山西大学学报(自然科学版)48(6):1092—1102,2025
Journal of Shanxi University(Nat. Sci. Ed.)

DOI:10.13451/j.sxu.ns.2025017

基于密度峰值的粒球邻域粗糙集

朱学勤[1],邵亚斌[1,2*],华有霖[1]
(1.重庆邮电大学 理学院,重庆 400065;
2.网络空间大数据智能安全教育部重点实验室,重庆 400065)

摘　要:属性约简是数据分析和建模中常用的技术之一。粒球邻域粗糙集能够自适应设置邻域半径,提高了属性约简的准确性和鲁棒性。然而,当前的粒球生成方法存在数量不确定和分布不稳定的问题。针对这一问题,本文提出了基于密度峰值的粒球生成方法,采用密度峰值点和质心最近点作为球心,确保球心由样本点构成的同时增强了粒球可解释性。在此基础上,使用新的粒球生成方法推导出基于密度峰值的粒球邻域粗糙集模型,突破了粒球邻域粗糙集使用生成正域进行属性约简的局限性,设计了后向属性约简算法。针对上述算法,在多个数据集上进行测试。实验结果表明,相较于现有方法,新模型在粒球生成的过程中实现稳定,且约简后的属性显著提高了分类能力。

关键词:粒计算;多粒度粒球计算;邻域粗糙集;密度聚类;属性约简
中图分类号:O436　　**文献标志码**:A　　**文章编号**:0253-2395(2025)06-1092-11

Granular-ball Neighborhood Rough Set Based on Density Peak

ZHU Xueqin[1], SHAO Yabin[1,2*], HUA Youlin[1]
(1. *School of Science, Chongqing University of Posts and Telecommunications, Chongqing* 400065*, China;*
2. *Key Laboratory of Cyberspace Big Data Intelligent Security, Ministry of Education, Chongqing* 400065*, China*)

Abstract: Attribute reduction is one of the commonly used techniques in data analysis and modeling. The granular-ball neighborhood rough set, which can adaptively set the neighborhood radius, enhances the accuracy and robustness of attribute reduction. However, current granular-ball generation methods face problems of uncertain numbers and unstable distributions. To address this issue, this paper proposed a granular-ball generation method based on density peaks. By using density peak points and the nearest centroid points as centers, this method ensures that the centers are composed of sample points, thereby enhancing the interpretability of granular-balls. Based on this new granular-ball generation method, a granular-ball neighborhood rough set model based on density peaks was derived. This model overcomes the limitation of using the positive region for attribute reduction in granular-ball neighborhood rough sets. And accordingly a backward attribute reduction algorithm was designed. The above algorithm was tested on multiple datasets. Experimental results show that, compared to existing methods, the new model achieves stable performance during the granular-ball generation process, and the reduced attributes significantly enhance classification performance.

Key words: granular computing; multi-granularity granular-ball computing; neighborhood rough set; density clustering; attribute reduction

收稿日期:2024-11-19;**接受日期**:2025-02-20
基金项目:国家自然科学基金(12061067;62176033);重庆市自然科学基金(CSTB2023NSCQ-MSX0707)
作者简介:朱学勤(1998—),男,甘肃兰州人,硕士,研究方向为不确定性数学。E-mail:s220603023@stu.cqupt.edu.cn
* **通信作者**:邵亚斌(SHAO Yabin),E-mail:shaoyb@cqupt.edu.cn
引文格式:朱学勤,邵亚斌,华有霖.基于密度峰值的粒球邻域粗糙集[J].山西大学学报(自然科学版),2025,48(6):1092-1102. DOI:10.13451/j.sxu.ns.2025017.

0 引言

粗糙集理论[1]是一种处理不确定和不完备信息的重要工具,自1982年由Pawlak教授提出以来,已被广泛应用于知识发现的各个领域。但经典粗糙集理论对数据的划分建立在等价关系的基础上,无法处理连续的数值型数据[2]。

针对这一问题,胡清华教授利用球形邻域的概念,构造了邻域粗糙集(Neighborhood Rough Set, NRS)模型[3-4]。邻域粗糙集可以直接处理数值型数据,同时借助邻域的拓扑结构,邻域粗糙集在处理空间数据、时序数据以及其他具有显著邻域特性的数据集方面表现出色[5-6]。在此基础上,胡清华等[7]设计了基于前向搜索的属性约简算法。并且在面对异构的特征子集时,根据误分类样本所占的百分比,提出邻域决策错误率[8],这是一种能同时适用于分类特征和数值特征的特征评价方法。

同时,邻域粗糙集在多样化的数据集上也得到了广泛的应用。对于不平衡数据,Chen等[9]考虑了类别分布不均衡的因素,利用可辨矩阵实现了邻域粗糙集对这类数据的特征选择。在多标记数据方面,段洁等[10]重新定义了邻域粗糙集的下近似和依赖度,将单标记的邻域粗糙集模型推广为多标记邻域粗糙集模型,提出了多标记分类任务的特征选择算法。而针对现实生活中属性重要度不同的问题,Hu等[11]使用相关系数为属性分配权重,提出了加权邻域粗糙集。

然而,邻域粗糙集的邻域半径需要手动设置为一个固定值,这会导致邻域间的重叠现象与单粒度划分的局限性。针对这一问题,Xia等[12]引入了粒球计算的思想,提出粒球邻域粗糙集(Granular-ball Neighborhood Rough Set, GBNRS)模型。这使得邻域半径能够自适应地根据数据的分布和特性进行设置,比传统的邻域粗糙集模型更高效。此后,Xia等[13]结合经典粗糙集和邻域粗糙集,建立了粒球粗糙集模型。这一模型保证了邻域粗糙集的知识表示能力,并且消除了邻域粗糙集的“异类传递”现象。

近年来,粒球计算在人工智能的许多应用领域取得了丰富的成果,已系统构建包括粒球分类器[14-15]、粒球聚类[16-17]、粒球采样[18],以及粒球模糊粗糙集[19]在内的完整理论体系,并衍生出各种计算模型。与传统的人工智能相比,粒球计算的方法能够快速的缩减大数据规模,并通过多粒度的表示方式增强鲁棒性。

然而,传统粒球生成算法的随机划分问题,往往会导致粒球数量的不确定和分布的不稳定,对后续的计算结果造成了极大的干扰。同时,使用粒球内样本点的质心作为球心的方法虽然保证了粒球能够覆盖在合理的位置,但会导致球心大多不是原始数据集的样本点,减弱了粒球替代样本点的可解释性。因此在粒球邻域粗糙集的属性约简算法[12]中,只能使用生成正域作为属性约简的判断条件。生成正域是由纯度为1的粒球球心构成,这些球心往往并不是数据集中的实际样本点。

针对这一问题,本文引入了密度峰值聚类算法[20],提出了基于密度峰值的粒球生成(Density Peak Granular-ball Generation, DPG-BG)算法。使用密度峰值点和质心最近点作为球心,在增强粒球可解释性的同时,提高后续粒球邻域粗糙集的属性约简精度。本文的主要贡献包括三个方面:

1)本文提出了一种基于密度峰值的粒球生成算法,包括新的粒球重叠消除方法,用密度峰值点和质心最近点替代传统粒球的球心。新的粒球使用实际样本点作为球心,增强了粒球替代样本点的可解释性;

2)新的粒球生成算法遵循了“大范围首先”的思想,使用密度峰值聚类由粗到细的生成多粒度粒球。由于摒弃了传统方法中随机划分的步骤,粒球生成的数量和分布均达到稳定状态;

3)结合密度峰值粒球与邻域粗糙集,推导出基于密度峰值的粒球邻域粗糙集模型。在此基础上设计了使用正域的后向属性约简算法,在保证半径自适应优点的同时使用实际样本点进行属性约简。实验结果表明,本文提出的算法具有更高的准确率。

1 相关工作

1.1 粒球计算

当前粒球的主要生成方式是通过分裂迭

代。从整个数据集出发,将其视为第一个初始粒球,这遵循的是人类认知的“大范围优先”原则。为了加速粒球的生成,Xia 等提出通过迭代使用 k-means 或 k-division 进行粒球的分裂划分[21],直到粒球内的纯度达到阈值要求。在这种方法中,k 表示的是在当下粒球中不同标签的样本类数,相比于原来单一的 2-means 划分方式高效了许多。然而,这样的随机划分方式会导致粒球生成的结果不稳定,对后续的应用也会产生影响。如图 1 所示,在一个具有三类标签的数据集 fourclass3 上,纯度保持为 0.95 使用 k-means 对粒球进行六次生成。图 1(a—f)是生成的粒球结果,由图可以看出数量足够的粒球可以对数据集进行良好地覆盖,然而粒球的分布始终不稳定,数量也在 17~35 之间波动。

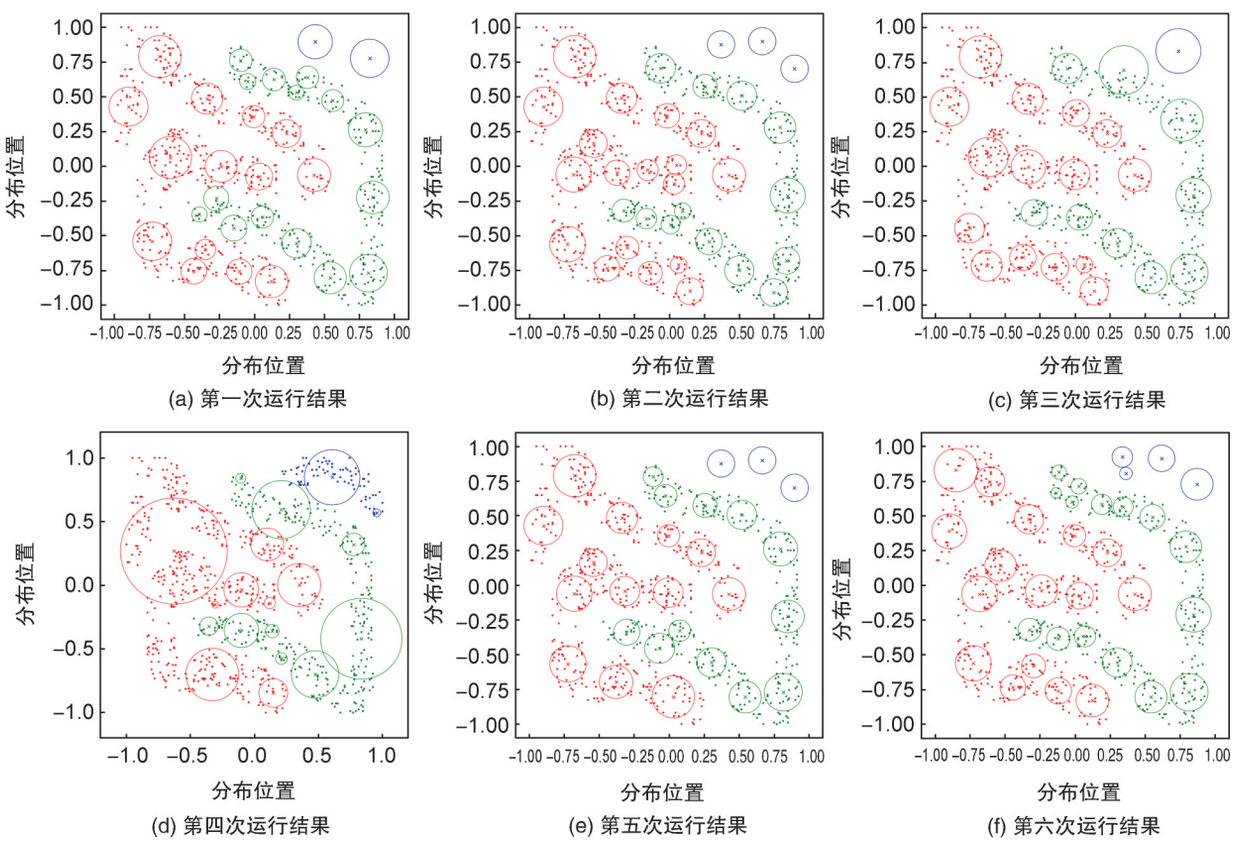


图 1 k-means粒球在同一数据集上生成的多种不稳定结果

Fig. 1 Multiple unstable results of k-means granular-balls generating on the same data set

粒球计算的重要思想是利用粒球(由球的中心点和半径组成)作为输入的基本单位,而不是单个数据点。由于其对数据的覆盖,使用粒球不仅简化了数据的规模,且适应于任何维度的空间。粒球的定义如下:

定义 1[14] 给定数据集 $S\subseteq R^N$ 和一个非空子集 $S'\subseteq S$。在 S 上生成一个粒球(Granular Ball,GB),其中心为 C,半径为 r。C 表示 GB 中所有样本点的质心,而 r 表示 GB 中所有点到 C 的平均距离。具体来说,对于 S' 中的每个点 $y_i(i=1,2,\cdots,N)$,其中 N 是 S' 中点的数量,球心和半径定义如下:

$$C=\frac{1}{N}\sum_{i=1}^{N}y_i,\qquad r=\frac{1}{N}\sum_{i=1}^{N}\|y_i-C\|。\tag{1}$$

定义 2[14] 设 G 是在数据集 $S\subseteq R^N$ 的非空子集 $S'\subseteq S$ 上生成的粒球,其球心为 C,半径为 r。G 的整体标签由球内多数样本标签决定,即在 G 中出现频次最高的标签。

1.2 粒球邻域粗糙集模型

邻域粗糙集是处理不确定数据非常有用的工具。然而,一个显著的缺点是邻域半径需要手动设置,这可能导致后续计算的准确性产生

波动。为了解决这个问题,Xia等引入了粒球计算的概念,并提出了GBNRS算法。

这个新模型可以处理连续数据,无需预先设置邻域半径,使其成为一个无参数的邻域粗糙集模型。根据数据分布的特性,它可以自适应地产生多粒度的邻域半径,从而突破了单一固定半径的限制,比传统的NRS模型更高效。粒球邻域粗糙集的数学模型来源于NRS的数学模型,如下所示:

定义3[12] 给定一个非空有限集合$S=\{y_1,y_2,\cdots,y_n\}$,第k个粒球用G_k表示,其中心为C_k,半径为r_k。对于$y_i\in G_k$,y_i的邻域被定义为$\tau(y_i)$,

$$\tau(y_i)=\{y\,|\,\forall y\in G_k,\ |y,C_k|\leqslant r_k\},\tag{2}$$

其中$|y,C_k|$表示y和C_k之间的距离。

定义4[12] 给定一个决策系统$<S,F,G>$,其中G将S分成N个不同的类别:$Y_1,Y_2,\cdots,Y_N$,使得粒球生成(Granular Ball Generation,GBG)算法构建出覆盖整个数据集的粒球。设第k个粒球为G_k,对于任意子集$E\subseteq F$,决策集D相对于条件子集H的上近似、下近似和生成下近似的定义如下:

$$\bar{H}D=\bigcup_{k=1}^{N}\bar{H}Y_k,\tag{3}$$

$$\underline{H}D=\bigcup_{k=1}^{N}\underline{H}Y_k,\tag{4}$$

$$\underline{H}D'=\bigcup_{k=1}^{N}\underline{H}Y_k',\tag{5}$$

其中,

$$\bar{H}Y_k=\{s_n\in S\,|\,s_n\in G_n(H),\tau(s_n)\cap Y_k\neq\varnothing\},$$

$$\underline{H}Y_k'=\left\{s=\frac{1}{n_l}\sum_{n=1}^{n_l}s_n\,\middle|\,s_n\in G_n(H),\ \tau(s_n)\subseteq Y_k\right\},$$

$$\underline{H}Y_k'=\left\{s=\frac{1}{n_l}\sum_{n=1}^{n_l}s_n\,\middle|\,s_n\in G_n(H),\tau(s_n)\subseteq Y_k\right\}。$$

如定义3和定义4所述,在GBNRS模型中,邻域的概念由粒球进行表示。当两个样本点的距离小于粒球的半径,即两个样本点在同一粒球内时,称这个粒球是球内所有样本点的邻域,并使用自适应生成的粒球半径替代邻域半径。而决策集D的下近似由纯度为1的粒球组成,这些粒球的中心点共同组成了D的生成下近似。因此,中心点的数量是属性约简的重要评价指标。

2 基于密度峰值的粒球邻域粗糙集

2.1 基于密度峰值的粒球生成

粒球邻域粗糙集得益于粒球计算的迭代分裂,突破了原有的缺陷:邻域半径需要人为指定。半径自适应的粒球邻域粗糙集也满足了多粒度的特性,突破了传统邻域粗糙集单一固定的半径约束。然而,k-means和k-division随机划分的特点导致了粒球生成的随机性,粒球的位置和数量都不稳定;其次,采用粒球内样本点的质心作为球心,这使得许多球心并不是数据集中原本的样本点,而是计算得到的数据。这一策略虽然保证了粒球覆盖在合理的位置,但也减弱了粒球替代样本点的可解释性。这些缺陷往往会干扰基于样本点的算法的精确度,例如邻域粗糙集的属性约简。

基于上述问题,本文提出了DPGBG算法,使用密度峰值点和质心最近点作为新的粒球球心,在稳定粒球生成结果的同时增强粒球替代样本点的可解释性。密度峰值点作为局部密度最大点可以代表这一局部区域内的其他样本点,而粒球去重叠后的质心最近点同样也可以近似地替代原来的球心。同时,这两种点都是由实际样本点构成,并非数据集的生成点。在此基础上,将DPGBG算法引入邻域粗糙集进行构造,新的球心将全部由实际样本点构成。因此邻域的中心也全部是实际样本点,不需要再使用生成正域进行属性约简,增强了属性约简的分类精度。

使用密度峰值聚类算法生成粒球,首先需要确定合适的距离阈值d_c,这是密度峰值聚类算法重要参数。为了达到自适应生成的目的,需要计算每个数据点到其他数据点的距离,形成一个距离矩阵。选择升序排列在2%位置的值作为距离阈值,这个值将用来判断“局部”的范围。然后统计每个数据点附近距离小于d_c的点的数量,这个数量就是数据点的局部密度。

接下来是密度峰值点的选取,计算每个点到密度高于它的点的最小距离。通过这一步可以确定哪些点不仅在高密度区域,而且远离其他高密度点。通过计算局部密度和最小距离的

组合,可以确定密度峰值点,这些点往往位于密度高且距离其他高密度点较远的位置。

确定了密度峰值点,其他点将被分配到最近的密度峰值点,形成不同的簇。密度峰值点将作为初始粒球的球心,而簇内样本点到球心的平均距离将作为半径。初始粒球接下来将以纯度为标准,进行迭代提纯,得到满足阈值要求的粒球列表。如图2所示,展示的是在二维和三维数据集上,纯度为0.95时,DPGBG算法的覆盖效果与表现。原始数据点的分布展现在图2(a)和图2(e),图2(b)和图2(f)表示的是未去重叠的粒球效果图,图2(c)和图2(g)表示的是最终的粒球结果,而图2(d)和图2(h)中只有粒球,能够直观的体现出粒球替代样本点的效果。

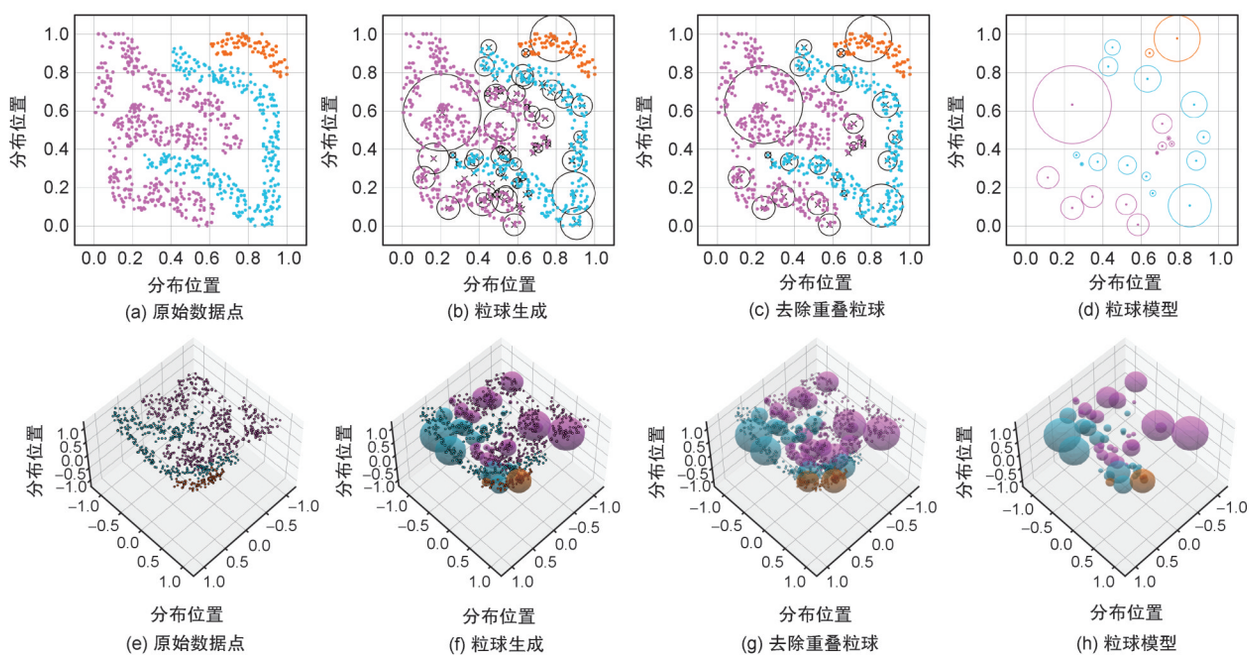


图2 密度峰值粒球生成结果

Fig. 2 The process of generating density peak granular-balls

2.2 粒球重叠的改进消除方法

使用密度峰值聚类形成簇之后,对于纯度达标的簇使用密度峰值点作为球心,计算其他点到球心的平均距离作为半径,生成密度峰值粒球。然而由于数据的分布特性,有一部分的密度峰值点出现的位置过偏或者与其他高密度点过近,并不适合作为粒球覆盖的定位点。因此本文对于重叠的粒球进行如下消除:

检测粒球是否重叠:粒球 $G_1^{\mathrm{dp}}=(c_1,r_1,l_1)$ 和 $G_2^{\mathrm{dp}}=(c_2,r_2,l_2)$,当 $|c_i,c_j|<r_i+r_j$ 时,则认为这两个粒球重叠。对于重叠的粒球,计算球内样本点的质心,将距离质心最近的样本点作为新的球心,并重新计算粒球的半径。

这样的重叠消除办法既使得覆盖效果不好的粒球得到移动,也保证了粒球球心由实际样本构成。新的密度峰值粒球生成方法继承了传统粒球生成方法的高效覆盖性,同时也增强了粒球替代样本点的可解释性。因此,完整的DPGBG算法如算法1所示。

算法1 基于密度峰值聚类的粒球生成算法。

输入:数据集 D,决策系统 $<U,C,D,g>$,纯度阈值 P;

输出:基于密度峰值的粒球列表 $G_{\mathrm{list}}^{\mathrm{dp}}$。

① 由全部属性 C 计算所有点的全局距离矩阵 $\boldsymbol{D}(\mathrm{M})$,选择从小到大位于前2%的距离值作为参数 d_c;

② 使用 d_c 参数计算点 i 的局部密度 D_l 和点 i 到密度更高点的最小距离 L_d;

③ 计算得分 $S=D_l\cdot L_d$,选择得分高的点作为密度峰值点,进行聚类;

④ 使用密度峰值点作为粒球的球心,并计算粒球半径,得到粒球列表 $G_{\mathrm{list}}^{\mathrm{dp}}=(c_i,r_i,l_i)$;

⑤ For $G_{\mathrm{list}}^{\mathrm{dp}}$ do

计算每个密度峰粒球 G_i^{dp} 的纯度 $P_i^{\mathrm{dp}G}$;

If $P_i^{\mathrm{dp}G}<P$ **then**

重复步骤②,③,④

End If

End For

⑥ 对每个密度峰粒球 G_i^{dp} 进行重叠检测:

⑦ **If** $l_i\neq l_j$,并且$\left|c_i,c_j\right|<r_i+r_j$ **then**

寻找质心最近点作为新的球心,并计算新的半径,得到新的密度峰粒球 G_i^{dp};

End If

⑧ 输出基于密度峰值的粒球列表 $G_{\mathrm{list}}^{\mathrm{dp}}$。

2.3 基于密度峰粒球的邻域粗糙集属性约简算法

本节将基于密度峰值的粒球引入邻域粗糙集中,推导出密度峰粒球邻域粗糙集(Density Peak Granular-Ball Neighborhood Rough Set, DP-GBNRS)数学模型,并以此为基础设计出后向的DPGBNRS属性约简算法。

定义5 给定一个决策系统$<S,F,G>$,生成覆盖S的密度峰粒球列表$G_{\mathrm{list}}^{\mathrm{dp}}=(c_i,r_i,l_i)$。对于任意样本点$\alpha_i\in G_{\mathrm{list}}^{\mathrm{dp}}$,$\alpha_i$的邻域定义为:

$$\sigma(\alpha_i)=\{\alpha\in S\left|\forall\alpha\in G_j^{\mathrm{dp}},d(\alpha_i,c_j)\leqslant r_j\right.\},\quad(6)$$

其中$d(\alpha_i,c_j)$表示样本点α_i到球心c_j的距离。

定义6 在决策系统中,密度峰粒球G^{dp}对全域进行划分,目标子集$A\subseteq S$的上下近似定义为:

$$\underline{G}_{\mathrm{dp}}^{N}(A)=\left\{\alpha_i\in U\left|\sigma(\alpha_i)\subseteq A\right.\right\},\quad(7)$$

$$\overline{G}_{\mathrm{dp}}^{N}(A)=\left\{\alpha_i\in U\left|\sigma(\alpha_i)\cap A\neq\phi\right.\right\}。\quad(8)$$

当$\underline{G}_{\mathrm{dp}}^{N}(A)\neq\overline{G}_{\mathrm{dp}}^{N}(A)$时,密度峰粒球邻域粗糙集成立,记为$\left(\underline{G}_{\mathrm{dp}}^{N}(A),\overline{G}_{\mathrm{dp}}^{N}(A)\right)$。

定义7 目标子集A的正负域以及边界域的定义分别如下:

$$\mathrm{POS}_C(\alpha)=\underline{G}_{\mathrm{dp}}^{N}(A),\quad(9)$$

$$\mathrm{NEG}_C(\alpha)=S-\overline{G}_{\mathrm{dp}}^{N}(A),\quad(10)$$

$$\mathrm{BND}_C(\alpha)=\overline{G}_{\mathrm{dp}}^{N}(A)-\underline{G}_{\mathrm{dp}}^{N}(A)。\quad(11)$$

如定义所述,认为每个样本点所属的粒球就是它的邻域。因此邻域半径也就是粒球的半径,样本点所属的纯度为1的粒球就是该样本点的正域。

定义8 对于决策系统$<S,F,G>$,$F=C\cup D$,条件属性集$B\subseteq D$,$a=C-B$,则条件属性a的属性重要度定义:

$$\mathrm{sig}(a,B,D)=\frac{1}{\sum_i N_i^{\mathrm{dp}G}},\quad P_i^{\mathrm{dp}G}=1,\quad(12)$$

其中$N_i^{\mathrm{dp}G}$表示在条件属性集B下生成的纯度为1的密度峰粒球N_i^{dp}样本数量。

在属性约简时,以正域内包含的样本数作为分类能力的评判标准。纯度为1的粒球包含样本点数越多,则该属性组合的分类能力越强。因此,基于密度峰粒球的邻域粗糙集属性约简算法如算法2所示。

算法2 基于密度峰粒球的邻域粗糙集属性约简算法

输入:数据集D,决策系统$<U,C,D,g>$;

输出:约简集R。

① 根据算法1,生成固定纯度为1的密度峰粒球列表$G_{\mathrm{list}}^{\mathrm{dp}}=(c_i,r_i,l_i)$;

② $R=\varnothing$;

③ **For** each $R_i=C-a_i$ **do**

计算每个属性组合R_i下的正域样本数量$N_i^{\mathrm{dp}G}$;

选择R_i,满足$N_i^{\mathrm{dp}G}=\max\limits_i(N_i^{\mathrm{dp}G})$;

If $N_i^{\mathrm{dp}G}>N_R^{\mathrm{dp}G}$ **then**

$R=R_i,C=R,N_R^{\mathrm{dp}G}=N_i^{\mathrm{dp}G}$;

Else

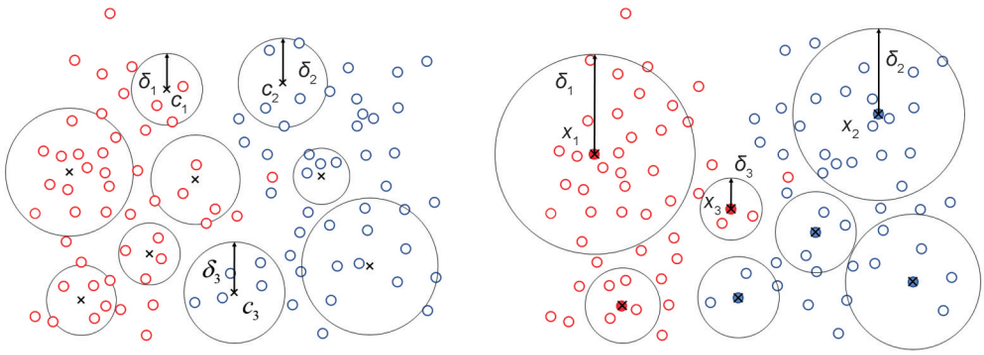
返回R;

End If

End For

④ 输出约简集R。

如图3所示,在相同的数据集上,两种粒球邻域粗糙集正域的构成并不相同。图3(a)为GBNRS模型,可以明显看到大部分粒球的球心并不在原始样本点上,而是出现在数据空间内空白的位置。在GBNRS模型中,这样的球心可以保证粒球对样本点高效率的覆盖,但直接使用这样的球心作为“生成正域”进行属性约简缺乏可解释性。图3(b)为本文提出的DPGBNRS模型,在纯度同为1的条件下,生成的粒球数量更少,并且球心由实际样本点构成。因此DPGBNRS模型突破了使用生成正域进行属性约简的局限,用实际样本点进行属性约简。



(a) GBNRS模型生成的邻域 (b) DPGBNRS模型生成的邻域

图 3 GBNRS模型和DPGBNRS模型对比

Fig. 3 Comparison between GBNRS model and DPGBNRS model

3 实验分析

为了验证本文所提出的DPGBG算法和DP-GBNRS属性约简算法的性能和效率,在本节中将DPGBG算法和传统的基于k-means的GBG算法进行时间和稳定性的比较。本文还使用K最邻近(K-Nearest Neighbor, KNN)分类算法和支持向量机(Support Vector Machine, SVM)算法对NRS、超球邻域粗糙集(Hypersphere Neighborhood Rough Set, HSNRS)[22]、GBNRS和DP-GBNRS的约简结果进行准确度比较。

本节选取了18种UCI公开数据集,包括连续数据和离散数据,数据集的具体描述见表1。本研究的计算实验在一台个人计算机上进行,该计算机配备了32 GB的动态随机存取存储器(Dynamic Random Access Memory,DRAM)和Intel酷睿i7-10700 CPU @ 2.90 GHz。在Python 3.10.9中使用PyCharm 2023.3.1开发环境进行编程。

3.1 粒球生成的时间消耗和结果稳定性对比

在本节实验中,将针对Xia等提出的粒球生成算法及本文提出的基于密度峰值的粒球生成算法,进行时间消耗和结果稳定性的对比。由于传统的GBG算法采取了k-means的随机中心划分策略,因此对GBG算法运行10次求解时间消耗的平均值,具体结果如表2所示。

观察表2,不难得出如下结论:在大部分数据集上,DPGBG算法的时间消耗要低于传统的GBG算法。这说明,由于迭代次数的减少,DP-GBG算法可以有效地提升粒球生成的效率。

同时,为了展示DPGBG算法生成结果的稳定性,本文对两种算法生成的球数进行了统计

表 1 实验数据集

Table 1 Datasets used in the experiments

编号	数据集	样本数	特征数	决策类
1	dataR2	116	9	2
2	lymphography	148	18	4
3	hepatitis	155	19	2
4	wine	178	13	3
5	sonar	208	60	2
6	Glass	214	9	6
7	Algerian_forest_fires_dataset_UP-DATE	244	10	2
8	heart	270	13	2
9	cleveland	297	13	5
10	heart2	303	13	5
11	iono	351	34	2
12	HVNT	606	100	2
13	credit	690	15	2
14	Diabetes	768	8	2
15	Pima	768	8	17
16	fourclas3	862	2	3
17	tic-tac-toe	958	9	2
18	Parkinson_Multiple_Sound_Recording	1 040	27	2

和对比。对GBG算法统计了10次的平均生成球数与最大最小球数,结果如表3和图4所示。

观察表3和图4,不难得出结论:GBG算法在多次迭代中生成的球数表现出明显的不稳定性,每次生成结果的最大值和最小值之间存在较大差距。相比之下,DPGBG算法每次生成的球数始终相同,表明其结果稳定且可靠,在需要稳定结果的应用领域中是更优的选择。

3.2 属性约简的分类精度对比

在本节实验中,首先对DPGBNRS的属性约简结果与NRS、HSNRS和GBNRS的属性约

表2 粒球生成算法时间消耗对比

Table 2 Comparison of time consumption for GBG algorithm

单位:s

数据集	GBG	DPGBG
dataR2	0.334 0	**0.113 7**
lymphography	0.363 8	**0.132 6**
hepatitis	0.398 3	**0.128 8**
wine	0.491 4	**0.086 8**
sonar	0.407 3	**0.165 2**
Glass	**0.217 7**	0.237 0
Algerian_forest_fires_dataset_UPDATE	0.074 0	**0.070 9**
heart	0.712 3	**0.230 1**
cleveland	0.738 8	**0.348 7**
heart2	0.948 4	**0.439 9**
iono	0.435 9	**0.325 5**
HVNT	1.669 0	**1.503**
credit	1.244 9	**0.682 2**
Diabetes	1.228 3	**1.008 1**
Pima	**2.632 3**	2.875 3
fourclas3	0.264 2	**0.258 1**
tic-tac-toe	**1.507 8**	1.911 2
Parkinson_Multiple_Sound_Recording	1.836 1	**1.416 4**
均值	0.861 3	**0.662 9**

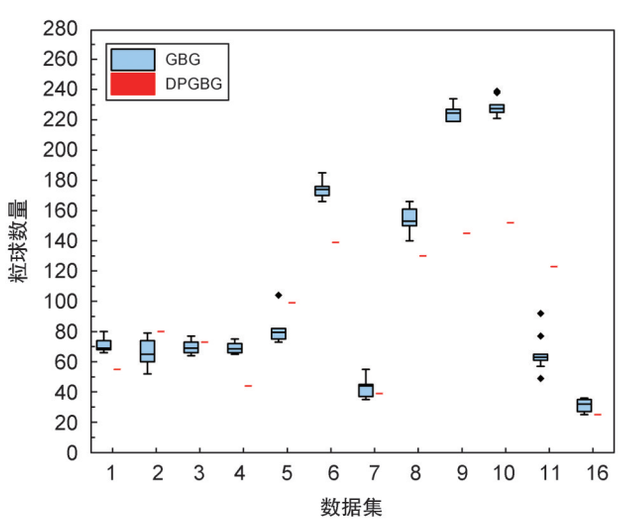
表3 粒球生成算法球数对比

Table 3 Comparison of numbers for GBG algorithm

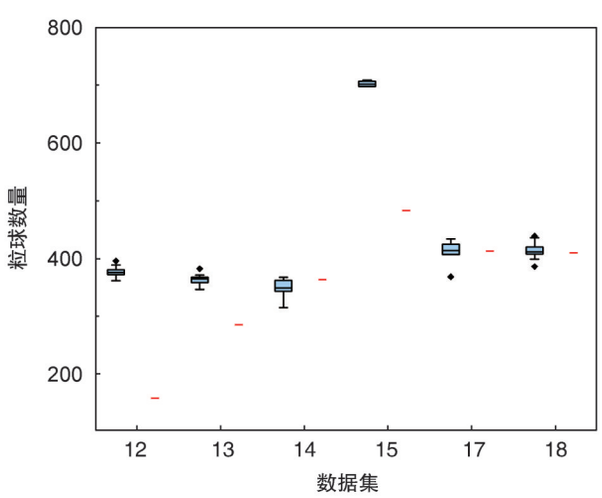
数据集	GBG			DPGBG
	平均球数	最大球数	最小球数	球数
1	70.7	80	66	55
2	66.8	79	52	80
3	69.6	77	64	73
4	68.8	75	66	44
5	80.8	104	74	99
6	173.7	185	166	139
7	43	55	35	39
8	154	166	140	130
9	224.2	234	219	145
10	228.5	239	221	152
11	65.2	92	49	123
12	376.8	396	361	158
13	362.8	346	382	285
14	347.4	315	367	363
15	702.3	709	698	483
16	30.9	36	25	25
17	412.1	434	368	419
18	413.8	439	386	410
均值	216.2	225.6	207.8	179

简结果进行对比。其中,对于NRS的半径选择问题,本文选取了0.02,0.04,…,0.40等20个不同半径,步长为0.02。

表4展示了在求解约简时得到的约简率,其中HNRS在第17个数据集(tic-tac-toe)上无法得到有效的约简结果,原因是HNRS不适合此类数据,表中用“—”代替。通过对比不同算法的约简率,可以看出所提出的DPGBNRS算法在约简效果上具有一定的优势。总体来说,DPGBNRS算法的平均约简率为0.206 2,虽然略低于传统NRS算法的0.669 8和HSNRS算法的0.482 9,但明显高于GBNRS算法的0.158 1。



(a) GBG和DPGBG算法分别在数据集1—11、16上的稳定性对比



(b) GBG和DPGBG算法分别在数据集12—15、17、18上的稳定性对比

图4 粒球生成算法的稳定性对比

Fig. 4 Comparison of stability for GBG algorithm

具体来看,DPGBNRS算法在第1、3、5、6、7和10个数据集上的表现优于GBNRS算法和HSNRS算法,显示出DPGBNRS算法在某些情况下能够更有效地进行属性约简。尽管DPGBNRS算法在部分数据集上未能达到NRS算法的约简率,但其在多个数据集上表现出色,表明其在特定条件下具有潜在优势。

在本节中,还分别采取KNN(K取值为5)和SVM(libSVM为默认函数)两种分类器,利用约简所求的属性,随机选择30%的测试集进行分类,具体的分类准确率如表5所示。

从表5展示的结果来看,可以得出以下结论:在多个数据集上,DPGBNRS算法展现出了较高的分类准确率,特别是在wine和Algerian_forest_fires_dataset_UPDATE数据集上,DPGBNRS算法表现突出,分别达到了0.981 5和0.972 6的高准确率。尽管在部分数据集上,该算法表现不如其他算法,但也能保持在一个平均的水平。综合来看,该方法在大多数情况下都取得了较好的分类效果。因此,本文认为DPGBNRS算法在属性约简中具有良好的性能和更高的稳定性。

表4 求解约简时得到的约简率

Table 4 Reduct ratios related to deriving reducts

数据集	NRS	HSNRS	GBNRS	**DPGBNRS**
1	**0.777 8**	0.333 3	0.111 1	0.111 1
2	**0.666 7**	0.444 4	0.166 7	0.222 2
3	**0.842 1**	0.263 2	0.157 9	0.052 6
4	0.769 2	0.153 8	0.153 8	**0.384 6**
5	**0.816 7**	0.066 7	0.050 0	0.100 0
6	0.333 3	**0.555 6**	0.222 2	0.111 1
7	**0.800 0**	0.500 0	0	0.100 0
8	0.769 2	0.230 8	0.230 8	**0.307 7**
9	**0.769 2**	0.692 3	0.076 9	0.307 7
10	**0.769 2**	0.307 7	0.076 9	0.153 8
11	0.852 9	0.441 2	**0.558 8**	0.147 1
12	0.980 0	**1.000 0**	0.190 0	0.020 0
13	0.666 7	**0.933 3**	0.133 3	0.266 7
14	0.250 0	0.375 0	0.250 0	**0.375 0**
15	0.250 0	**0.875 0**	0.125 0	0.625 0
17	0.111 1	—	0	**0.111 1**
18	**0.963**	0.555 6	0.185 2	0.111 1
均值	**0.669 8**	0.482 9	0.158 1	0.206 2

注:—表示HNRS在第17个数据集(tic-tac-toe)上无法得到有效的约简结果。

表5 KNN和SVM分类器的分类准确率

Table 5 Classification accuracy of KNN and SVM classifiers

数据集	KNN分类准确率				SVM分类准确率			
	NRS	HSNRS	GBNRS	**DPGBNRS**	NRS	HSNRS	GBNRS	**DPGBNRS**
1	**0.714 3**	0.678 3	0.571 4	0.685 7	0.742 8	0.742 9	0.600 0	**0.771 4**
2	0.777 8	0.775 9	**0.800 0**	0.755 6	0.666 6	0.711 1	**0.755 5**	0.711 1
3	0.723 4	0.838 1	0.744 7	**0.808 5**	0.702 1	0.787 2	0.808 5	**0.808 5**
4	0.925 9	0.768 4	0.962 9	**0.981 5**	0.925 9	0.851 9	0.981 4	**0.981 5**
5	0.698 4	0.651 9	**0.841 2**	0.761 9	**0.981 1**	0.634 9	0.793 6	0.730 2
6	**0.384 6**	0.380 3	0.281 3	0.359 4	0.353 84	0.359 4	0.312 6	**0.359 4**
7	0.837 8	0.942 5	0.917 8	**0.972 6**	0.824 3	0.931 5	0.917 8	**0.958 9**
8	0.567 9	0.717 3	0.740 7	**0.777 8**	0.604 9	0.666 7	**0.740 7**	0.728 4
9	0.511 1	0.564 2	**0.606 7**	0.595 5	**0.622 2**	0.550 6	0.550 6	0.584 3
10	0.472 5	0.563 1	0.472 5	**0.549 5**	**0.560 4**	0.560 4	0.560 4	0.549 5
11	**0.971 7**	0.862 9	0.857 1	0.904 8	**0.971 6**	0.914 3	0.895 2	0.876 2
12	0.533 0	0.542 1	0.445 1	**0.560 4**	**0.478 0**	**0.478 0**	0.428 7	**0.478 0**
13	0.681 2	**0.860 7**	0.826 1	0.835 7	0.613 5	0.821 3	0.816 4	**0.862 1**
14	0.722 9	0.718 4	0.735 9	**0.770 6**	0.753 2	0.770 6	0.757 5	**0.783 5**
15	0.138 5	0.152 6	**0.207 8**	0.203 5	0.134 2	0.225 1	**0.233 7**	0.194 8
17	0.393 3	—	**0.878 5**	0.670 1	0.715 2	—	**0.836 8**	0.670 1
18	0.493 6	0.849 9	0.891 0	**0.980 8**	0.532 1	0.910 3	0.891	**1.000 0**
均值	0.620 5	0.679 2	0.693 0	**0.716 1**	0.668 9	0.682 2	0.698 8	**0.708 7**

注:—表示HNRS在第17个数据集(tic-tac-toe)上无法得到有效的约简结果。

4 总结与展望

在使用粒球邻域粗糙集进行属性约简时,对粒球生成有很大的依赖性,时间和精度都受到粒球的影响。因此,如果粒球生成的结果不稳定,后续的属性约简计算都会产生波动。

因此,本文将密度峰值聚类算法引入粒球的生成过程中,使用密度峰值点和质心最近点作为粒球的球心,目的是能够更准确的用粒球表示样本点。同时,得益于密度峰值聚类的特性,粒球生成的结果更加稳定。实验结果也表明,新的粒球生成算法和属性约简算法更加高效且准确。

在本文的基础上,未来可对密度峰值的参数进行深入探索,如何更加快速高效地找到适合数据集的密度参数,将会更快的生成粒球。同时,其他的聚类策略也可以用来提升粒球的稳定性与准确性。

参考文献:

[1] PAWLAK Z. Rough Sets[J]. *Int J Comput Inf Sci*, 1982, **11**(5): 341−356. DOI:10.1007/bf01001956.

[2] 王国胤, 姚一豫, 于洪. 粗糙集理论与应用研究综述[J]. 计算机学报, 2009, **32**(7): 1229−1246. DOI: 10.3724/SP.J.1016.2009.01229.
WANG G Y, YAO Y Y, YU H. A Survey on Rough Set Theory and Applications[J]. *Chin J Comput*, 2009, **32**(7): 1229−1246. DOI: 10.3724/SP.J.1016.2009.01229.

[3] 胡清华, 于达仁, 谢宗霞. 基于邻域粒化和粗糙逼近的数值属性约简[J]. 软件学报, 2008, **19**(3): 640−649. DOI: 10.3724/SP.J.1001.2008.00640.
HU Q H, YU D R, XIE Z X. Numerical Attribute Reduction Based on Neighborhood Granulation and Rough Approximation[J]. *J Softw*, 2008, **19**(3): 640−649. DOI: 10.3724/SP.J.1001.2008.00640.

[4] HU Q H, YU D R, LIU J F, *et al*. Neighborhood Rough Set Based Heterogeneous Feature Subset Selection[J]. *Inf Sci*, 2008, **178**(18): 3577−3594. DOI: 10.1016/j.ins.2008.05.024.

[5] WANG C Z, SHAO M W, HE Q, *et al*. Feature Subset Selection Based on Fuzzy Neighborhood Rough Sets[J]. *Knowl Based Syst*, 2016, **111**: 173−179. DOI: 10.1016/j.knosys.2016.08.009.

[6] XU W H, YUAN Z T, LIU Z. Feature Selection for Unbalanced Distribution Hybrid Data Based on K-nearest Neighborhood Rough Set[J]. *IEEE Trans Artif Intell*, 2024, **5**(1): 229−243. DOI:10.1109/TAI.2023.3237203.

[7] 胡清华, 赵辉, 于达仁. 基于邻域粗糙集的符号与数值属性快速约简算法[J]. 模式识别与人工智能, 2008, **21**(6): 732−738. DOI: 10.3969/j.issn.1003-6059.2008.06.004.
HU Q H, ZHAO H, YU D R. Efficient Symbolic and Numerical Attribute Reduction with Neighborhood Rough Sets[J]. *Pattern Recognit Artif Intell*, 2008, **21**(6): 732−738. DOI: 10.3969/j.issn.1003-6059.2008.06.004.

[8] HU Q H, PEDRYCZ W, YU D R, *et al*. Selecting Discrete and Continuous Features Based on Neighborhood Decision Error Minimization[J]. *IEEE Trans Syst Man Cybern B Cybern*, 2010, **40**(1): 137−150. DOI: 10.1109/TSMCB.2009.2024166.

[9] CHEN H M, LI T R, FAN X, *et al*. Feature Selection for Imbalanced Data Based on Neighborhood Rough Sets[J]. *Inf Sci*, 2019, **483**: 1−20. DOI:10.1016/j.ins.2019.01.041.

[10] 段洁, 胡清华, 张灵均, 等. 基于邻域粗糙集的多标记分类特征选择算法[J]. 计算机研究与发展, 2015, **52**(1): 56−65. DOI: 10.7544/issn.1000-1239.2015.20140544.
DUAN J, HU Q H, ZHANG L J, *et al*. Feature Selection for Multi-label Classification Based on Neighborhood Rough Sets[J]. *J Comput Res Dev*, 2015, **52**(1): 56−65. DOI: 10.7544/issn.1000-1239.2015.20140544.

[11] HU M, TSANG E C C, GUO Y T, *et al*. A Novel Approach to Attribute Reduction Based on Weighted Neighborhood Rough Sets[J]. *Knowl Based Syst*, 2021, **220**: 106908. DOI:10.1016/j.knosys.2021.106908.

[12] XIA S Y, ZHANG H, LI W H, *et al*. GBNRS: a Novel Rough Set Algorithm for Fast Adaptive Attribute Reduction in Classification[J]. *IEEE Trans Knowl Data Eng*, 2022, **34**(3): 1231−1242. DOI: 10.1109/TKDE.2020.2997039.

[13] XIA S Y, WANG C, WANG G Y, *et al*. GBRS: a Unified Granular-ball Learning Model of Pawlak Rough Set and Neighborhood Rough Set[J]. *IEEE Trans Neural Netw Learn Syst*, 2025, **36**(1): 1719−1733. DOI: 10.1109/TNNLS.2023.3325199.

[14] XIA S Y, LIU Y S, DING X, *et al*. Granular Ball Computing Classifiers for Efficient, Scalable and Robust Learning[J]. *Inf Sci*, 2019, **483**: 136−152. DOI:10.1016/j.ins.2019.01.010.

[15] XIE J, KONG W Y, XIA S Y, *et al*. An Efficient Spectral Clustering Algorithm Based on Granular-ball[J]. *IEEE Trans Knowl Data Eng*, 2023, **35**(9): 9743−9753. DOI:10.1109/TKDE.2023.3249475.

[16] XIA S Y, PENG D W, MENG D Y, *et al*. Ball k-Means: Fast Adaptive Clustering With No Bounds[J]. *IEEE Trans Pattern Anal Mach Intell*, 2022, **44**(1): 87-99. DOI:

10.1109/tpami.2020.3008694.

[17] CHENG D D, LI Y, XIA S Y, *et al*. A Fast Granular-ball-based Density Peaks Clustering Algorithm for Large-scale Data[J]. *IEEE Trans Neural Netw Learn Syst*, 2024, **35**(12): 17202−17215. DOI: 10.1109/TNNLS.2023.3300916.

[18] XIA S Y, ZHENG S Y, WANG G Y, *et al*. Granular Ball Sampling for Noisy Label Classification or Imbalanced Classification[J]. *IEEE Trans Neural Netw Learn Syst*, 2023, **34**(4): 2144−2155. DOI: 10.1109/TNNLS.2021.3105984.

[19] QIAN W B, XU F K, HUANG J T, *et al*. A Novel Granular Ball Computing-based Fuzzy Rough Set for Feature Selection in Label Distribution Learning[J]. *Knowl Based Syst*, 2023, **278**: 110898. DOI: 10.1016/j.knosys.2023.110898.

[20] RODRIGUEZ A, LAIO A. Clustering by Fast Search and Find of Density Peaks[J]. *Science*, 2014, **344**(6191): 1492−1496. DOI:10.1126/science.1242072.

[21] XIA S Y, DAI X C, WANG G Y, *et al*. An Efficient and Adaptive Granular-ball Generation Method in Classification Problem[J]. *IEEE Trans Neural Netw Learn Syst*, 2022, **35**(4): 5319−5331. DOI: 10.1109/TNNLS.2022.3203381.

[22] FANG Y, CAO X M, WANG X, *et al*. Hypersphere Neighborhood Rough Set for Rapid Attribute Reduction [C]//Pacific−Asia Conference on Knowledge Discovery and Data Mining. Cham: Springer International Publishing, 2022: 161−173. DOI: 10.1007/978-3-031-05936-0_13.