一种基于聚类与多距离度量的下采样算法

杨 涛 [1]
YANG Tao

摘 要 数据进行采样处理可以有效解决许多领域在模型训练时所面临的数据不平衡问题。为提升数据下采样算法性能，有效剔除冗余样本，文章设计了一种基于聚类和多距离度量的下采样算法。首先，使用 K-means 算法对样本进行聚类；然后，根据簇内样本的分布情况，选择多数类较多的用于下采样；接着，从马氏距离、欧氏聚类和余弦相似度多个距离维度度量簇内样本分布情况；最后，根据综合距离和多数类样本间数量关系，确定单簇采样数量，剔除冗余样本，解决样本数量不平衡问题。在四个真实数据集和五个分类器上的实验结果表明，该算法能够有效提升分类模型的性能。

关键词 不平衡数据；K-means；多距离度量；下采样；数据处理

doi：10.3969/j.issn.1672-9528.2026.04.027

0 引言

在实际的模型训练与测试工作中，常受限于数据的不平衡问题。所谓数据不平衡，就是指数据集中，某些类别的样本数量多于其他类别，从而导致模型的训练和测试结果向多数类偏移，影响模型的整体性能。

在包括信用评分 [1]、故障车辆检测 [2]、医疗保险欺诈检测 [3]、网络异常流量检测 [4] 以及轴承故障诊断 [5] 在内的众多领域，均存在数据的不平衡问题。为了解决数据不平衡所带来的负面影响，各种方法应运而生，包括基于数据采样的数据层面的解决方法 [6]；基于特征工程 [7] 的方法；以及基于模型优化改进 [8] 的模型层面的方法。其中，数据层面的采样方法能够适应不同领域的多种类型数据，因而应用广泛。采样算法根据其对样本的处理方式，可以细分为上采样（over-sampling）、下采样（under-sampling）和集成采样（ensemble-sampling）。

上采样方法通过合成数据集中的少数类样本来平衡数据集，常用的少数类样本合成方法以 SMOTE（synthetic minority over-sampling technique）及其各种衍生算法 [9] 为主，具有代表性的有 SMOTE、ADASYN（adaptive synthetic sampling）、K-means SMOTE[10] 等算法。在处理不平衡程度较高的数据集时，算法需要合成大量样本，这可能会导致合成的少数类样本数量超过原有的少数类样本数量，造成数据集失真。

下采样算法通过清除多数类样本中的冗余样本和噪声样本来降低数据集的不平衡度，从而降低数据集的不平衡特性对后续模型的影响。最直接的采样算法是随机下采样，该算法通过随机删除多数类样本来降低数据不平衡度 [6]，这可能会损失重要样本，因而较少使用；而 NCR（neigh-

1. 重庆工信职业学院 重庆沙坪坝 400037

[4] 李爱华，续维佳，石勇 . 基于数据融合的商务智能与分析架构研究 [J]. 计算机科学 ,2022,49(12):185-194.
[5] 张合沛，陈馈 . 面向多源异构信息融合的隧道群组装备数据提取技术研究 [J]. 隧道建设 (中英文),2022,42(1):41-47.
[6] 蒲未来，刘敦龙，桑学佳，等 . 融合多源异构数据的滑坡变形阶段智能判识方法 [J]. 灾害学 ,2023,38(4):179-186.
[7] 吴羿龙，余珊，陈莹捷，等 . 融合多源异构大数据的同安内涝风险评估 [J]. 测绘与空间地理信息 ,2025,48(4):70-73.
[8] 杨童博，魏政，周坤 . 基于近似失效点的多源异构数据融合可靠性评估模型 [J]. 机械强度 ,2025,47(3):136-142.
[9] 卢加荣，廖彬，刘怡，等 . 融合多源异构数据的 ICO 欺诈预测与可解释分析模型 [J]. 计算机应用研究 ,2025,42(2):357-364.
[10] 费英群，田林 . 面向多源异构的电力工程数据融合处理技术研究 [J]. 电子设计工程 ,2025,33(1):104-108.

【作者简介】
王伟（1978—），男，山东诸城人，本科，高级工程师，研究方向：电子信息。

（收稿日期：2025-09-04 修回日期：2026-04-08）

borhood cleaning rule）和 ENN（edited nearest neighbors）[6] 则对多数类样本进行清洗，删除其中的冗余和噪声样本，从而避免清除其中的重要样本，因此应用较为广泛；除此之外，还有将整个数据集作为整体，通过计算样本之间的互类势能，从而清除冗余样本的 RBU（radial-based under sampling）[11] 算法；将聚类与距离度量结合，清除冗余样本的谱聚类算法 [12]。

将上采样和下采样算法结合，在扩充少数类样本的同时，清除多数类中的冗余样本，从而降低数据集的不平衡度，便可得到集成采样算法，SMOTETomek 与 SMOTEENN 是其中较为常用的两种；除此之外，还有各种根据具体需求而结合的集成采样算法 [13]。

针对不平衡数据集中，多数类中存在冗余和噪声样本这一问题，本文提出了一种基于 K-means 聚类和多距离度量的不平衡数据采样方法 USCD（under-sampling based on clustering and multi-distance measurement），首先，该算法先将整个数据集聚类成多个簇，并分析各簇内部的样本分布情况；然后，根据样本分布情况选定符合条件的簇，用于下采样；接着，根据整体需要清除的冗余样本数量和选定簇内的多数类样本数量，分配各簇所需清除的样本数；最后，在簇内，根据多距离度量结果，删除冗余和噪声样本。实验结果表明，USCD 算法具有较高的冗余和噪声样本清除性能。

1 相关理论与算法模型

1.1 K-means 聚类算法

K-means 聚类算法是一种简单、易于理解和实现的高效聚类算法 [14]，其主要过程如下：

输入：

（1）数据集：$D=\{x_1, x_2, \cdots, x_n\}$，其中 x_i 表示第 i 个样本，具有 n 个特征和一个 label；

（2）簇的数量：K，通过设置超参数 K 从而控制聚类后的簇的数量；

（3）最大迭代次数 max_iter，收敛阈值 ε。

输出：

（1）每个样本 x 的簇标签 c_label = $\{C_1, C_2, \cdots, C_k\}$；

（2）K 个簇的中心 cenrtoids = $\{\mu_1, \mu_2, \cdots, \mu_k\}$。

算法处理步骤：

Step1：初始化

随机选择 K 个样本作为簇的初始中心点 centroids。

Step2：迭代

for (iter=0; iter <= max_iter; iter++) or（簇中心变化小于收敛阈值 ε）

{

（1）对每一个样本 x_i：

（a）计算其到每个簇中心 μ_j 的距离；

（b）将 x_i 分配给距离最近的簇中心所属的簇 c_i。

（2）对每个簇 $c_j=\{C_1, C_2, \cdots, C_k\}$：

if 簇 j 中没有数据点：

选择保留簇中心或者重新随机初始化；

else 簇 j 中有数据点：

将簇中心 μ_j 更新为该簇中所有样本的均值；

$\mu_j=\text{mean}(\{x_i \mid c_i=j\})$

（3）检查是否收敛与达到最大迭代次数：

if 所有簇中心的变化小于 ε or (iter==max_iter)：

停止迭代。

}

Step3：返回 c_label 和 centroids。

1.2 距离度量方法

1.2.1 余弦相似度

在计算余弦相似度时，将两个样本看成两个向量，通过计算二者之间的夹角余弦值，判断二者的相似程度 [15]，计算过程用公式表示为：

$$\text{cosinesimilarity}=\frac{\sum_{i=1}^{n}a_i b_i}{\sqrt{\sum_{i=1}^{n}a_i^{\,2}}\times\sqrt{\sum_{i=1}^{n}b_i^{\,2}}} \tag{1}$$

式中：n 表示样本的特征数量；a 和 b 表示样本；a_i 和 b_i 表示样本的第 i 个特征值。

当 cosine 值越大，则二者相似度越大，二者完全相同时对应的值为 1；反之则相似度越小。在后续使用过程中，为了保持距离的一致性，通过式（2）将余弦相似度转换成余弦距离，相似程度越大，则其距离越小。

$$\text{cosdistance=1-cosinesimilarity}=1-\frac{\sum_{i=1}^{n}a_i b_i}{\sqrt{\sum_{i=1}^{n}a_i^{\,2}}\times\sqrt{\sum_{i=1}^{n}b_i^{\,2}}} \tag{2}$$

1.2.2 欧氏距离

欧氏距离是两个点之间的直线距离 [16]，在多维度样本空间，欧氏距离的计算公式为：

$$d=\sqrt{\sum_{i=1}^{n}\left(x_{1i}-x_{2i}\right)^2} \tag{3}$$

式中：d 表示两个样本之间的欧氏距离；x_1 和 x_2 表示样本点；x_{1i} 和 x_{2i} 表示样本 x_1 和 x_2 的第 i 个特征；n 表示每个样本有 n 个特征。

1.2.3 马氏距离

马氏距离的计算过程将各变量之间的相关性纳入考查，

利用向量间的协方差矩阵表示距离[17]，其计算过程为：

$$D_n(x)=\sqrt{(x-\mu)^{\mathrm{T}}\boldsymbol{A}^{-1}(x-\mu)} \quad (4)$$

式中：x 表示一个具有 n 个特征的样本 $x=\{x_1, x_2,\cdots, x_n\}$；$\boldsymbol{A}^{-1}$ 表示 $\boldsymbol{A}$ 的逆矩阵。

2 方法设计

为有效清除不平衡数据集中的冗余样本，降低数据的不平衡度，提升模型的训练和测试性能，设计了一种基于 K-means 聚类和多聚类度量的下采样算法 USCD，该算法主要包括以下部分，其采样过程如图 1 所示。

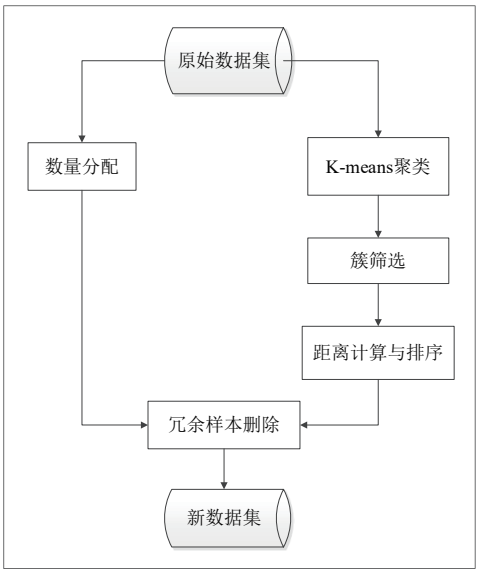

图 1 采样流程

首先，使用 K-means 聚类算法对整个数据集进行聚类，图 2 中 a 部分表示聚类前的数据。在聚类后的簇中，有的簇可能只有多数类样本，有的可能只有少数类样本，有的则包含两类样本，如图 2 中 b 部分所示，因此需要对簇进行筛选，方便后续的采样操作。通过统计簇内多数类样本和少数类样本的分布情况，对簇进行筛选：若簇只有少数类样本，如图 2 中 b 部分的 b2，则不用于下采样；若簇内包含两类样本或只含有多数类样本，且多数类样本数量大于少数类样本，即簇内存在不平衡情况如图 2 中 b 部分的 b1 和 b4，则用于下采样。

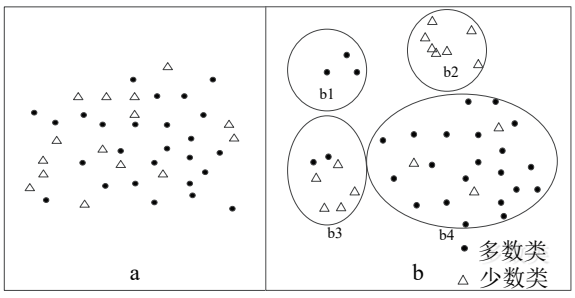

图 2 簇选择图示

然后，根据整体需要清除的样本数量以及各选定簇内的多数类样本数量，分配每个簇所需删除的冗余样本数量。在选定的簇内，计算每个样本的余弦距离、欧氏距离和马氏距离并归一化，按照距离进行排序从而删除冗余样本。

在对样本进行删除时，基于以下原则：在多数类样本内部，样本 x 距离其他样本的距离越大，则说明 x 越远离多数类样本的分布中心，所具备的有用信息越少，因此距离进行逆序排列，则排名靠前的样本具有更大的冗余性；在少数类样本中，一个多数类样本 x 距离少数类样本的整体距离越近，说明该样本越靠近分类边界，越可能是噪声样本，因此通过计算所有的多数类样本与少数类样本的距离，可以得到一个多数类样本与少数类样本相似程度的度量结果，将其正序排列，则距离越近的样本，排名越靠前，越可能是边界和噪声样本。

最后，综合衡量样本 x 在多数类样本内部的距离排名和在少数类样本间的距离排名，根据分配的删除数量，依次清除冗余样本。

用伪代码和步骤描述的算法过程如下：

输入：数据集 D，类别标签向量 $\boldsymbol{L}$；K-means 聚类算法的 K 值。

输出：平衡后的数据集 D'。

Step1：数据预处理

（1）获取数据的全局类别分布：

（a）C = unique($\boldsymbol{L}$) // 提取唯一类别标签。

（b）if $|C|<2$ then return (D, $\boldsymbol{L}$) // 单类别无需处理。

（2）确定全局类别：

（a）G_maj = argmax(| $\boldsymbol{L}$ = C_major |); // 全局多数类。

（b）G_min = argmin(| $\boldsymbol{L}$ = C_minor |); // 全局少数类。

Step2：聚类成 K 个簇

（1）cluster_id={1,2,⋯,K}

（2）统计各簇的原始类别分布情况：

for(i=0; i<=cluster_id; i++)

{

（a）提取该簇的样本标签 cluster_labels；

（b）统计簇中多数类与少数类数量；

（c）保存至 cluster_class_stats；

}

Step3：判断是否需要进行下采样

（1）minority_count = 全局少数类总样本数；

（2）total_majority = 全局多数类总样本数；

（3）total_remove = total_majority - minority_count；// 计算需要删除的多数类样本总数。

if (total_remove <= 0 或类别数 < 2)

返回原始数据 X、y 及采样前统计信息。

Step4：收集需要处理的簇的信息

for(i=0; i<=cluster_id; i++)

{

if(当前簇多数类数量 > 少数类数量)

{

（a）加入需处理簇列表 cluster_data_list;

（b）初始化删除信息 deletion_info;

}

else:

直接加入最终结果集合

}

Step5：按簇处理多数类样本

for(i in cluster_data_list)

{

（1）读取 (cluster_id, cluster_data, cluster_labels);

（2）提取簇中多数类与少数类样本 X_majority, X_minority;

（3）计算簇中心 mean_vec 与少数类均值 minority_mean;

（4）计算簇内多数类样本距离：

（a）intra_distances = compute_distances(X_majority, mean_vec, onents, inv_cov_matrix)；

（b）标准化并计算综合簇内多数类样本之间的距离 avg_intra；

（5）计算簇内多数类样本与少数类样本间距离：

（a）inter_distances = compute_distances(X_majority, minority_mean, inv_cov_matrix)；

（b）标准化并计算综合多数类样本与少数类样本之间的距离 avg_inter；

（6）综合排名：

（a）a_ranks = argsort(argsort(-avg_intra)); # 簇内多数类样本间距离降序排名。

（b）b_ranks = argsort(argsort(avg_inter)); # 簇内多数类和少数类样本间距离升序排名。

（c）combined_scores = a_ranks + b_ranks;

（d）sorted_indices = argsort(combined_scores); # 综合排序。

（7）计算需删除样本数：

remove_count=int(round(total_remove×(X_majority/total_majority_in_clusters)))；

（8）构建处理后样本：

（a）processed_X = [X_majority, X_minority];

（b）processed_y = [y_majority, y_minority];

（9）合并到最终结果：

（a）undersampled_data = undersampled_data +processed_X;

（b）undersampled_labels = undersampled_labels +processed_y;

}

Step6：返回处理后的数据集 D'。

从以上过程可以发现，USCD 算法并没有直接将整个数据集当成一个整体处理，而是分成了很多簇，再对簇进行采样处理，这是因为在数据集中，样本并不是均匀分布的，有些地方样本数量多，而有的地方少，存在空间分布不均。先采用聚类算法对其进行聚类处理，可以有效缓解这种空间分布式的不均衡，同时在处理大规模数据时，将整个数据集进行聚类拆分，可以降低单次计算所需的算力资源。在进行采样时，USCD 算法从欧氏距离、马氏距离以及余弦相似度三个维度来衡量样本之间的亲近关系，有效避免了使用单一距离所造成的距离度量不全面的问题；通过对多数类距离和少数类距离进行综合排序，将簇内样本的整体分布情况进行了衡量，有效地找到了冗余样本和噪声样本。

3 实验

为验证 USCD 采样算法的可行性和有效性，选取了 UCI 机器学习库中的四个数据集、imblearn 库中的多种采样算法以及 sklearn 库中的五个基本分类算法进行实验。

3.1 实验数据集

选用 UCI 数据库中的 Churn、Default、German 和 Wisconsin 四个数据集作为测试数据。Churn 共有 3 150 个样本，来自一家伊朗电信公司，描述了用户的流失情况，其特征包含了此前 12 个月用户的呼叫、短信、年龄、金额等信息；Default 数据集则包含 30 000 个样本，描述了中国台湾客户的支付违约案例，每个样本包含一个标签和 23 组特征，这些特征包括年龄、性别、受教育程度等；German 数据集则包含 1 000 个信用评分数据，每个样本具有 20 个特征和一个类别标签，良好与不良的数量分别是 700 和 300；而 Wisconsin 数据集则来自医学临床实践，一共有 699 个样本，删除其中特征缺失的，还剩 683 个完整样本。表 1 展示了以上四个数据集的基本情况。

表 1 数据集基本信息

数据集	样本数量	特征维度	多数类 / 少数类
Churn	3 150	13	2 655/495
Default	30 000	23	23 364/6 636
German	1 000	20	700/300
Wisconsin	683	9	444/239

3.2 评价指标

因为最终结果是通过分类器的训练和测试效果进行衡量的，且数据存在不平衡特性，因此选用 Accuracy、AUC（area under the curve）[18]、*G*-mean 和 *F*-measure[19] 四个指标来衡量测试结果。

通过表 2 所示的混淆矩阵，可以计算得到相应的分类指标。

表 2 混淆矩阵

真实标签	预测标签	
	“正”	“负”
“正”	TP(True Positive)	FN(False Negative)
“负”	FP(False Positive)	TN(True Negative)

$$\text{Accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{5}$$

$$F-\text{measure}=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{6}$$

式（6）中的 Precision 和 Recall 的计算公式为：

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{7}$$

$$\text{Recall}=\text{TPR}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{8}$$

G-mean 的计算公式为：

$$G\text{-mean}=\sqrt{\frac{\text{TP}}{\text{TP}+\text{FN}}\times\frac{\text{TN}}{\text{TN}+\text{FP}}} \tag{9}$$

AUC 的数则是通过计算 ROC（receiver operating characteristic）曲线下的面积得到。

3.3 实验设计

（1）为验证 USCD 算法的可行性，用该算法对表 1 中的四个数据集进行采样处理，清除多数类中的冗余样本，降低数据集的不平衡度。

（2）为验证 USCD 算法有效性，选用 imblearn 库中的、SMOTENN、SMOTETomek、SMOTE、ADASYN、K-means SMOTE、ENN、NCR 以及 RBU 算法作为对比算法，这些算法均使用库中的默认参数；其中 SMOTENN 和 SMOTETomek 是集成采样，SMOTE、ADASYN 和 K-means SMOTE 是上采样，ENN、NCR 以及 RBU 则为下采样算法。经过多次参数优化，在 USCD 算法中，K-means 聚类算法中 *K* 值均设置为 5。

（3）为了测试不同采样算法在分类器上分类效果，在使用采样算法对数据集进行处理以后，使用 LR、KNN、SVM[20]、NB 和 DT[21] 模型对其进行分类测试。分类器参数均采用 sklearn 库中的默认参数。为了降低偶然性，提升测试结果的可靠性，每个分类器均使用 5 折交叉验证，每次选择 20% 的样本作为测试集，剩余 80% 为训练集。在此基础上，将五个分类器得到的指标求平均值，作为对应采样算法的测试结果，从而进一步降低测试结果的偶然性，提高测试的可靠性。

4 实验结果分析

4.1 采样前后数据情况

图 3 展示了采样前后四个数据集的样本数量变化情况，在采样前后，数据集中少数类样本数量不发生改变，因此图中二者的折线相互重合；多数类样本中的冗余和噪声样本被清除，整体数量下降到少数类样本相同水平。

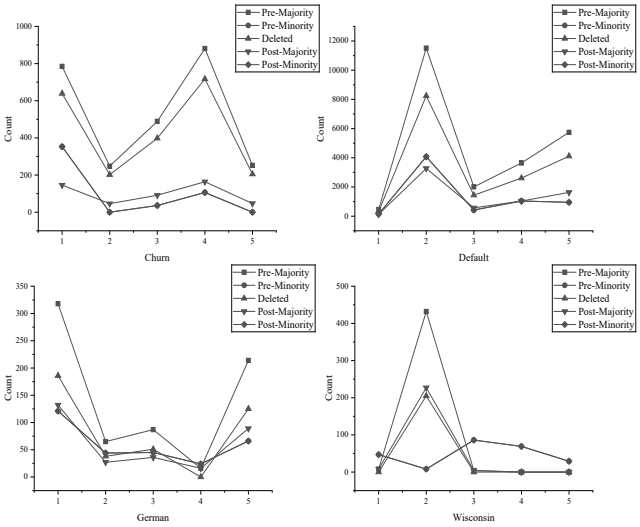

图 3 USCD 采样前后样本情况

图 3 中横轴的 1 到 5 表示簇的编号，可以看到，在 Churn 和 Default 数据集中，在采样之前，5 个簇中都是多数类样本多于少数类样本，因此，每一个簇都进行了采样处理；在 German 数据集中，簇 4 的多数类样本少于少数类样本，因此该簇被剔除，没有进行采样处理，USCD 前后其样本数量不变化；而 Wisconsin 数据集，只有簇 2 才满足采样条件，即多数类样本大于少数类样本，簇 4 和 5 少数类样本为 0，簇 1 和 3 则是少数类大于多数类，因此这四个簇都未进行采样处理。在对四个数据集进行 USCD 采样操作以后，有效降低了数据集的不平衡度，基本实现了类别平衡，以上采样结果证明 USCD 算法是可行的。

4.2 分类指标情况

使用 USCD 和几种对比采样算法对四个数据集进行采样处理，处理后的数据经过单一分类器 5 折交叉验证并计算所有分类器的平均值以后，整理得到表 3~6，每一个表格展示一个数据集的分类结果。表格中，Algorithm 表示采样算法，

其中 None 表示未经采样处理的原始数据集；四个指标均保留小数点后四位，性能最好的指标进行了加粗显示。

表 3 Churn 数据集分类结果

Algorithm	Accuracy	AUC	*G*-mean	*F*-measure
None	0.899 8	0.931 0	0.781 5	0.808 7
SMOTEENN	0.918 8	0.957 9	0.917 0	0.918 1
SMOTETomek	0.881 2	0.932 4	0.880 3	0.880 9
ADASYN	0.880 8	0.933 4	0.880 1	0.880 6
K-means SMOTE	0.906 6	0.950 3	0.905 3	0.906 3
SMOTE	0.879 4	0.930 9	0.878 4	0.879 1
USCD	0.905 9	0.954 6	0.905 3	0.905 8
ENN	0.921 8	0.946 9	0.876 7	0.875 5
NCR	0.918 8	0.944 4	0.866 7	0.867 6
RBU	**0.949 9**	**0.984 6**	**0.949 0**	**0.949 7**

表 4 Default 数据集分类结果

Algorithm	Accuracy	AUC	*G*-mean	*F*-measure
None	0.735 1	0.687 1	0.514 3	0.591 0
SMOTEENN	0.770 6	0.830 6	0.718 4	0.741 1
SMOTETomek	0.710 5	0.778 3	0.698 1	0.704 5
ADASYN	0.702 0	0.765 7	0.691 0	0.696 7
K-means SMOTE	0.764 5	**0.834 0**	0.756 0	0.760 7
SMOTE	0.708 3	0.775 3	0.696 0	0.702 4
USCD	**0.779 7**	0.821 7	**0.776 3**	**0.778 5**
ENN	0.712 1	0.742 4	0.643 6	0.670 0
NCR	0.705 8	0.726 7	0.618 3	0.650 2
RBU	0.663 6	0.722 5	0.652 3	0.658 2

表 5 German 数据集分类结果

Algorithm	Accuracy	AUC	*G*-mean	*F*-measure
None	0.719 6	0.720 9	0.489 6	0.603 1
SMOTEENN	0.753 3	0.823 5	0.714 7	0.734 3
SMOTETomek	0.755 4	0.811 8	0.754 3	0.755 0
ADASYN	0.753 1	0.808 6	0.752 2	0.752 8
K-means SMOTE	0.758 9	0.816 1	0.757 7	0.758 4
SMOTE	0.753 6	0.810 9	0.752 6	0.753 2
USCD	**0.782 7**	**0.832 2**	**0.781 4**	**0.782 2**
ENN	0.691 8	0.759 1	0.685 7	0.688 7
NCR	0.695 6	0.745 9	0.691 5	0.693 6
RBU	0.653 3	0.698 7	0.651 6	0.652 5

表 6 Wisconsin 数据集分类结果

Algorithm	Accuracy	AUC	*G*-mean	*F*-measure
None	0.923 3	0.954 4	0.927 3	0.917 6
SMOTEENN	0.954 6	**0.975 1**	0.954 2	0.954 5
SMOTETomek	0.935 7	0.963 2	0.934 8	0.935 6
ADASYN	0.920 8	0.954 0	0.920 5	0.920 7
K-means SMOTE	0.935 6	0.961 9	0.934 4	0.935 4
SMOTE	0.934 5	0.961 1	0.933 4	0.934 3
USCD	**0.955 3**	0.967 3	**0.954 6**	**0.955 2**
ENN	0.944 7	0.970 4	0.947 2	0.940 9
NCR	0.947 1	0.972 1	0.949 5	0.943 4
RBU	0.898 8	0.940 7	0.896 7	0.898 3

在表 3~6 的四个数据集中，可以看到未经采样处理的 None 数据在 Accuracy 和 AUC 指标上表现较好，但是其在 *G*-Mean 和 *F*-measure 上表现却并不理想，因为这两个指标均衡考虑了两类样本的分类结果，而未经过采样处理的数据会使得分类结果偏向多数类，导致 *G*-Mean 和 *F*-measure 值较低。

在表 3 的 Churn 数据集上，RBU 是表现最好的方法，四个指标均超过了 0.94。相较于 SMOTE、SMOTETomek、ADASYN 等传统采样方法，USCD 在 Accuracy 指标上提升超过 2.5%，AUC 提升 2% 以上，*G*-Mean 和 *F*-measure 提升均超过 2.8%。虽然在 Accuracy 上略逊于 ENN、NCR 和 RBU，但在 AUC、*G*-Mean 和 *F*-measure 上表现更均衡。

在表 4 的 Default 数据集中，USCD 表现良好。具体而言，USCD 在 Accuracy、AUC、*G*-Mean 和 *F*-measure 四项指标上分别达到了 0.779 7、0.821 7、0.776 3 和 0.778 5。与未采用任何处理方法（None）的结果相比，USCD 在 Accuracy 提升了约 6.1%，AUC 提升 19.6%，*G*-Mean 提升 51.3%，*F*-measure 提升 31.5%。即使与表现较好的 K-means SMOTE 相比，USCD 在 Accuracy 提升约 2.0%，AUC 虽略有下降，但在 *G*-Mean 和 *F*-measure 上均有约 2.7% 和 2.3% 的提升。

在表 5 的 German 数据集中，与 SMOTEENN、SMOTE-Tomek 相比，USCD 在准确性和 *F*-Measure 上提升了约 3%；相较于同为下采样算法的 ENN 和 NCR，在四个指标上的提升均超过了 8%；相比 RBU，提升则超过了 10%，这显示出 USCD 算法具有优秀的冗余和噪声样本清除性能。

在表 6 的 Wisconsin 数据集中，USCD 在各项指标中表现优秀，其准确率达到 0.955 3，略优于 SMOTEENN 和 NCR，显示出更强的分类性能；AUC 则为 0.967 3，与最高的 SMOTEENN 算法的差距在一个百分点以内；*G*-Mean 和

F-Measure 则为 0.954 6 和 0.955 2，超过其他算法，表明其在处理类别不平衡问题上具有良好的平衡能力。

以上实验结果表明，USCD 算法处理不平衡数据是可行且有效的。在处理效果上，USCD 在多个数据集上明显优于同为下采样算法的 ENN、NCR 和 RBU；和 SMOTEENN 以及 SMOTETomek 这种集成采样算法以及 SMOTE、ADASYN、K-means SMOTE 等上采样算法相比，USCD 在多个指标上的表现也更优，这证明了 USCD 算法的有效性。

5 总结

为有效清除不平衡数据集中多数类样本所存在的冗余和噪声样本，设计了一种基于 K-means 聚类和多距离度量的下采样方法，将数据集进行聚类以后，选择符合要求的簇进行下采样处理；在采样过程中使用多种距离从多维度衡量样本与簇内所有样本之间的亲疏关系，从而有效筛选其中的冗余和噪声样本。在四个真实数据集和五个分类器上的实验结果表明，USCD 算法是可行且有效的。

参考文献：

[1] 王轶群，王笑，高燕程 . 信用风险不平衡数据的表格生成对抗网络优化与分类 [J]. 计算机科学与探索 ,2026,20(2):561-573.

[2] 王天硕，高景伯，童盛军，等 . 面向不平衡数据的 SMOTE-LSTM 车辆事故检测方法 [J]. 交通信息与安全，2025, 43(1): 52-60.

[3] 李原，刘升 . 改进黑翅鸢算法优化 Catboost 的不平衡医保数据的欺诈检测 [J/OL]. 计算机工程与应用 ,1-14[2026-04-13].https://link.cnki.net/urlid/11.2127.tp.20250711.1655.007.

[4] 张光华，王子昱，蔡明伟 . 基于不平衡数据的物联网异常流量检测 [J]. 信息安全研究 ,2024,10(11):1012-1019.

[5] 周建民，夏晓枫，李家辉 . 改进生成对抗网络的不平衡数据下轴承故障诊断 [J]. 控制工程 ,2025,32(6):1039-1048.

[6]NIKPOUR B, RAHMATI F, MIRZAEI B, et al. A comprehensive review on data-level methods for imbalanced data classification[J]. Expert systems with applications, 2026, 295: 128920.

[7] 陈丽芳，白云，施永辉，等 . 面向不平衡数据的特征子空间增强的异质集成学习 [J]. 计算机工程与科学，2025, 47(5): 940-950.

[8]LIAN Z X, LI S Y, HUANG Q X, et al.UBMF: uncertainty-aware bayesian meta-learning framework for fault diagnosis with imbalanced industrial data[J]. Knowledge-based systems, 2025, 323: 113772.

[9] 王晓霞，李雷孝，林浩 .SMOTE 类算法研究综述 [J]. 计算机科学与探索 ,2024,18(5):1135-1159.

[10]DOUZAS G, BACAO F, LAST F. Improving imbalanced learning through a heuristic oversampling method based on K-means and SMOTE[J]. Information sciences, 2018, 465: 1-20.

[11]KOZIARSKI M. Radial-based undersampling for imbalanced data classification[J]. Pattern recognition, 2020, 102: 107262.

[12] 杨晓月 . 基于谱聚类的不平衡数据欠采样方法研究 [J]. 计算机与数字工程 ,2021,49(11):2305-2309.

[13] 杨涛 . 基于不平衡数据采样和卷积神经网络的信用评分模型研究 [D]. 西安 : 西北大学，2021.

[14] 杨云熙 . 基于 K-means 的不平衡数据过采样研究方法 [J]. 信息技术与信息化 ,2025(3):143-146.

[15] 陶维青，李雪婷，华玉婷，等 . 基于曼哈顿平均距离和余弦相似度的配网单相接地故障定位 [J]. 电力工程技术，2023, 42(2): 130-138.

[16] 李志明，唐永中 . 基于欧氏距离的 AES 算法模板攻击 [J]. 计算机工程与应用 ,2022,58(2):110-115.

[17] 李贝贝，彭力，戴菲菲 . 结合马氏距离与自编码器的网络流量异常检测方法 [J]. 计算机工程 ,2022,48(4):133-142.

[18]YE-BIN M, HYEON-WOO N, CHOI W, et al. SYNAuG: exploiting synthetic data for data imbalance problems[J]. Pattern recognition letters, 2025, 193: 115-121.

[19]TAO X M, GUO X Y, XU A N, et al. Majority data-based overlapping shift technique for imbalanced datasets classification with small disjuncts and outliers[J]. Expert systems with applications, 2025, 289: 128204.

[20]VAIRETTI C, ASSADI J L, MALDONADO S. Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification[J]. Expert systems with applications, 2024, 246: 123149.

[21] 黄文鹏 . 基于机器学习的银行信贷风控应用研究 [D]. 西安 : 西安石油大学 ,2024.

【作者简介】

杨涛（1993—），男，四川南充人，硕士研究生，高级系统分析师，研究方向：不平衡数据处理、机器学习、人工神经网络。

（收稿日期：2025-09-28 修回日期：2026-04-16）