG B. A geometrical representation of McCulloch-Pitts neural model and its applications[J]. IEEE Transactions on Neural Networks, 1999, 10(4): 925-929.
[22] EFRAIMIDIS P S, SPIRAKIS P G. Weighted random sampling with a reservoir[J]. Information Processing Letters, 2006, 97(5): 181-185.
[23] VUTTIPITTAYAMONGKOL P, ELYAN E. Neighbourhood-based undersampling approach for handling imbalanced and overlapped data[J]. Information Sciences, 2020, 509: 47-70.
[24] SEIFFERT C, KHOSHGOFTAAR T M, VAN HULSE J, et al. RUSBoost: A hybrid approach to alleviating class imbalance[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2009, 40(1): 185-197.
[25] GALAR M, FERNÁNDEZ A, BARRENECHEA E, et al. EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling[J]. Pattern recognition, 2013, 46(12): 3460-3471.
[26] BŁASZCZYŃSKI J, DECKERT M, STEFANOWSKI J, et al. IIvotes ensemble for imbalanced data[J]. Intelligent Data Analysis, 2012, 16(5): 777-801.

YAN Yuanting, born in 1986, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include data mining, machine learning and granular computing.

（责任编辑：何杨）