文章编号：2095-2163(2022)06-0046-08 中图分类号：TP391 文献标志码：A

基于密度峰值聚类算法的自适应加权过采样算法

穆伟蒙[1]，宋 燕[2]，窦 军[2]
（1 上海理工大学 理学院，上海 200093；2 上海理工大学 光电信息与计算机工程学院，上海 200093）

摘 要：不平衡数据是监督学习中的一个挑战性问题。传统的分类器通常偏向多数类，忽略了少数类，而少数类样本往往包含很多重要信息，需要得到更多的关注。针对此问题，提出了一种基于密度峰值聚类算法的过采样技术（An Oversampling Technique based on Density Peak Clustering，DPCOTE）。DPCOTE 的主要思想是：（1）利用 k 近邻算法去除多数类和少数类噪声样本；（2）基于密度峰值聚类算法（Density peaks clustering algorithm，DPC）中的 2 个重要因子，即样本局部密度和样本到局部密度较高的最近邻的距离，来为每个少数类样本分配采样权重；（3）对于 DPC 算法中涉及到的距离，使用马氏距离来度量，以消除样本特征量纲不一致问题。最后，在 12 个 UCI 数据集上进行了对比实验，用不同的指标评价分类结果，结果表明本文提出的算法在处理不平衡分类问题时优于其它过采样方法。

关键词：不平衡数据；k 近邻算法；密度峰值聚类算法；马氏距离

An adaptive weighted oversampling algorithm based on density peak clustering

MU Weimeng[1]，SONG Yan[2]，DOU Jun[2]
（1 College of Science，University of Shanghai for Science and Technology，Shanghai 200093，China；2 School of Optical-Electrical and Computer Engineering，University of Shanghai for Science and Technology，Shanghai 200093，China）

【Abstract】Imbalanced data is a challenge in supervised learning. Traditional classifiers usually favor the majority class and ignore the minority class，while the minority class samples often contain more important information and need more attention. The oversampling algorithm based on density peak clustering（DPCOTE）is proposed to deal with imbalanced classification problem. The main idea of DPCOTE is as follows：（1）The k-nearest neighbor algorithm is used to remove noise samples of majority class and minority class；（2）Two important factors in the density peaks clustering algorithm（DPC），namely the local density of the sample and the distance of the sample to the nearest neighbor with high local density，are used to assign sample weights for each minority class sample；（3）The distance involved in DPC algorithm is measured by Mahalanobis distance to eliminate the inconsistency of sample feature dimensions. Finally，comparative experiments conducted on 12 UCI datasets with different indexes show that the proposed algorithm is superior to other oversampling methods in dealing with the imbalanced data.

【Key words】imbalanced data；k-nearest neighbor algorithm；density peak clustering algorithm；Mahalanobis distance

0 引 言

数据不平衡问题在许多应用，如医疗诊断、人脸识别和网络诈骗等领域[1-3]都受到了广泛关注。不平衡问题是指不同类别的样本数量差距很大，样本数量多的类别称为多数类，样本数量少的类别称为少数类。一般来说，少数类样本包含很多有用的信息，如果没有很好的分类，可能会付出很大的代价[4]。因此，提高少数类的识别精度至关重要[5]。

解决不平衡问题的方法可以分为 2 类：基于数据的和基于算法的。其中，算法层面的策略包括代价敏感学习[6]、单类学习[7]、集成学习[8]等，主要通过修改现有算法来提高对少数类样本的分类精度。数据层面的策略包括过采样技术和欠采样技术[9-10]，通过调节多数类或者少数类的样本数量使不同类别的样本趋于平衡。总地说来，欠采样技术能够减少多数类样本来使类趋于平衡，容易实现，但易造成有用信息的丢失。而过采样技术既能使不同类别样本达到平衡，又能保留原始数据的分布特点，所以过采样在处理不平衡数据分类方面得到了更多的关注。

由于过采样技术应用更为广泛，因此有学者提出了许多过采样方法，如，为了解决随机过采样技术可能会造成的过拟合问题，Chawla 等人[11]提出了合成少数类过采样技术（Synthetic minority oversampling technique，SMOTE），其原理为：对于任

基金项目：国家自然科学基金（62073233，61873169）。

作者简介：穆伟蒙（1995-），女，硕士研究生，主要研究方向：不平衡数据分类；宋 燕（1979-），女，博士，教授，博士生导师，CCF 高级会员（No. 93073SM），主要研究方向：大数据算法、图像处理、预测控制；窦 军（1994-），男，博士研究生，主要研究方向：不平衡数据的分类。

通讯作者：宋 燕 Email：sonya@usst.edu.cn

收稿日期：2021-12-22

意一个目标少数类样本 x_i，利用欧式距离随机选取 x_i 的其中一个近邻样本 x_j，通过线性插值，人工合成样本 x_{syn}，即：

$$x_{syn} = x_i + \alpha(x_j - x_i) \tag{1}$$

其中，$\alpha \in [0,1]$。

虽然 SMOTE 在一定程度上克服了过拟合问题，并解决了类间不平衡，但是 SMOTE 合成样本时，对于所有的少数类样本，采用统一的样本分配策略合成新的样本，很容易造成类内不平衡，改变原始数据的分布。

为了解决上述问题，学者提出了加权过采样方法，为不同的子簇或者样本分配不同的权重，来解决类间不平衡和类内不平衡问题。He 等人[12]提出了自适应合成过采样（ADASYN）方法，来对每个少数类样本赋予不同的权值，而权值越大，学习难度就越大。Nekooeimehr 等人[13]提出自适应半无监督加权过采样方法（A-SUWO），通过利用分类复杂度和交叉验证来自适应地确定每个子簇的过采样大小。Douzas 等人[14]提出基于 K 近邻（KNN）过采样算法（SMOM）来给每个目标样本的近邻分配选择权重，对可能会产生过度泛化的方向赋予较小的选择权重。此外，为了增强边界少数类样本的学习，安全水平过采样（Safe-Level-SMOTE）算法[15]、边界过采样（Borderline-SMOTE）算法[16]和多数加权少数的过采样[17]（MWMOTE）即已陆续提出。虽然如上研究通过不同的方法对少数类样本赋予一定的权重[18]，但却没有充分考虑少数类样本权重分配所必须的因素，如样本间的相似性、样本分布特点等，这也是本文的主要研究背景。

针对上述问题，本文提出了一种基于密度峰值聚类算法的自适应加权过采样算法（DPCOTE）来解决不平衡分类问题。该方法核心思想为：

（1）利用 k 近邻算法去除多数类和少数类噪声样本。

（2）基于密度峰值聚类算法中的重要因子，为每个少数类样本赋予采样权重，以此来为少数类样本合成不同数量的新样本。

（3）在 DPC 算法中，引入马氏距离，来消除样本特征间量纲不一致的问题。

1　基于密度峰值加权过采样方法

1.1　马氏距离

马氏距离是由印度统计学家 Mahalanobis 提出的[19]，马氏距离考虑了各个特征变量之间的联系，且不受特征量纲不一致的干扰。马氏距离与欧氏距离的关系示意如图 1 所示。由图 1 可知，在计算欧式距离时，A 与 C 距离最近，但是在马氏距离中，A 与 B 距离最近，因为原始数据呈现椭圆分布，欧氏距离没有考虑数据分布。马氏距离除以协方差矩阵，可以把各个分量之间的方差都除掉，消除了量纲性，详见图 1(b)。

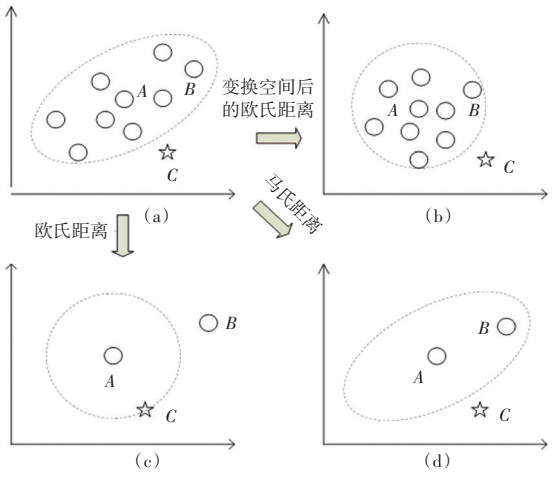


图 1　马氏距离与欧氏距离示意图

Fig. 1　The schematic diagram of Mahalanobis distance and Euclidean distance

对于任意 2 个样本点 $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iN})^{\mathrm{T}}$ 和 $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jN})^{\mathrm{T}}$，其中 N 为样本的特征数量，则样本之间的欧式距离为：

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{x}_j)} \tag{2}$$

马氏距离表示为：

$$D_M(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)} \tag{3}$$

其中，$\boldsymbol{\Sigma}$ 为样本的协方差矩阵，计算公式为：

$$\boldsymbol{\Sigma} = \frac{\sum_{k=1}^{N}(x_{ik} - \bar{\boldsymbol{x}}_i)^{\mathrm{T}}(x_{jk} - \bar{\boldsymbol{x}}_j)}{N - 1} \tag{4}$$

如果协方差矩阵是单位矩阵，则马氏距离等同于欧氏距离。

1.2　密度峰值聚类算法

密度峰值聚类算法（Density peaks clustering algorithm，DPC）由 Rodriguez 等人于 2014 年提出[20]。该算法无须迭代就可确定聚类中心，且能够识别任意形状的类簇，目前已经得到了广泛的应用。DPC 算法的核心思想建立在 2 个基本假设上：

（1）聚类中心被局部密度较低的邻域点包围。

（2）密度较高的点之间的距离相对较大。

基于这 2 个假设,DPC 引入了 2 个重要因子,即目标样本的局部密度 ρ_i 和相对距离 δ_i。对于第一个假设,利用高斯核函数计算任一样本点 x_i 的局部密度 ρ_i,其值可由如下公式计算得出:

$$\rho_i = \sum_{j \neq i} e^{-\left(\frac{d_{ij}}{d_c}\right)^2} \tag{5}$$

其中,d_{ij} 为样本 x_i 和 x_j 之间的距离,d_c 为截断距离,通常将其设为距离降序排列的 1%~2%。

DPC 算法示意如图 2 所示。在确定了截断距离后,就可以得到目标样本的局部密度,如样本点 A,D,E。对于第二个假设,通过计算相对距离,即对于任一样本点 x_i,其局部密度比其更大、且距离最近的样本点 x_j 的距离 δ_i 可表示为:

$$\delta_i = \min_{j:\rho_i<\rho_j}(d_{ij}) \tag{6}$$

如,对于样本点 C,局部密度比该点要大、且距离最近的样本点为样本点 C,则 D 的相对距离即为 CD 的距离。对于局部密度最大的数据点 x_i,则 $\delta_i = \max_j(d_{ij})$。

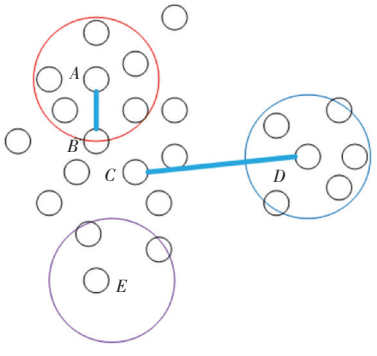


图 2 DPC 算法示意图

Fig. 2 The schematic diagram of DPC

在计算出所有样本的因子后,如果样本的 ρ_i 和 δ_i 足够大,其附近样本分布较为密集,则将其视为密度峰值。

1.3 DPCOTE 方法

在本节,提出了新的基于密度峰值聚类算法的过采样算法(DPCOTE)。该算法中,使用马氏距离代替 DPC 算法涉及到的欧氏距离。该算法主要步骤可阐释分述如下:

(1) 去噪。在数据预处理阶段,使用 k 近邻算法去除噪声样本。在此阶段中,对所有的样本使用 k 近邻算法。先是计算目标样本与近邻样本的距离,找到目标样本的 k 个近邻。如果目标样本的 k 个近邻样本的类标签与目标样本的类标签都不一样,则将目标样本归为噪声样本,并删除。

(2) 合成样本。利用 DPC 算法对所有少数类样本赋予采样权重,来确定每个少数类样本需要合成的样本数,并使用 k 近邻算法和线性插值来对每个少数类样本合成新样本。

和传统的 DPC 算法不同的是,本文在计算任意 2 个样本的距离时,使用马氏距离代替欧氏距离,这样就解决了特征间量纲不一问题。所以,利用上述描述的 DPC 算法,基于马氏距离,可以得到每个少数类样本的局部密度 ρ_i 和到局部密度较高的最近邻的距离 $\delta_i(i=1,2,\cdots,n)$,此处的 n 表示少数类样本数。

下面,利用 ρ_i 和 δ_i 来确定每个少数类样本的采样权重。为此,先对这 2 个因子做归一化,即:

$$\rho_i = \frac{\rho_i - \min(\rho_i)}{\max(\rho_i) - \min(\rho_i)}, \tag{7}$$

$$\delta_i = \frac{\delta_i - \min(\delta_i)}{\max(\delta_i) - \min(\delta_i)} \tag{8}$$

综合上述 2 个因子,考虑到每个少数类样本的密度信息和相对距离信息,为此构造一个新的因子,即:

$$\varepsilon_i = \rho_i * \delta_i \tag{9}$$

事实上,如果样本的密度较大,基于该样本合成新样本时,会生成较多重复的样本,导致模型过拟合。所以,每个少数类样本需要合成的样本数与密度成反比,具体数学公式如下:

$$w_i = \frac{1}{\varepsilon_i} \tag{10}$$

将其标准化来确定第 i 个少数类样本的采样权重为:

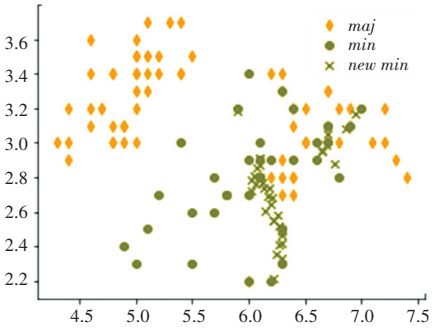
$$w'_i = \frac{w_i}{\sum_{j=1}^{n} w_j} \tag{11}$$

若给定 G 为需要合成的少数类样本总数,则第 i 个少数类样本需要合成的样本数可以通过下式得到:

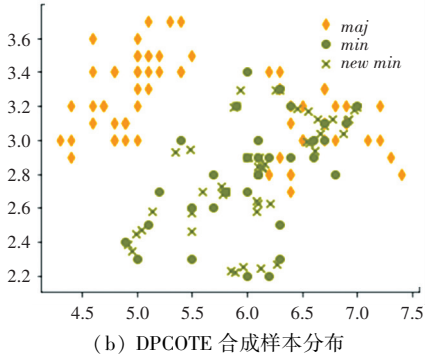
$$g_i = w'_i * G \tag{12}$$

确定每个少数类样本的合成数后,利用 k 近邻算法和线性插值来合成新的样本,使少数类样本与多数类样本达到相对平衡。图 3 为 ADASYN 算法和本文提出 DPCOTE 算法生成的样本分布示意图。图 3 中,*maj* 表示多数类样本,*min* 表示少数类样本,*new min* 表示新合成的少数类样本。由于 ADASYN 算法对于学习难度高的样本赋予更高的权重,所以其在边界附近合成了更多的样本,容易模糊类边界,DPCOTE 算法考虑每个少数类样本的分布

情况，在不改变原始数据分布的情况下，生成更多有用的新样本。图 4 给出了 DPCOTE 算法的流程图，相应算法的伪代码设计表述具体如下。



（a）ADASYN 合成样本分布



（b）DPCOTE 合成样本分布

图 3　合成样本分布示意图

Fig. 3　The schematic diagram of synthetic sample distribution

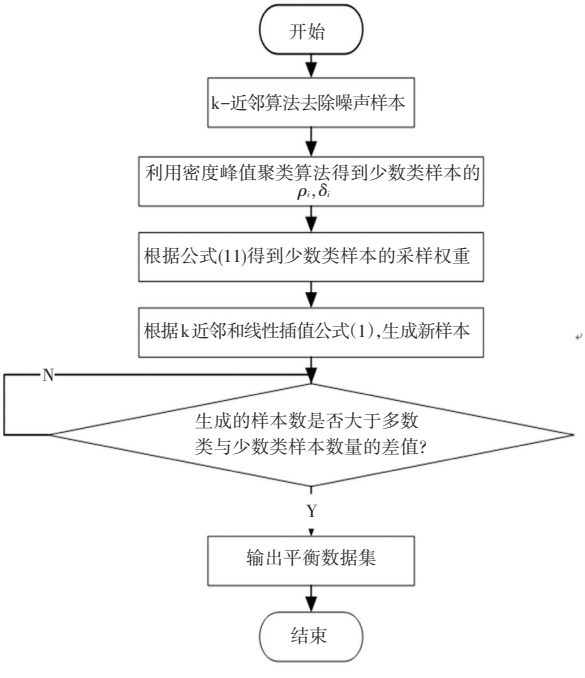


图 4　DPCOTE 算法流程图

Fig. 4　Flow chart of DPCOTE

算法　DPCOTE 算法

输入：训练数据集 D

输出：平衡数据集 D_{new}

1：计算所有训练样本的 k 近邻，如果目标样本的 k 近邻类标签与目标样本类标签都不一样，则将其删除。

2：获得去除噪声后的数据集 D_1。其中，少数类样本数为 $l_{\min}$，多数类样本数为 $l_{\max}$。

3：计算要合成的样本总数 $G = l_{\max} - l_{\min}$。

4：for i = 1 to $l_{\min}$

5：用式(3)、式(5)、式(6)计算 ρ_i, δ_i；

6：用式(9)~(11)计算第 i 个少数类样本采样权重 w'_i；

7：用式(12)计算第 i 个少数类样本合成样本数 g_i；

8：用式(1)生成新样本；

9：end for

2　实验结果及分析

2.1　数据集

为了更加全面地验证 DPCOTE 算法的性能，本文从 UCI 机器学习库中选取了 12 组二类不平衡数据集，这些数据集样本数量和特征数量都不同，且不平衡率的范围为 2.78~22.7。表 1 为本文选用的数据集。

表 1　数据集信息

Tab. 1　Information of the datasets

数据集	样本数	特征数	少类数	不平衡率
haberman	306	3	81	2.78
yeast4	1 484	8	244	5.08
abalone1	4 177	8	487	7.58
yeast1	1 484	8	163	8.10
ecoli2	336	7	35	8.60
yeast2	1 484	8	115	11.90
ecoli1	336	7	25	12.44
libras	360	90	24	14.00
pageblocks2	5 473	10	329	15.64
yeast3	1 484	8	87	16.06
abalone2	4 177	8	229	17.24
pageblocks1	5 473	10	231	22.70

2.2　评价指标

在不平衡分类问题中，分类器通常偏向多数类样本，不能反映少数类的分类精度，而少数类的识别精度往往很重要，因此分类精度不适用于不平衡数据。$F-measure$ 和 $G-mean$ 通常用来评价模型的

性能,此处需涉及的数学公式可写为:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad Specifity = \frac{TN}{TN + FP} \tag{13}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

$$G - mean = \sqrt{Recall \times Specificity} \tag{15}$$

其中,TP 表示预测和真实都为少类的样本数;FN 表示预测与真实都为多类的样本数;FP 表示少类预测为多类的样本数;FN 表示多类预测为少类的样本数。

2.3 实验分析

为了验证本文提出的采样方法的有效性,将 SMOTE、Safe - Level - SMOTE(SLS)、Borderline - SMOTE(BS)、ADASYN、CBSO 与本文提出的 DPCOTE 算法进行了对比实验。此外,使用逻辑回归(LR)和支持向量机(SVM)两个分类器来验证 DPCOTE 算法的泛化能力。所描述的实验均采用 5 折交叉验证,每组数据重复 5 次,记录每个评估指标的平均值,以消除数据随机分组时可能出现的偏差。最好的结果以粗体字突出显示。每次实验都在 2.9 GHz CPU、8 GB 内存的电脑上进行,软件环境是 Python 3.7。其中,F,G 是 $F - measure$ 和 $G - mean$ 的缩写。

表 2 显示了使用 LR 分类器,所提出的 DPCOTE 算法在 $F - measure$ 和 $G - mean$ 方面与典型对比算法之间的性能比较。由表 2 可知,DPCOTE 算法的表现远远好于对比的过采样方法。具体来说,在指标 $F - measure$ 方面,12 个数据集中,DPCOTE 算法有 9 个数据集取得了最好的结果;在指标 $G - mean$ 方面,有 7 个数据集取得了最好的结果。

表 2 在 LR 分类器上的对比结果

Tab. 2 Comparison results on LR

数据集	指标	SMOTE	SLS	BS	ADASYN	CBSO	DPCOTE
haberman	F	0.787 92	**0.846 64**	0.784 04	0.785 99	0.780 91	0.832 68
	G	0.777 03	0.733 33	0.791 80	0.794 12	**0.794 83**	0.793 13
yeast4	F	0.886 54	0.877 11	0.883 78	0.876 91	0.876 62	**0.890 56**
	G	0.882 23	0.870 86	0.885 04	0.879 43	0.882 63	**0.940 15**
abalone1	F	0.984 99	0.984 08	**0.987 59**	0.984 39	0.983 88	0.983 35
	G	0.986 46	0.982 13	**0.986 93**	0.984 12	0.984 05	0.978 11
yeast1	F	0.944 06	0.932 53	0.940 82	0.941 64	0.942 52	**0.952 53**
	G	0.936 92	0.930 91	0.940 18	0.929 76	0.935 94	0.981 41
ecoli2	F	0.965 25	0.944 88	0.963 08	0.964 24	0.964 43	**0.969 13**
	G	**0.968 12**	0.947 85	0.958 09	0.950 73	0.966 94	0.966 70
yeast2	F	0.951 51	0.943 87	0.956 63	0.946 22	0.951 79	**0.966 12**
	G	0.950 67	0.942 08	0.956 16	0.955 62	0.951 54	**0.982 03**
ecoli1	F	**0.983 66**	0.961 36	0.979 09	0.969 49	0.971 63	0.973 55
	G	0.980 82	0.961 90	0.986 28	0.979 71	0.980 14	**0.998 31**
libras	F	0.975 55	0.980 56	0.980 22	0.976 99	0.978 69	**0.983 33**
	G	0.973 98	0.980 43	0.987 56	0.979 04	0.984 12	**0.998 27**
pageblocks2	F	0.961 65	0.959 32	0.965 53	0.961 39	0.960 68	**0.978 82**
	G	0.956 97	0.961 34	**0.965 34**	0.963 70	0.962 12	0.963 33
yeast3	F	0.948 13	0.951 90	0.947 29	0.949 98	0.950 01	**0.959 92**
	G	0.944 14	0.956 94	0.942 25	0.953 58	0.949 95	**0.978 99**
abalone2	F	0.945 99	0.950 64	0.948 43	0.944 89	0.944 54	**0.954 60**
	G	0.946 06	0.949 31	0.954 84	0.942 48	0.945 43	**0.970 24**
pageblocks1	F	0.968 53	0.964 46	0.968 31	0.964 40	0.964 95	**0.979 71**
	G	0.969 04	0.962 91	**0.969 34**	0.964 80	0.966 14	0.963 77

图 5 为使用 LR 分类器，数据 yeast4 在指标 *F* − *measure* 和 *G* − *mean* 上的箱线图结果，箱线图包括一个矩形箱体和上下 2 条线，箱体中间的线为中位线，上、下限分别为数据的上四分位数和下四分位数，箱子的宽度可以体现数据的波动程度，箱体的上、下方各有一条线是数据的最大、最小值，超出最大、最小值线的数据为异常数据。从图 5(a)中可以看出，虽然 DPCOTE 算法数据波动较大，但数据的中值和上、下四分位数是优于对比算法的。在图 5(b)中，DPCOTE 算法的中值和上、下四分位数是相对较好的，在箱体宽度方面，除了 ADASYN 算法，DPCOTE 算法的数据波动优于其它方法，但是 ADASYN 算法存在异常值。图 6 为使用 SVM 的可视化，结果显示 DPCOTE 算法的中值和上、下四分位数是大于对比算法的。

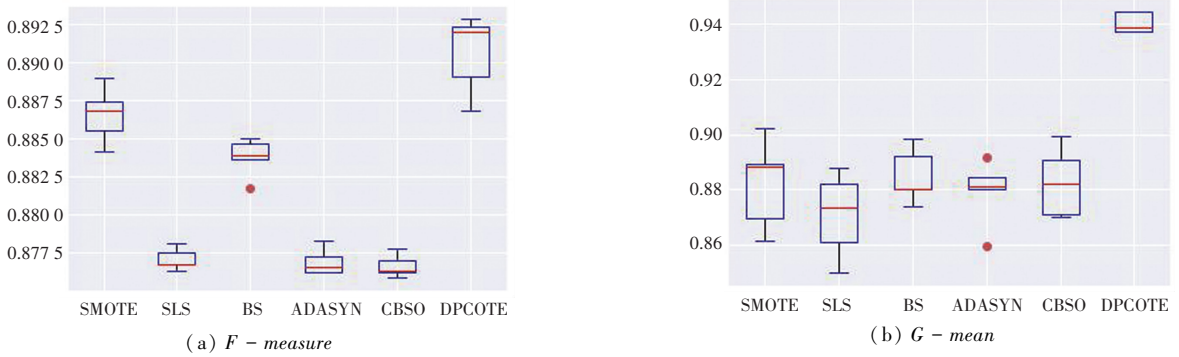


图 5　使用 LR 分类器数据 yeast4 的箱线图

Fig. 5　A boxplot using LR on yeast4

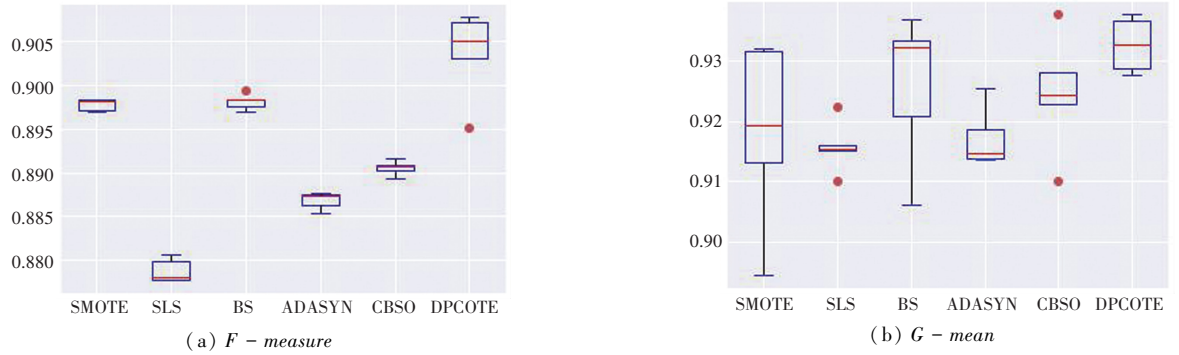


图 6　使用 SVM 分类器数据 yeast4 的箱线图

Fig. 6　A boxplot using SVM on yeast4

为了全面对比本文提出的算法与其他采样方法在性能上的有效性，研究中使用了 Wilcoxon 符号秩检验来评估 DPCOTE 算法与对比算法之间是否有显著性差异。表 3 为使用 LR，SVM 分类器，在 *F* − *measure* 和 *G* − *mean* 的 Wilcoxon 符号秩检验的结果，其中 *R* + 表示 DPCOTE 算法的秩和，*R* − 表示相应对比方法的秩和。从表 3 中可以观察到，在使用 LR 分类器、显著性水平为 0.05 的情况时，除了 DPCOTE 算法与 Borderline−SMOTE 在 *G* − *mean* 对比的 *p* 值大于 0.05 以外，大部分原假设都被拒绝，而且 *R* + 的值远大于 *R* −，说明 DPCOTE 算法和其他采样方法相比有显著性差异。从表 3 可以看出，在使用 SVM 分类器时，DPCOTE 算法在 *F* − *measure* 和 *G* − *mean* 方面的表现好于对比算法。使用 LR，SVM 分类器的 Wilcoxon 实验结果见表 4。使用 SVM 分类器的 Wilcoxon 检验的结果显示，除了 DPCOTE 算法与 Borderline−SMOTE 在 *G* − *mean* 下接受原假设外，所有的原假设都被拒绝，表明 DPCOTE 算法显著优于其他对比算法。

表 3 在 SVM 分类器上的对比结果

Tab. 3 Comparison results on SVM

数据集	指标	SMOTE	SLS	BS	ADASYN	CBSO	DPCOTE
haberman	*F*	0.843 70	0.843 29	0.841 82	**0.843 83**	0.840 01	0.834 81
	G	0.825 64	0.795 33	0.858 37	0.840 88	0.838 89	**0.871 14**
yeast4	*F*	0.897 71	0.878 78	0.898 10	0.886 84	0.890 53	**0.903 62**
	G	0.918 09	0.915 77	0.925 93	0.917 24	0.924 61	**0.932 72**
abalone1	*F*	0.988 21	0.994 05	**0.992 40**	0.986 76	0.988 55	0.988 50
	G	0.988 21	**0.994 05**	0.992 39	0.986 75	0.988 55	0.987 37
yeast1	*F*	0.955 99	0.938 72	0.958 79	0.955 37	0.952 19	**0.968 99**
	G	**0.970 29**	0.963 91	0.969 52	0.970 72	0.962 70	0.969 96
ecoli2	*F*	0.968 16	0.959 88	0.968 77	0.967 20	0.968 56	**0.970 25**
	G	0.979 89	0.978 52	0.983 58	0.957 88	**0.984 97**	0.981 85
yeast2	*F*	0.953 63	0.947 74	0.958 86	0.945 19	0.954 73	**0.974 62**
	G	0.961 65	0.965 87	0.966 74	0.969 36	0.967 42	**0.975 78**
ecoli1	*F*	0.989 70	0.987 04	0.989 64	0.988 60	0.988 92	**0.990 32**
	G	0.989 51	**0.996 76**	0.993 91	0.993 29	0.996 65	0.991 87
libras	*F*	0.984 59	0.985 22	0.986 45	0.984 86	0.984 01	**0.987 69**
	G	0.994 34	0.989 11	0.997 12	0.997 03	0.995 46	**0.998 81**
pageblocks2	*F*	0.969 03	0.965 27	0.972 07	0.964 44	0.964 82	**0.983 65**
	G	0.974 98	0.971 50	0.983 79	0.979 33	0.979 06	**0.986 10**
yeast3	*F*	0.948 47	0.949 40	0.949 17	0.948 12	0.947 00	**0.963 51**
	G	0.956 03	0.967 59	0.958 07	0.963 53	0.963 05	**0.968 49**
abalone2	*F*	0.940 52	0.944 98	0.946 83	0.941 69	0.940 71	**0.956 52**
	G	0.959 69	0.963 47	0.965 71	0.956 41	0.961 19	**0.967 38**
pageblocks1	*F*	0.971 07	0.967 70	0.972 16	0.966 94	0.968 14	**0.981 92**
	G	0.978 03	0.977 82	0.977 68	0.976 61	0.975 29	**0.985 59**

表 4 使用 LR,SVM 分类器的 Wilcoxon 实验结果

Tab. 4 Wilcoxon experimental results on LR,SVM

分类器	对比方法	*F*				*G*			
		$R+$	$R-$	p − value	$\alpha = 0.05$	$R+$	$R-$	p − value	$\alpha = 0.05$
LR	DPCOTE vs SMOTE	70	8	0.015	**rejected**	69	9	0.018	**rejected**
	DPCOTE vs SLS	70	8	0.015	**rejected**	74	4	0.006	**rejected**
	DPCOTE vs BS	73	5	0.007	**rejected**	60	18	0.099	not rejected
	DPCOTE vs ADASYN	77	1	0.002	**rejected**	68	10	0.022	**rejected**
	DPCOTE vs CBSO	77	1	0.002	**rejected**	65	13	0.041	**rejected**
SVM	DPCOTE vs SMOTE	72	6	0.009	**rejected**	75	3	0.004	**rejected**
	DPCOTE vs SLS	71	7	0.012	**rejected**	68	10	0.022	**rejected**
	DPCOTE vs BS	68	10	0.022	**rejected**	62	16	0.071	not rejected
	DPCOTE vs ADASYN	73	5	0.007	**rejected**	73	5	0.007	**rejected**
	DPCOTE vs CBSO	72	6	0.009	**rejected**	71	7	0.012	**rejected**

3 结束语

本文提出了一种基于密度峰值聚类算法的自适应加权过采样算法、即 DPCOTE 算法来解决不平衡分类问题。DPCOTE 算法的基本思想为:考虑了类内不平衡问题,利用密度峰值聚类算法中的 2 个重要因子,为每个少数类样本赋予采样权重,从而使每个少数类样本合成不同数量的新样本。同时,在 DPC 算法中,引入马氏距离代替欧氏距离,消除特征间量纲不一致的问题。为了验证该算法的有效性,在 *F* − *measure* 和 *G* − *mean* 指标下,使用 LR 和 SVM 分类器进行了对比试验,且使用 Wilcoxon 检验对结果进行分析。试验结果表明,DPCOTE 算法在 12 个大小、不平衡率不同的数据集上取得了较好的结果。

参考文献

[1] MENA L J, GONZALEZ J A. Machine learning for imbalanced datasets: application in medical diagnostic [C]//Proceedings of the 19th International FLAIRS Conference. Melbourne Beach, Florida, USA:dblp, 2006: 574-579.

[2] LIU Y H, CHEN Y T. Face recognition using total margin-based adaptive fuzzy support vector machines[J]. IEEE Transactions on Neural Networks, 2007, 18 (1): 178-192.

[3] FIORE U, De SANTIS A, PERLA F, et al. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection[J]. Information Sciences, 2019, 479: 448-455.

[4] 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(01): 135-144.

[5] NEKOOEIMEHR I, LAI-YUEN S K. Cluster-based weighted oversampling for ordinal regression (cwos - ord) [J]. Neurocomputing, 2016, 218:51-60.

[6] SIERS M J, ISLAM M Z. Novel algorithms for cost-sensitive classification and knowledge discovery in class imbalanced datasets with an application to NASA software defects [J]. Information Sciences, 2018, 459: 53-70.

[7] LUCA S, CLIFTON D A, VANRUMSTE B. One - class classifification of point patterns of extremes [J]. Journal of Machine Learning Research, 2016, 17(1): 6581-6601.

[8] GALAR M, FERNANDEZ A, BARRENECHEA E, et al. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2012, 42(4): 463-484.

[9] 林静怀,刘治宇,李军良,等. 面向不平衡数据分类的高维超球体过采样方法[J]. 微电子学与计算机, 2021, 38(05): 65-72.

[10] XIE Xiaoying, LIU Huawen, ZENG Sshouzhen , et al. A novel progressively undersampling method based on the density peaks sequence for imbalanced data[J]. Knowledge-Based Systems, 2021, 213:106689.

[11] 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 (1): 321-357.

[12] HE Haibo, BAI Yang, GARCIA E A,et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning [C]//2008 IEEE International Joint Conference on Neural Networks. Hong Kong:IEEE, 2008: 1322-1328.

[13] NEKOOEIMEHR I, S. K.LAI-YUEN I S K, Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets[J]. Expert Systems with Applications, 2016, 46: 405-416.

[14] DOUZAS G, BACAO F. Self-organizing map oversampling (SOMO) for imbalanced data set learning[J]. Expert Systems with Applications, 2017, 82: 40-52.

[15] BUNKHUMPORNPAT C, SINAPIROMSARAN K, LURSINSAP C. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem [M]//Theeramunkong T, KIJSIRIKUL B, CERCONE N, et al. Advances in knowledge discovery and data mining. PAKDD 2009. Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, 2009,5476: 475-482.

[16] HAN Hui, WANG Wenyuan, MAO Binghuan. Borderline-smote: a new over-sampling method in imbalanced data sets learning[M]//HUANG D S, ZHANG X P, HUANG G B. Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science. Berlin/ Heidelberg: Springer , 2005, 3644: 878-887.

[17] BARUA S, ISLAM M M, YAO Xin, et al. Mwmote-majority weighted minority oversampling technique for imbalanced data set learning [J]. IEEE Transactions on Knowledge and Data Engineering, 2014,26 (2): 405-425.

[18] 李艳霞, 柴毅, 胡友强, 等. 不平衡数据分类方法综述[J]. 控制与决策, 2019, 34 (04): 4-19.

[19] MAHALANOBIS P C. On the generalized distance in statistics [J]. Proceedings of the National Instute of Science (India), 1936, 2: 49-55.

[20] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science(6191), 2014, 344: 1492- 1496.

(上接第 45 页)

[15] 李创新,马耀峰,张颖,等. 时空二元视角的入境旅游流集散空间场效应与地域结构——以丝路东段典型区为例[J]. 地理科学, 2012, 32(02): 176-185.

[16] 温馨,高维新,朱金勋. 粤港澳大湾区城市时空演变测度及协同发展研究[J]. 统计与决策, 2021, (11): 108-111.

[17] 刘亚萍,于杰,王富强. 中国赴东盟旅游流重心移动轨迹及旅游市场态分析[J]. 旅游科学, 2019, 33(04): 85-95.

[18] 赵书虹,白梦. 云南省品牌旅游资源竞争力与旅游流耦合协调特征及其影响因素分析[J]. 地理科学, 2020, 40(11): 1878-1888.

[19] 曹芳东,黄震方,黄睿,等. 江苏省高速公路流与景区旅游流的空间关联及其耦合路径[J]. 经济地理, 2021, 41(01): 232-240.

[20] 郭向阳,穆学青,明庆忠,等. 旅游地快速交通优势度与旅游流强度的空间耦合分析[J]. 地理研究, 2019, 38(05): 1119-1135.

[21] 李涛,王姣娥,黄洁. 基于腾讯迁徙数据的中国城市群国庆长假城际出行模式与网络特征[J]. 地球信息科学学报, 2020, 22(06): 1240-1253.

[22] SALAS-OLMEDO M H, MOYA-GÓMEZ B, GARCÍA-PALMOARES J C, et al. Tourists′ digital footprint in cities: Comparing big data sources[J]. Tourism Management, 2018, 6613-25.

[23] 马丽君,肖洋. 典型城市居民国内旅游流网络结构特征[J]. 经济地理, 2018, 38(02): 197-205.

[24] 马莉,刘培学,张建新,等. 景区旅游流与网络关注度的区域时空分异研究[J]. 地理与地理信息科学, 2018, 34(02): 87-93.

[25] GAO Yong, YE Chao, ZHONG Xiang, et al. Extracting spatial patterns of intercity tourist movements from online travel blogs [J]. Sustainability, 2019, 11(13):1-18.

[26] ZHENG Yunhao, MOU Naixia, ZHANG Lingxian, et al. Chinese tourists in Nordic countries: An analysis of spatio-temporal behavior using geo-located travel blog data [J]. Computers Environment and Urban Systems, 2021, 85:101561.

[27] MOU Naixia, YUAN Rongzheng, YANG Tengfei, et al. Exploring spatio-temporal changes of city inbound tourism flow: The case of Shanghai, China[J]. Tourism Management, 2020, 76:103955.