计算机科学
COMPUTER SCIENCE

GBDEN:一种基于粒球的大规模数据快速聚类方法

薛任煊, 伊士超, 王平心

引用本文

薛任煊, 伊士超, 王平心. GBDEN:一种基于粒球的大规模数据快速聚类方法[J]. 计算机科学, 2024, 51(12): 166-173.

XUE Renxuan, YI Shichao, WANG Pingxin. GBDEN:A Fast Clustering Algorithm for Large-scale Data Based on Granular Ball [J]. Computer Science, 2024, 51(12): 166-173.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240600002

GBDEN:一种基于粒球的大规模数据快速聚类方法

薛任煊[1]　伊士超[2]　王平心[2]
1 江苏科技大学计算机学院　江苏 镇江 212100
2 江苏科技大学理学院　江苏 镇江 212100
(231210702103@stu.just.edu.cn)

摘　要　聚类用于将数据集中的对象划分为具有相似特征的组或类别,使得同一组内的对象之间的相似度较高,而不同组之间的相似度较低。密度聚类是无监督聚类方法之一,它不需要提前指定类簇的数量,而是根据数据的密度来自动确定。与 K 均值等方法相比,密度聚类对初始点的选择不敏感,因此更容易得到稳健的聚类结果。在众多的密度聚类算法中,DENCLUE(DENsity-based CLUstEring)算法采取了爬山策略,它具有坚实的数学基础,在大量噪声的数据集中具有良好的聚类性能,且在高维数据集中允许对任意形状进行聚类。但其在处理大规模数据集时,需要耗费大量的计算资源和时间。为此,使用粒计算的粒化模型来构建数据集。首先构建一个粗粒度的粒球,然后将粗粒度的粒球划分为细粒球,最后以粒球的形式作为 DENCLUE 算法的输入,从而进行聚类。实验结果表明,该算法在多个数据集上具有有效性。

关键词:聚类;粒计算;粒球;DENCLUE;核函数
中图分类号　TP391

GBDEN:A Fast Clustering Algorithm for Large-scale Data Based on Granular Ball

XUE Renxuan[1], YI Shichao[2] and WANG Pingxin[2]
1 School of Computer, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212100, China
2 School of Science, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212100, China

Abstract　Clustering is a technique used to partition the objects in a dataset into groups or clusters based on their similar features, aiming to form groups where objects within each group are more similar to each other than to those in other groups. Density-based clustering is one of the unsupervised clustering methods that does not require the number of clusters to be specified in advance. On the contrary, it adaptively determines the clusters based on the density of the data. Compared to methods like K-MEANS, density-based clustering is less sensitive to the selection of initial points. It also can produce more robust and reliable clustering results. Among various density-based clustering algorithms, DENCLUE(DENsity-based CLUstEring) utilizes a hill-climbing approach, which is grounded in a solid mathematical foundation. At the same time, it performs well in datasets with considerable noise, allowing clustering of arbitrarily shaped clusters in high-dimensional datasets. However, processing large-scale datasets with DENCLUE requires significant computational resources and time. To address this challenge, this paper proposes a fast clustering algorithm for large-scale data based on granular ball. This involves creating a coarse-grained granular ball initially, which is then refined into fine-grained granular balls. These granular balls served as input for the DENCLUE algorithm for clustering. Experimental findings demonstrate the effectiveness of this approach across multiple datasets.

Keywords　Clustering, Granular computing, Granular ball, DENCLUE, Kernel function

1　引言

聚类作为一种无监督学习方法,能够挖掘那些没有原始类别标签的数据集的内部信息。聚类的目的是将相似度高的样本划分为同一类,将相似度低的样本划分到不同的类,从而使得同一类中的样本保持较高的相似性,而不同类间的样本则保持较高的相异性。作为机器学习的常用方法之一,聚类分析可以有效解决现实场景中众多的建模问题,如医学图像分割[1]、生物学研究以及异常检测等。目前存在的聚类算法大致归类为:划分聚类[2]、层次

到稿日期:2024-06-03　返修日期:2024-08-30
基金项目:国家自然科学基金(62076111)
This work was supported by the National Natural Science Foundation of China(62076111).
通信作者:伊士超(shichaoyi@just.edu.cn)

聚类[3]、密度聚类[4]等。

在众多的聚类方法中,密度聚类具有数据适应性强、无需指定类簇数量等优点,受到学者们的青睐。其中,DENCLUE算法作为一种密度聚类算法,采用了核密度估计,解决了传统聚类方法在处理非线性结构、噪声、密度变化以及任意形状聚类方面的局限性。此外,该算法开创性地考虑了样本空间的密度梯度,可以在非线性分布的数据中精确地寻找到类簇中心(密度吸引子),并对一定距离范围内的样本(密度吸引点)进行合理分配,使其在众多密度聚类算法[5]中脱颖而出。然而,随着数据规模和复杂度的不断增加,算法需要在全局数据中计算每个数据点的局部密度,并且需要在迭代过程中更新每个点的密度以及密度梯度,这导致算法的计算复杂度较高。

粒计算由 Zadeh[6]提出,是一种通过将信息粒化,从不同粒度解决建模问题的方法。粒化[7]可分为构建和分解这两个相反的过程。构建指将底层的粒合并为更高层的粒,而分解则是将更粗的粒分割为更细的粒。粒球则是粒化处理后的数据单元,其原理为将数据进行层次化处理,划分为不同层次的粒度。这种层次化的处理可以是从粗粒度到细粒度的划分,其思想是将相似性的数据聚合成一个数据单元,形成一个粒球,从而使得数据处理规模大幅度减小。这一优点刚好能弥补 DENCLUE 算法计算复杂度高的不足。基于此,本文提出了基于粒球的 DENCLUE 聚类算法(Granular ball based DENCLUE,GBDEN)。首先,以一种粗粒度形式(母球)来表征数据;然后,将母球中分布相似的数据再次分割生成更为细化的粒球,在保留数据结构的前提下,大大缩小了数据规模,从而使得后续处理更加高效;最后,将粒球作为 DENCLUE 算法的输入,通过仅估计粒球的密度以及密度梯度来划分类簇,使得算法计算复杂度大大降低。同时,粒球的结果也缓解了 DENCLUE 算法对数据中的噪声点和密度差异较为敏感的问题。

这种多粒度的结构能够自适应地处理未标记数据,从而使得数据的处理更加高效。以粒球的形式作为 DENCLUE 算法的输入,从而使得计算复杂度大大降低,运行效率显著提升。

本文的研究动机在于利用粒计算的原理来有效优化 DENCLUE 算法,以提高其在处理大规模数据集时的效率和可扩展性。通过将粒化处理引入 DENCLUE 算法,可以降低其在全局数据集上计算每个数据点的密度和密度梯度时的计算复杂度,从而使其更适用于实际应用场景中处理海量数据的需求。这一优化将有助于提高 DENCLUE 算法在现实世界中的实用性,进而更好地满足复杂数据分析和模式识别等领域的需求。本文的主要贡献体现在以下 3 个方面:

1)提出了自适应的粒球生成策略,有效剔除噪声点的影响,可以在不同密度和分布特征的数据中实现更加均衡和有效的粒化表示。

2)通过粒球覆盖数据集中的所有样本,并估计粒球的密度和密度梯度,建立了一种更加紧凑和高效的表示,显著提高了 DENCLUE 算法的计算效率。

3)提出的 GBDEN 算法适用于不同类型的数据,具有较高的鲁棒性。

2 相关工作

2.1 DENCLUE 算法

DENCLUE 算法是由 Hinneburg 等[8]提出的,它利用核密度估计(Kernel Density Estimation,KDE)将整体点密度解析为数据点的核(或影响)函数的和,然后通过确定密度吸引子来识别簇,任意形状的簇可以很容易地用一个简单的总密度函数方程来描述;Hinneburg 等[9]又提出了 DENCLUE 2.0,该算法引入了新的高斯核爬山过程,即自动调整步长;He 等[10]提出了 DDT,该算法的输入不依赖于任何参数;Idrissi 等[11]提出 DENCLUE-SA 以及 DENCLUE-GA 来提升算法性能;Khader1 等[12]提出了 VDENCLUE,该算法设计每个点的核带宽根据其局部密度条件而变化。尽管上述方法通过改进对密度梯度的处理,使得聚类精确度得到有效提升,但这也增加了算法的复杂度,使得聚类过程变得困难。特别是在数据量较大时,针对全局样本的密度梯度进行复杂处理,将大大折损计算效率。

DENCLUE 主要通过两个阶段进行操作,即预聚类步骤和聚类步骤。第一步是构造数据库的映射(一个超矩形),用于加快密度函数的计算速度。至于第二步,它允许从高度密集的超立方体(网格)及其邻近的密集网格中识别聚类。DENCLUE 聚类算法是一种基于一组密度分布函数的聚类算法,其核心思想是对每一个空间数据点通过影响函数事先对空间产生影响,影响值可以叠加,即表示为密度函数。两点之间的影响函数有很多,在该算法中常用高斯核函数,如式(1)所示:

$$K(x)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x^2\right) \tag{1}$$

式(2)表示由具有恒定平滑度 h 的高斯核函数得到的密度函数。

$$f^D(x)=\frac{1}{Nh^d}\sum_{i=1}^{N}K\left(\frac{1}{h}(x-x_i)\right) \tag{2}$$

其中,D 表示为数据集,N 是样本数量,x 为观察点,x_i 为样本点。

由于只有计算数据集中接近 x 的点才对密度有贡献,因此其他的点都可以忽略而不会造成重大的错误。局部密度函数的思想便是考虑近点的影响,而忽略远点的影响,如式(3)所示:

$$\hat{f}^D(x)=\frac{1}{Nh^d}\sum_{x'\in near(x)}K\left(\frac{1}{h}(x-x')\right) \tag{3}$$

为了寻找局部极大值点作为密度吸引子,即类中心点,运用了基于梯度上升的爬山算法,如式(4)所示:

$$x=x^0,x^{i+1}=x^i+\delta\frac{\nabla\hat{f}(x^i)}{\|\nabla\hat{f}(x^i)\|} \tag{4}$$

其中，δ 是爬山过程中的一个参数，它控制着收敛速度，可以像其他爬山过程一样被动态设置。

当 $\hat{f}^D(x^{k+1})<\hat{f}^D(x^k)$ 时，则迭代就会停止($k\in N$)，并取 $x^*=x^k$ 作为密度吸引子。如果在爬山迭代过程中，$\exists x^k$，x^{k+1} 使得 $d(x^k,x^{k+1})<2\delta$，则认为 x^k，x^{k+1} 属于局部最大值的同一类簇。

由于经过爬山法确定后的类簇数量众多，区分聚类效果好坏的关键一步就是合并类簇，故定义了阈值 ξ，如图 1 所示。当 $\hat{f}^D(x^*)\geqslant\xi$ 时，x^* 为密度吸引子；当 $\hat{f}^D(x^*)<\xi$ 时，x^* 及被 x^* 密度吸引的区域为噪声点或离群点。当两个相邻的类簇之间存在距离足够小的点 o_1 和 o_2，即 $dis(o_1,o_2)\leqslant h/2$，并且 $\hat{f}^D(o_1)\geqslant\xi$ 和 $\hat{f}^D(o_2)\geqslant\xi$ 时，就合并这两个类簇。例如，在图 1 中可以合并 x_3^* 和 x_4^*。

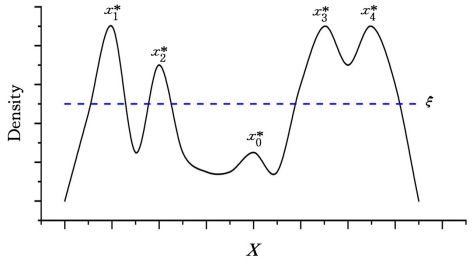


图 1 密度函数

Fig. 1 Density function

2.2 无监督粒球生成

Xia 等[13]为提升基于粒计算的分类器的性能，提出了粒球[13-15]的概念。其生成过程是迭代地使用 2-means 聚类，依据数据自身的分布自动地生成大小不一的粒球，直至粒球的纯度达到给定的阈值。粒球的纯度实际上就是粒球中那些样本标签与粒球标签一致的样本的比重。

基于粒球的概念，Xia 等[14]还提出了粒球粗糙集模型，并基于该模型探索了属性约简的相关问题。相较于 Hu 等[16]提出的邻域粗糙集下的约简求解算法，面向粒球粗糙集的约简求解算法显著地提升了时间效率。

在有监督的数据粒球生成算法中，为了判断粒球是否要进行分割，Xia 等提出了“纯度”的概念，并将其定义为粒球中多数类别的百分比，当粒球的纯度值过低时，便需要进行下一步的分割。而在无监督的数据中，因为没有标签信息，“纯度”的概念将不再适用，进而只能通过数据之间的分布关系，定义一个分割标准才能判断粒球是否进行分割。Cheng 等[17]提出了基于粒球的密度峰值算法，称为 GB-DP；Xie 等[18]提出了基于粒球的谱聚类算法，并改进了粒球分割的标准，即加权子球质量优于母球质量。但是上述粒球分割方法中，分割标准较为严苛，使得噪声点对其影响较大，生成的粒球分布较为稀疏，提升了后续聚类过程的难度。为了克服上述问题，本文提出了新的分割标准，根据子球质量的平方对母球进行分割，可以使得在有噪声的数据集中获得更合理的粒球集合。

3 基于粒球的 DENCLUE 算法

3.1 改进的无监督粒球生成算法

粒球中的本定义可以描述如下：

空间 $\Omega\subseteq R^d$ 中给定数据集 $D=\{p_1,p_2,\cdots,p_n\}$，定义了粒球(GB_j)和分布质量(DM_j)。对于每个 GB_j，其中心和半径分别为 c_j 和 r_j，如式(5)和式(6)所示：

$$c_j=\frac{1}{n_j}\sum_{i=1}^{n_j}p_i \tag{5}$$

$$r_j=\max(\|p_i-c_j\|) \tag{6}$$

其中，n_j 表示 GB_j 中的数据个数，$\|\cdot\|$ 表示二范数。

DM_j 是通过计算 GB_j 中数据个数 n_j 与 GB_j 中半径和 s_j 的比值来测量，如式(7)所示：

$$DM_j=\frac{n_j}{s_j} \tag{7}$$

其中，$s_j=\sum_{i=1}^{n_j}\|p_i-c_j\|$。

如图 2(a)所示：首先，将整个数据集看作是一个 GB_A；然后，计算出最远的两个点 c_1 和 c_2，将 GB_A 分成 GB_{A_1} 和 GB_{A_2}，如图 2(b)所示；最后，通过式(7)计算并比较 DM_A，DM_{A_1} 和 DM_{A_2} 来判断 GB_A 是否分割。但是该方法会受到离群点及噪声点的影响，导致粒球分割不完全。本文改进了粒球分割的方法，式(8)定义了 DM_{child}。

$$DM_{\text{child}}=DM_{A_1}^2+DM_{A_2}^2 \tag{8}$$

考虑到子球中分布不均匀的现象会导致 DM_j 值变小从而导致停止分割，我们的目标是将粒球分割完全，即将 DM_j 值变大。由于 $DM_j=1/\bar{r}_c$，$\bar{r}_c$ 表示 GB_j 中距离球中心 c_j 的平均半径(r_c 经过归一化后值为 0～1)，因此 $\forall j$，$DM_j>1$，又有平方函数 $f(x)=x^2$，其增长速率是关于原始数的增长速率的平方，这意味着 DM_{child} 值会变大，从而继续进行粒球的分割，如图 2(c)所示，能够更好地适应异常情况。

但如图 2(c)所示，一些半径过大的粒球可能仍会受到一些边界点或噪声点的影响，需要进行分割。

如果 $r_j>2\times\max(mean(r),median(r))$，则 GB_j 需要拆分。其中，$mean(r)$ 表示的是均值，而 $median(r)$ 表示的是中位数。

粒球生成算法的步骤如算法 1 所示。

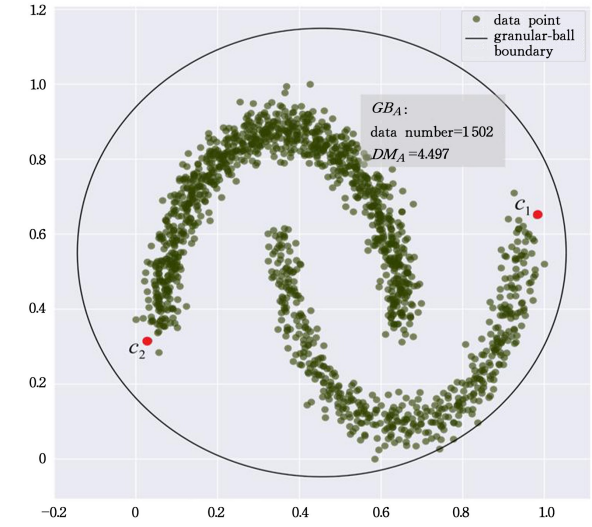
算法 1 粒球生成算法

输入：GB_A

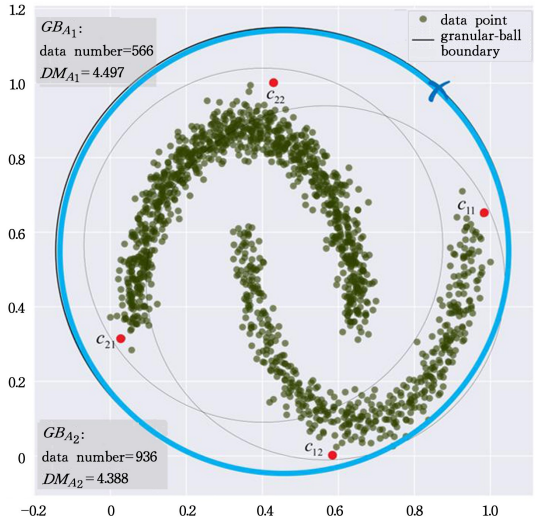
输出：粒球集合 GB_{set}

1. 在 GB_A 中选择最远的两个点 c_1 和 c_2，将 GB_A 分成两个子球 GB_{A_1} 和 GB_{A_2}；
2. 利用式(5)－式(8)计算 DM_A 和 DM_{child}；
3. if $DM_{\text{child}}>DM_A$ then
4. 将 GB_A 分割为 GB_{A_1} 和 GB_{A_2}；
5. end if
6. 对新生成的 GB_{A_1} 和 GB_{A_2} 重复步骤 1－步骤 5 操作，直到粒球的个数不再改变；

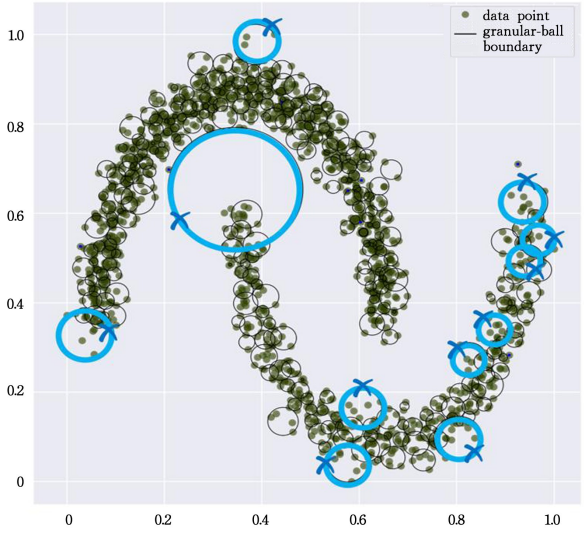
7. for 遍历每个 GB_j
8. 计算 mean(r)，median(r)；
9. if $r_j > 2 \times \max(\text{mean}(r), \text{median}(r))$
10. then 将 GB_j 分割；
11. end if
12. if 粒球的个数不再改变 then
13. break
14. end for



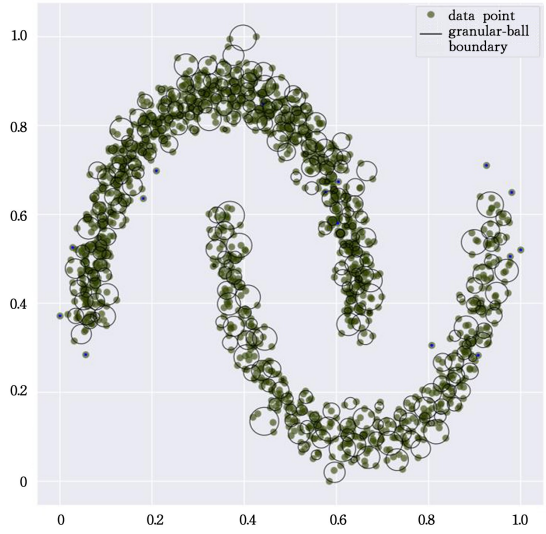
(a)Granular ball A 覆盖整个数据集



(b)Granular ball A 的子球



(c)半径过大的蓝色粒球



(d)归一化的最终结果

图 2 粒球生成过程

Fig. 2 Generation process of granular balls

3.2 基于粒球的 DENCLUE 算法

大多数传统的聚类算法的输入都是点或者像素。受到粒计算的启发，我们认为可将传统结构转化为新结构。在新结构中，使用粒球来生成每个子数据集的一般描述，将原本数据点的输入改进为粒球中心点的输入。

两个粒球的距离定义为两个粒球中心之间的距离减去两个粒球的半径（如果两个球重叠，则距离设为 0）[18]。参与 DENCLUE 粒球构建的粒球不包括所有属于离群集的粒球。两个粒球的距离 $dis(GB_{j_1}, GB_{j_2})$ 和离群集的定义如式(9)和式(10)所示：

$$dis(GB_{j_1}, GB_{j_2}) = dis(c_{j_1}, c_{j_2}) - (r_{j_1} + r_{j_2}) \tag{9}$$

$$outlier = \{GB_j \mid n_j \leqslant 2\} \tag{10}$$

由数据集 $D=\{p_1, p_2, \cdots, p_n\}$ 生成的粒球个数为 m，表示为 $g_j (j=1,2,\cdots,m)$，其将作为 DENCLUE 算法的新输入。

DENCLUE 算法分为两步。第一步是预聚类步骤，构建数据空间相关部分的映射，将数据集的最小边界（超）矩形划分为 d 维网格，边长为 $2h$，并且只考虑实际包含数据点的网格；此外，网格根据它们相对于给定原点的相对位置进行编号。为了加快访问速度，需要连接相邻的网格。若 $dis(mean(c_1), mean(c_2)) < 4h$，则网格 c_1 和网格 c_2 是连通的。

第二步是聚类过程，只考虑高密度的网格以及邻近高密度网格的网格。利用算法 1 构造的粒球，能够高效地计算出局部密度函数和局部梯度，从而找到密度吸引子和吸引点。最后，使用启发式算法进行聚类。

GBDEN 算法的步骤如算法 2 所示。

算法 2 GBDEN 算法

输入：GB_{set}，h，ξ

输出：聚类结果

1. 将数据集划分为边长为 2h 的 d 维网格，只考虑点数超过参数确定的阈值 ξ 的网格；
2. 计算每个填充网格的平均值；
3. 找到高密度的网格；
4. 确定各高密度的网格与其它网格的连接；
5. if dis(mean(c_1)，mean(c_2))<4h
6. then 网格 c_1 和网格 c_2 是连通的；
7. 只考虑高密集网格以及与高密集网格连通的网格；
8. 利用式(4)寻找到密度吸引子；
9. 合并具有相同路径的网格以形成聚类。

4 实验验证

4.1 评价指标

本文所用的聚类评价指标为准确率[19](Accuracy，ACC)、调整兰德指数[20](Adjusted Rand Index，ARI)和归一化互信息[21](Normalized Mutual Information，NMI)。

ACC 用于对比预测结果与真实结果，其计算式如下：

$$ACC=\frac{1}{N}\sum_{i=1}^{k}C_i \tag{11}$$

其中，N 表示样本总数，C_i 表示正确划分到类 i 的样本个数，k 表示类簇数。ACC 值越大，说明聚类结果越好。

ARI 通过计算位于同一类簇和不同类簇中的样本点对的数量来度量两个类簇结果之间的相似性，其计算式如下：

$$ARI=\frac{2(ad-bc)}{(a+b)(b+d)(a+c)(c+d)} \tag{12}$$

其中，a 表示真实情况和实验情况下属于同一类簇的点对数量，b 表示真实情况下属于同一类簇而实验情况下不属于同一类簇的点对数量，c 表示不属于真实情况而实验情况下属于同一类簇的点对数量，d 表示在真实情况和实验情况下不属于同一类簇的点对的数量。ARI 的取值范围为[−1，1]，取值越大，聚类效果越好。

NMI 用于度量两个聚类结果的相似度，其计算式如下：

$$NMI=\frac{2I(Y;C)}{H(Y)+H(C)} \tag{13}$$

$$H(X)=-\sum_{i=0}^{|X|}P(i)\log_2 P(i) \tag{14}$$

$$I(Y;C)=H(Y)-H(Y|C) \tag{15}$$

其中，Y 表示真实数据的类别，C 表示聚类结果，$H(\cdot)$表示交叉熵，$I(Y;C)$表示互信息。NMI 的取值范围为[0，1]，该值越大，与实际结果越一致，即聚类效果越好。

4.2 实验部署

为了在实验中验证 GBDEN 算法的聚类性能，选取了 8 组合成数据集和 5 组 UCI 真实数据集进行了实验。详细情况分别如表 1 和表 2 所列。

表 1 合成数据集的信息

Table 1 Information of synthetic datasets

Datasets	Instances	Dimensions	Clusters
D1	1043	2	2
D2	1039	2	4
D3	567	2	2
D4	876	2	2
D5	1741	2	6
D6	1427	2	4
D7	3603	2	3
D8	1020	2	3

表 2 真实数据集的信息

Table 2 Information of real datasets

Datasets	Instances	Dimensions	Clusters
Iris	150	4	3
Wine	178	13	3
Ecoli	336	7	8
Forest	523	28	4
Dermatology	366	34	6

在 DENCLUE 聚类算法中，聚类结果的好坏取决于平滑参数 h 以及阈值 ξ 的选择。其中，h 确定一个点在其邻域中的影响，h 的值太小，这个点的影响会扩大，会产生过多的聚类中心点；反之，则只会产生一个聚类中心点。可以通过考虑不同数值的 h，来确定 h_{max} 与 h_{min} 的最大间隔，其中密度吸引子的数量 $m(h)$保持不变，这种方法产生的聚类可以看作是对数据集的自然适应。阈值 ξ 描述了密度吸引子是否显著，ξ 的值太小，多个邻近的高密度聚类将被划分到一个类簇中；反之，则会丢失低密度的类簇。假设数据集 D 是无噪声的，那么所有的密度吸引子都是显著的，所以 ξ 的选择范围是 $0\leqslant\xi\leqslant\min\limits_{x^*\in X}\{f^D(x^*)\}$。

在 8 组合成数据集中，都选择 0.05 作为平滑参数 h 的值并选择 0.05 作为阈值 ξ 的值。在 5 组真实数据集中，对关键参数的选择如表 3 所列。

表 3 真实数据集的参数选择

Table 3 Parameters selection of real datasets

Datasets	h	ξ
Iris	0.10	0.01
Wine	0.28	0.01
Ecoli	0.09	0.01
Forest	0.40	0.01
Dermatology	0.20	0.01

此外，为了检验 GBDEN 算法的有效性，在真实 UCI 数据集上将其与经典的 K-MEANS 算法、DBSCAN 算法、SC 算法以及 DENCLUE 算法进行了比较。

4.3 实验结果

4.3.1 准确度分析

GBDEN 算法在合成数据集上的聚类结果如图 3 所示，图中左侧为粒球分割的结果，右侧为对应的粒球聚类结果。

观察粒球分割结果可知，分割得到的粒球所包含的样本数量较为均匀，缓解了传统粒球分割方法中因噪声点和离群点的消极影响而产生的分割不完全粒球。此外，观察粒球的分布情况，分割后的粒球依旧保留了原始的样本空间的分布特征以及样本点之间的结构。因此，本文中优异的粒球分割

策略保证了后续聚类算法的优异表现，减少了离群点和噪声点的影响，使得各类簇之间有着清晰的边界。

此外，使用 ACC，ARI 和 NMI 对上述 5 种算法产生的聚类结果进行分析，如表 4—表 6 所列。由表 4—表 6 可知，GBDEN 在大部分数据集中表现优异，各项指标都优于其他方法，因此算法的适应性较强，在真实场景中具有现实意义。

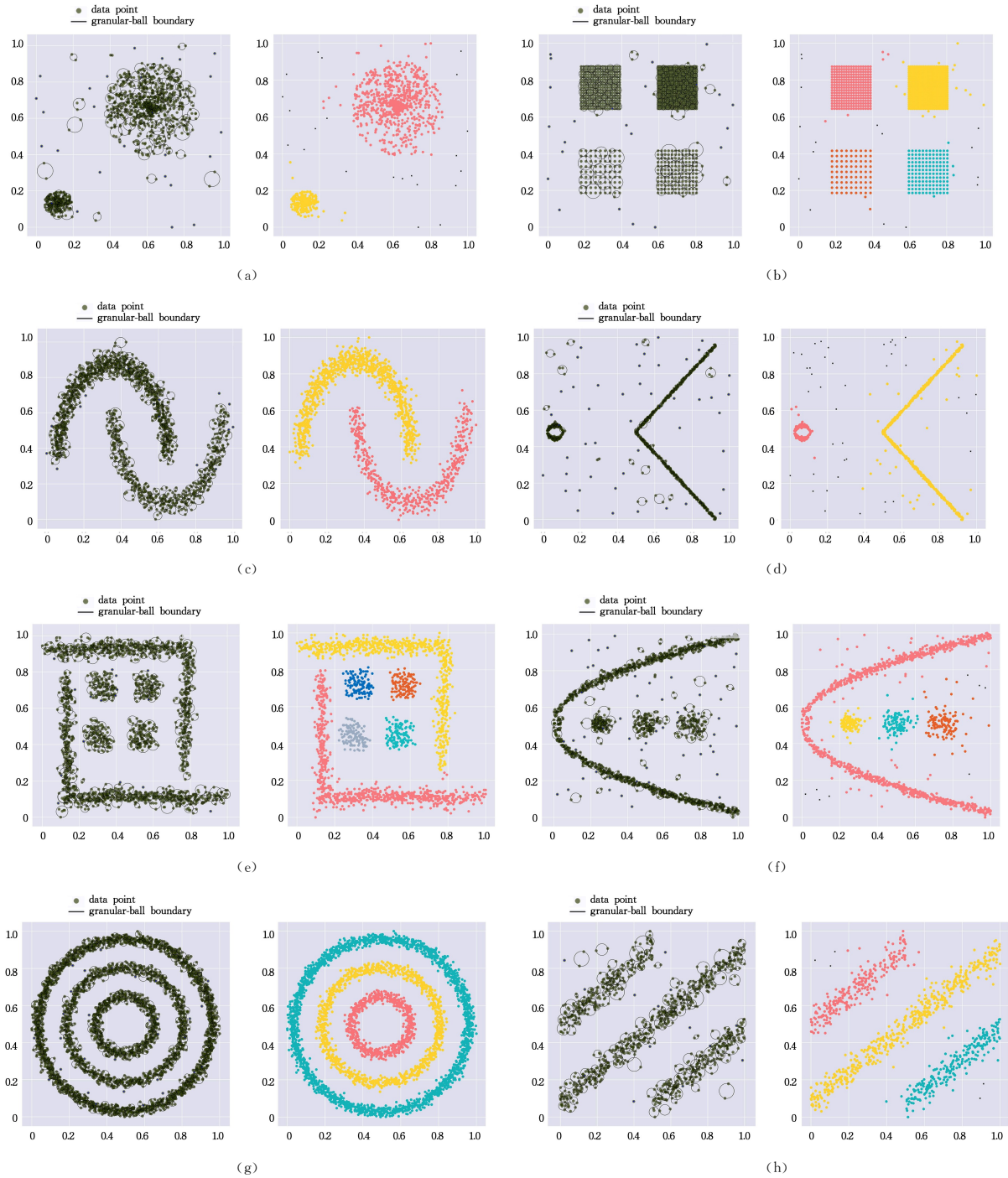


图 3 合成数据集上的粒球构建及粒球聚类

Fig. 3 Construction and clustering of granular balls on synthetic datasets

表 4 ACC 对比

Table 4 Comparison of ACC values

	K-MEANS	DBSCAN	SC	DENCLUE	GBDEN
Iris	0.867	0.667	0.621	0.681	**0.926**
Wine	0.605	0.701	**0.978**	0.674	0.921
Ecoli	0.742	0.485	0.577	0.649	**0.744**
Forest	**0.908**	0.809	0.826	0.785	0.792
Dermatology	0.779	0.495	0.711	0.819	**0.822**

表 5 ARI 对比

Table 5 Comparison of ARI values

	K-MEANS	DBSCAN	SC	DENCLUE	GBDEN
Iris	0.716	0.568	0.623	0.564	**0.826**
Wine	0.869	0.423	**0.931**	0.477	0.771
Ecoli	0.686	0.498	0.434	0.435	**0.702**
Forest	0.791	0.659	0.667	**0.825**	0.815
Dermatology	0.743	0.392	0.561	0.721	**0.788**

表 6 NMI 对比

Table 6 Comparison of NMI values

	K-MEANS	DBSCAN	SC	DENCLUE	GBDEN
Iris	0.742	0.734	0.658	0.717	**0.834**
Wine	0.853	0.526	**0.909**	0.628	0.741
Ecoli	0.649	0.542	0.563	0.539	**0.665**
Forest	0.777	0.683	0.683	**0.789**	0.706
Dermatology	0.834	0.624	0.748	0.805	**0.845**

4.3.2 时间和空间复杂度分析

针对传统 DENCLUE 算法，算法中的核密度估计以及全局的密度梯度计算使得自身的时间复杂度较高，其时间复杂度为 $O(d^2 Nlog(N))$，空间复杂度是 $O(d^2 N)$。而 GBDEN 算法采用粒球来表征全局样本，且粒球生成算法可以独立于后续的聚类过程，可以先行计算。对于 N 个样本的 d 维的数据集，假设生成了 N' 个粒球，则 GBDEN 算法对应的时间复杂度为 $O(d^2 N'\log(N'))$，空间复杂度为 $O(d^2 N')$，其中粒球的数量 N' 远小于样本数量 N。因此，GBDEN 算法在时间和空间复杂度上较小。

本文选用了不同的合成数据集，分别统计了采取传统 DENCLUE 算法与 GBDEN 算法的运行时间，如图 4 所示。

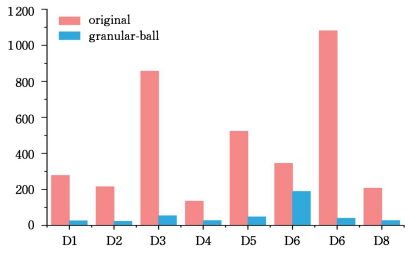


图 4 时间消耗的比较

Fig. 4 Comparison of time consumption

由图 4 可知，在 8 个合成数据集上，GBDEN 运行的时间相较于 DENCLUE 显著减少。特别是对数据集 D7，当样本点的数量增加时，传统的 DENCLUE 算法所需运行时间大幅上升，聚类过程异常缓慢。而 GBDEN 的运行时间仅为原始方法的 1/20，展示了采用粒球的密度和密度梯度的策略的高效性。

此外，实验统计了合成数据集的样本数量以及生成的粒球的数量，如图 5 所示。

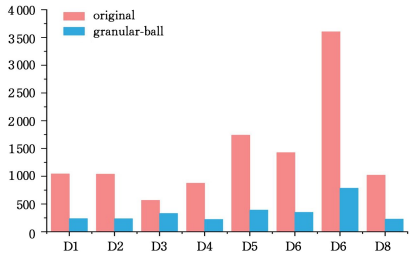


图 5 合成数据集上的样本数量的比较

Fig. 5 Comparison of sample size on synthetic datasets

如图 5 所示，相较于合成数据集的样本数量，经过粒球分割后，以粒球中心为数据集的样本数量最少可以减少到原样本的 1/2，最多大约可以减少到原样本的 1/5。而真实数据集的样本数量至少可以减少到原样本的 1/4，如图 6 所示。相较于传统的 DENCLUE 算法，GBDEN 算法所消耗的存储空间大大减少。

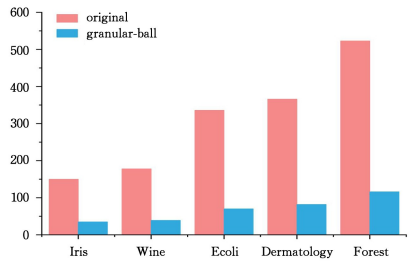


图 6 真实数据集上的样本数量的比较

Fig. 6 Comparison of sample size on real datasets

可以验证，GBDEN 算法在提高聚类精度的条件下，还提高了算法的聚类性能。

4.4 参数敏感度

本节检查了 GBDEN 的参数 h 和阈值 ξ 的灵敏度，并选取一定范围内的参数进行实验。图 7 展示了当 GBDEN 使用 h 从 0.3 到 0.4(步长为 0.01)，和使用 ξ 从 0 到 0.03(步长为 0.0003)的 NMI。在图中可以清晰地看出，NMI 随 h 值的增加有显著的变化，而每一个 h 值的 ξ 所对应的 NMI 并未有明显变化。故可以得出：当 h 取 0.4 时能产生更高的 NMI，GBDEN 对 ξ 的不同值略微敏感。

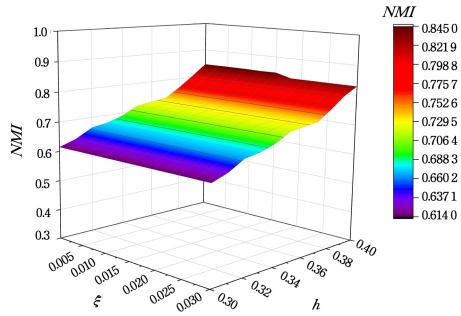


图 7 参数敏感

Fig. 7 Parameter sensitivity

结束语 本文提出了一种基于粒球的 DENCLUE 算法，即 GBDEN。该算法基于多粒度的思想，利用粒球来对无监督数据进行聚类，从而大大降低了时间复杂度和空间复杂度。实验表明，该算法在时间和精度方面都取得了较好的效果。本文中使用的是全局平滑参数 h，有时也可能需要对不同维度使用不同的 h 来更好地适应数据的特点，从而优化聚类效果。

参 考 文 献

[1] ZHANG D Q, CHEN S C. A novel kernelized fuzzy c-means algorithm with application in medical image segmentation[J]. Artificial Intelligence in Medicine, 2004, 32(1): 37-50.

[2] WANG Q, WANG C, FENG Z Y, et al. A Review of K-means Clustering Algorithm Research[J]. Electronic Design Engineering, 2012, 20(7): 21-24.

[3] CAO G H, JIAO Y Y, CHENG Q. Research on Label Clustering

Based on Agglomerative Hierarchical Clustering Algorithm[J]. New Technology of Library and Information Service,2008(4):23-28.

[4] WANG S,XIN S J,LIU C. Review of Density Peak Clustering Algorithm [J]. Journal of East China Jiaotong University,2023,40(1):106-116.

[5] KRIEGEL H P,KROGER P,SANDER J,et al. Density-based clustering[J]. Wiley Interdisciplinary Reviews:Data Mining and Knowledge Discovery,2011,1(3):231-240.

[6] ZADEH L A. Toward human level machine intelligenceis it achievable? the need for a paradigm shift[J]. IEEE Computational Intelligence Magazine,2008,3(3):11-22.

[7] YAO J T,VASILAKOS A V,PEDRYCZ W. Granular computing:perspectives and challenges[J]. IEEE Transactions on Cybernetics,2013,43(6):1977-1989.

[8] HINNEBURG A,KEIM D A. Ageneral approach to clustering in large databases with noise[J]. Knowledge and Information Systems,2003,5:387-415.

[9] HINNEBURG A,GABRIEL HH. Denclue 2.0:Fast clustering based on kernel density estimation[C]// International Symposium on Intelligent Data Analysis. Berlin,Heidelberg:Springer,2007:70-80.

[10] HE J,PAN W. A DENCLUE based approach to neuro-fuzzy system modeling[C]// 2010 2nd International Conference on Advanced Computer Control. IEEE,2010:42-46.

[11] IDRISSI A,REHIOUI H,LAGHRISSI A,et al. An improvement of DENCLUE algorithm for the data clustering[C]// 2015 5th International Conference on Information & Communication Technology and Accessibility(ICTA). IEEE,Marrakech,Morocco,2015:1-6.

[12] KHADER M S,Al-NAYMAT G. VDENCLUE:An Enhanced Variant of DENCLUE Algorithm[C]// Intelligent Systems and Applications:Proceedings of the 2020 Intelligent Systems Conference(IntelliSys). Springer,2021:425-436.

[13] XIA S Y,LIU Y,DING X,et al. Granular ball computing classifiers for efficient,scalable and robust learning[J]. Information Sciences,2019,483:136-152.

[14] 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 Transactions on Knowledge and Data Engineering,2020,34(3):1231-1242.

[15] XIA S Y,PENG D W,MENGD Y,et al. A fast adaptive k-means with no Bounds[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2022,44(1):87-99.

[16] HU Q H,YU D R,XIE Z X. Numerical attribute reduction based on neighborhood granulation and rough approximation [J]. Journal of Software,2008,19(3):640-649.

[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 Transactions on Neural Networks and Learning Systems,2023,31(12):4878-4889.

[18] XIE J,KONG W Y,XIA S Y,et al. An Efficient Spectral Clustering Algorithm Based on Granular-Ball[J]. IEEE Transactions on Knowledge & Data Engineering,2023,35(9):9743-9753.

[19] WU M. A Local Learning Approach for Clustering[C]// Twentieth Annual Conference on Neural Information Processing Systems(NIPS 2006). MIT Press,Vancouver,British Columbia,Canada,2007:1529-1536.

[20] STEINLEY D. Properties of thehubert-arable adjusted rand index[J]. Psychological Methods,2004,9(3):386.

[21] MCDAID A F,GREENE D,HURLEY N. Normalized mutual information to evaluate overlapping community finding algorithms[J]. arXiv:1110.2515,2011.

XUE Renxuan, born in 2001, postgraduate. Her main research interests include machine learning and so on.

YI Shichao, born in 1983, Ph.D, associate professor, master supervisor. His main research interests include fast computation, matrix analysis and three-way decision.

(责任编辑:柯颖)