文章编号:1672-3961(2024)03-0012-10 DOI:10.6040/j.issn.1672-3961.0.2023.107

面向密度峰值聚类的高效相似度度量

王丽娟[1,2],徐晓[1*],丁世飞[1]
(1.中国矿业大学计算机科学技术学院,江苏 徐州 221116; 2.徐州工业职业技术学院信息工程学院,江苏 徐州 221114)

摘要:针对密度峰值聚类(density peaks clustering, DPC)计算复杂度高的问题,提出一种面向密度峰值聚类的高效相似度度量(efficient similarity measure, ESM)法,通过仅度量最近邻之间的相似度构建不完全相似度矩阵。最近邻的选择基于一个随机第三方数据对象,无需另外引入参数。基于 ESM 法构建相似度矩阵,提出一种改进的高效密度峰值聚类(efficient density peaks clustering, EDPC)算法,在保持准确率的同时提高 DPC 识别聚类中心的效率。理论分析和试验结果表明,ESM 法通过减少一定不相似的相似度,可以有效提高 DPC 及其改进算法基于 *K* 最近邻的密度峰值聚类(density peaks clustering based on *K*-nearest neighbors, DPC-KNN)和模糊加权 *K* 最近邻密度峰值聚类(fuzzy weighted *K*-nearest neighbors density peaks clustering, FKNN-DPC)的计算效率,具有较强的可扩展性。

关键词:密度峰值聚类;聚类中心;相似度矩阵;计算复杂度;大规模数据集
中图分类号:TP391 **文献标志码**:A
引用格式:王丽娟,徐晓,丁世飞. 面向密度峰值聚类的高效相似度度量[J].山东大学学报(工学版),2024,54(3):12-21.
WANG Lijuan, XU Xiao, DING Shifei. Efficient similarity measure for density peaks clustering[J]. Journal of Shandong University (Engineering Science), 2024, 54(3):12-21.

Efficient similarity measure for density peaks clustering

WANG Lijuan[1,2], XU Xiao[1*], DING Shifei[1]
(1. School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, Jiangsu, China; 2. School of Information Engineering, Xuzhou College of Industrial Technology, Xuzhou 221114, Jiangsu, China)

Abstract: An efficient similarity measure (ESM) method was proposed for density peaks clustering (DPC) to address the issue of high computational complexity. The ESM method constructed an incomplete similarity matrix by only measuring the similarity between nearest neighbors, without the need for additional parameters, based on a randomly selected third-party data object. Based on the similarity matrix constructed by ESM, an improved efficient density peaks clustering (EDPC) algorithm was proposed to improve the efficiency of DPC to identify cluster centers while maintaining accuracy. Theoretical analysis and experimental results proved that the proposed ESM could effectively improve the computational efficiency of DPC and its improved algorithms density peaks clustering based on *K*-nearest neighbors (DPC-KNN) and fuzzy weighted *K*-nearest neighbors density peaks clustering (FKNN-DPC) by reducing certain dissimilar similarity measures. ESM had robust scalability.

Keywords: density peaks clustering; cluster center; similarity matrix; computational complexity; large-scale dataset

0 引言

随着信息技术的高速发展,如何有效挖掘复杂数据的有效信息变得越来越重要[1]。聚类分析利用数据样本之间的相似度将数据集划分为不同的簇[2],使同类簇中样本高度相似,而不同类簇的样本相似度最小化[3],是一种重要的数据挖掘手段[4-5],广泛应用于社区发现、生物医学、图像处理等领域[6-8]。

传统聚类算法一般可分为 4 种[9-11]:基于划分的聚类算法,例如 *K*-means 算法;基于层次的聚类算

收稿日期:2023-05-29
基金项目:国家自然科学基金资助项目(62206296);中央高校基本科研业务费专项资金资助项目(2022QN1095);江苏省高等职业院校专业带头人高端研修资助项目(2022GRFX063)
第一作者简介:王丽娟(1981—)女,江苏赣榆人,博士研究生,主要研究方向为机器学习和聚类分析。E-mail:327732566@qq.com
*通信作者简介:徐晓(1992—)女,江苏海门人,讲师,硕士生导师,博士,主要研究方向为数据挖掘和机器学习。
E-mail:xu_xiao@cumt.edu.cn

法,例如基于代表对象的聚类(clustering using representative, CURE)算法;基于网格的聚类算法,例如统计信息网格(statistical information grid, STING)算法;基于密度的聚类算法,例如含噪声应用的基于密度的空间聚类(density-based spatial clustering of applications with noise, DBSCAN)算法。然而,大多数传统聚类算法在处理任意形状数据时结果不尽人意,不同的输入参数往往导致不同的聚类结果,因此,如何设置输入参数也是聚类任务面临的巨大挑战。

针对任意形状的数据集,密度峰值聚类(density peaks clustering, DPC)不需要先验知识确定类的簇数,且性能优异,近年引起了广大学者的兴趣[12-13]。文献[14]结合 K 最近邻(K-nearest neighbor, KNN)提出一种基于 K 最近邻的密度峰值聚类(density peaks clustering based on K-nearest neighbors, DPC-KNN)算法,通过 KNN 为 DPC 定义一种新的局部密度,在处理密度不均匀的数据上表现优异;文献[15]提出一种模糊加权 K 最近邻密度峰值聚类(fuzzy weighted K-nearest neighbors density peaks clustering, FKNN-DPC)算法,在 DPC 基础上重新度量局部密度并设计一种新颖的分配策略,提高非中心点的聚类精度;文献[16]提出一种基于快速搜索和密度峰值查找的共享最近邻聚类(shared-nearest-neighbor-based clustering by fast search and find of density peaks, SNN-DPC)算法,提高 DPC 在交叉缠绕、多尺度及变化密度处理中的优越性能。

DPC 算法根据聚类中心密度大于周围邻居点,且与其他密度大的点相对较远的属性绘制决策图,准确识别聚类中心[17-18]。将剩余的数据对象分配到其最近高密度邻居类,去除噪声点,完成对数据集的聚类[19-20]。

DPC 聚类中心的两大属性度量的关键是数据间相似度矩阵的构建[21]。然而,面对大规模数据时,计算复杂度高严重降低了 DPC 的聚类效率[22]。文献[23]设计了一种高效分布式密度峰值聚类(efficient distributed density peaks clustering, EDDPC)算法,通过 Voronoi 分割、数据复制和数据过滤的方式,避免大规模的数据传输耗时和距离计算,极大减少了距离成本,也将大大降低空间复杂度;文献[24]引入非空网格的思想,提出一种基于网格的密度峰值聚类(density peaks clustering based on grid, DPCG)算法,用非空网格代替对应的数据对象进行密度峰值聚类,保持 DPC 特性的同时有效降低 DPC 算法的计算复杂度;文献[25]提出一种基于网格筛选的密度峰值聚类(density peaks clustering based on grid screening, SDPC)算法,根据数据的不均匀分布去除部分密度稀疏网格,使用密度峰值聚类算法中决策图的方法在剩余网格对应的数据集上选取聚类中心,在保持聚类精度的基础上有效降低计算复杂度。尽管上述算法具有理论和实践上的优势,但都引入了新的参数,且大部分研究虽然降低了复杂度,也降低了聚类准确率。因此,面向大规模数据集的 DPC 改进算法研究依旧至关重要。

针对以上问题,本研究提出面向密度峰值聚类的高效相似度度量(efficient similarity measure, ESM)法,通过仅度量最近邻居之间的相似度,构造一个不完整的相似度矩阵,以提高聚类中心的识别效率。其中,最近邻居的选择借助一个第三方点,无需另外引入邻域参数;基于 ESM 法提出一种改进的 DPC 算法,即高效密度峰聚类(efficient density peaks clustering, EDPC)算法,通过减少不必要的相似度计算,构建一个不完全相似度矩阵,在不使用任何其他参数的情况下,降低计算复杂度,保证聚类结果的准确率,提高聚类性能。

1　密度峰值聚类算法

DPC 是一种基于密度的聚类算法,无需先验知识,无需迭代,可以面向任意形状的数据集创建类簇,有且只有一个输入参数[26]。

1.1　DPC 描述

DPC 通过发现高密度峰并快速分配非中心点创建任意形状的类簇,主要思想基于 2 个假设:聚类中心被具有较低密度的数据点包围;聚类中心与具有较高密度的任何其他点之间的距离相对较远[27]。假设数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$,$d_{ij}$ 为数据对象间的相似度,用样本 $\boldsymbol{x}_i$ 和样本 $\boldsymbol{x}_j$ 的欧氏距离表示,则 DPC 可以概括为以下 4 个步骤[28]。

(1)通过度量数据集中所有数据对象之间的欧氏距离构建相似度矩阵

$$\boldsymbol{D}=[\boldsymbol{d}_1 \quad \boldsymbol{d}_2 \quad \cdots \quad \boldsymbol{d}_n]^{\mathrm{T}}\in\mathbf{R}^{n\times n}, \tag{1}$$

式中 $\boldsymbol{d}_i=[d_{i1} \quad d_{i2} \quad \cdots \quad d_{in}]$。

(2)根据聚类中心的 2 个假设,为每个数据点赋予 2 个属性,即局部密度 ρ_i 和相对距离 δ_i,其中局部密度

$$\rho_i=\sum_j \chi(d_{ij}-d_{\mathrm{c}}), \tag{2}$$

式中:d_{c} 为 DPC 的输入参数,表示截断距离,通常为数据对象之间的相似度以升序排序,取 2% 位置处

的值；$\chi(x)=\begin{cases}1, & x<0\\0, & x\geqslant 0\end{cases}$。相对距离

$$\delta_i=\begin{cases}\min\limits_{j:\rho_j>\rho_i}(d_{ij}), & \text{if } \exists j \text{ s.t. } \rho_j>\rho_i\\ \max\limits_{j}(d_{ij}), & \text{otherwise}\end{cases}。\tag{3}$$

（3）根据 ρ_i 和 δ_i 将数据映射到二维决策图，如图 1 所示（分别用红、蓝和黑色圈表示不同类簇的数据），选择 ρ_i 和 δ_i 都较大的数据对象作为聚类中心。从图 1 中可以看出：点 1 和点 10 明显与其他数据点分离，即为聚类中心。

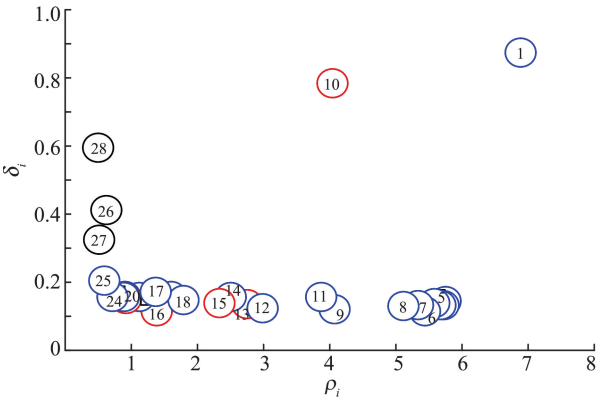


图 1　密度峰值聚类算法决策图

Fig.1　Decision graph of DPC algorithm

（4）按照最近邻分配非中心点，将当前点分配至局部密度大于等于该点的最近点同一类簇。类簇中局部密度小于等于边界阈值的数据对象作为噪声点。

DPC 算法具体步骤如下[29]。

算法 1　DPC 算法

输入　数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$，截断距离 d_c。

输出　聚类结果 $\boldsymbol{Y}$。

（1）度量数据集所有对象间相似度 d_{ij}，构建相似度矩阵 $\boldsymbol{D}$。

（2）基于相似度矩阵，根据式（2）（3）计算每个数据对象的局部密度 ρ_i 和相对距离 δ_i。

（3）基于 ρ_i 和 δ_i 将数据点映射到二维决策图，并在决策图中选择具有较高 ρ_i 和 δ_i 的点为聚类中心。

（4）将非聚类中心点按照最近邻原则分配，去除当前类中的噪声点。

（5）返回聚类结果 $\boldsymbol{Y}$。

1.2　DPC 计算复杂度分析

DPC 通过聚类中心的局部密度和相对距离特点绘制决策图，准确识别聚类中心，可以发现任意形状的簇[30]。局部密度和相对距离的度量依赖于数据对象之间相似度矩阵的构建。假设数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$，DPC 的时间复杂度主要包括：相似度矩阵的构建 $O(n^2)$，也是该算法空间复杂度的主要来源；非聚类中心点的分配 $O(n)$。因此，随着数据规模 n 不断增大，时间复杂度和空间复杂度都将以数据量的二次幂增长，聚类效率受到很大限制[31]。针对 DPC 相似度矩阵的构建，本研究提出一种不完全相似度构建方法，通过仅度量局部密度和相对距离依赖的相似度，在不影响准确率的情况下，提高聚类中心的识别效率。

2　ESM 相似度度量

分析式（2）（3）发现，局部密度 ρ_i 表示距离当前点不超过 d_c 的数据对象的集合，而相对距离 δ_i 表示局部密度大于当前点的最近距离。因此，DPC 的局部密度和相对距离仅取决于当前点与最近邻点的相似度。如何判断最近邻点是关键。最近邻点的选择基于以下定理：如果两个点和同一个其他点的距离差较大，那么这两个点相离较远，即不相似性较大。定理证明如图 2 所示，根据三角形性质，$d_{12}>|d_{1A}-d_{2A}|$。假设存在一个较大的 ξ，如果 $|d_{1A}-d_{2A}|>\xi$，则 $d_{12}>|d_{1A}-d_{2A}|>\xi$，即 d_{12} 较大。值得注意的是，该定理反之不成立，即如果两个点和同一个其他点距离差较小，不能判断这两个点较相似。

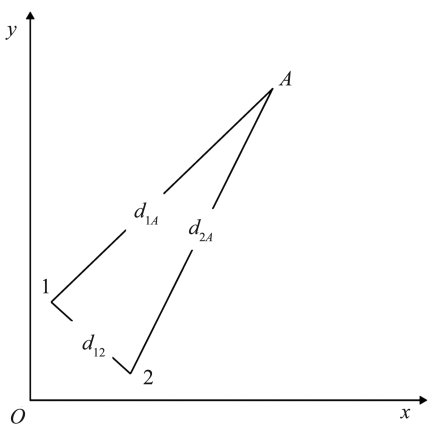


图 2　定理证明图

Fig.2　Theorem proof graph

基于上述定理，本研究提出 ESM 法，假设数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$，则 ESM 法的具体步骤如下。

（1）随机选取第三方点 A，$A=\text{rand}(1,D_N)$，其中 D_N 为数据集的维度。计算所有数据对象与点 A 的欧氏距离，得到距离矩阵

$$\boldsymbol{D}_s=[d_{x_1A}\quad d_{x_2A}\quad\cdots\quad d_{x_nA}],\tag{4}$$

将此距离矩阵中的元素按照从小到大排序，得到一个对应的新数据集

$$\boldsymbol{X}'=\{\boldsymbol{x}_{n1},\boldsymbol{x}_{n2},\cdots,\boldsymbol{x}_{nn}\},\tag{5}$$

与 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$ 存在一一对应关系。

(2) 根据排序后新的数据集 $\boldsymbol{X}'=\{\boldsymbol{x}_{n1},\boldsymbol{x}_{n2},\cdots,\boldsymbol{x}_{nn}\}$,以 $0.4n$ 为步长,度量当前点与向后 $0.4n$ 步长内最近邻点之间的相似度,直到度量完所有的数据对象。

ESM 法仅度量当前点与最近邻点的相似度,不需要度量与非近邻点之间的相似度,有效降低相似度的计算复杂度。同时,度量最近邻点之间的相似度可以准确计算局部密度和相对距离,完成聚类中心的高效识别。

3　基于 ESM 改进的密度峰值聚类算法

3.1　EDPC 算法描述

EDPC 算法根据 ESM 法构造一个不完整的相似度矩阵,通过相似度矩阵完成 DPC 聚类。ESM 法的引入减少了 EDPC 算法相似度的度量,不再度量非近邻点之间的相似度,虽然减少了相似度的信息,但是由于 DPC 聚类过程只依赖于高相似点之间的相似度信息,因此 EDPC 算法可以在保持聚类中心准确识别的基础上提高聚类效率。EDPC 算法的具体步骤如下。

算法 2　EDPC 算法

输入　数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$。

输出　相似度矩阵 $\boldsymbol{S}$、聚类结果 $\boldsymbol{Y}$。

(1) 设置 $A=\mathrm{rand}(1,D_{\mathrm{N}})$,计算所有数据对象与点 A 的距离,构建距离矩阵 $\boldsymbol{D}_{\mathrm{s}}=[d_{x_1A}\ d_{x_2A}\cdots d_{x_nA}]$。

(2) 将距离矩阵 $\boldsymbol{D}_{\mathrm{s}}=[d_{x_1A}\quad d_{x_2A}\quad\cdots\quad d_{x_nA}]$ 中的元素排序,得到对应的新数据集 $\boldsymbol{X}'=\{\boldsymbol{x}_{n1},\boldsymbol{x}_{n2},\cdots,\boldsymbol{x}_{nn}\}$。

(3) 设置步长为 $0.4n$,从 $\boldsymbol{X}'=\{\boldsymbol{x}_{n1},\boldsymbol{x}_{n2},\cdots,\boldsymbol{x}_{nn}\}$ 开始,度量当前点与后几步之内的数据点之间的相似度。

(4) 当遍历完所有数据对象,构建不完全相似度矩阵 $\boldsymbol{S}$。

(5) 根据 $\boldsymbol{S}$ 计算 ρ_i 和 δ_i,绘制决策图识别聚类中心。

(6) 根据算法 1 分配非中心点,去除噪声数据。

(7) 返回相似度矩阵 $\boldsymbol{S}$ 及聚类结果 $\boldsymbol{Y}$。

3.2　EDPC 计算复杂度分析

EDPC 算法基于 ESM 法改进了 DPC,借助第三方数据点判断数据对象的近邻点,即减少不相似点之间的相似度计算。由于局部密度和相对距离的计算只依赖于近邻对象之间的相似度,因此基于 ESM 法构造的不完全相似度矩阵在保证聚类准确率的同时有效降低了 EDPC 算法的计算复杂度。

假设数据集包含 n 个数据对象。EDPC 算法主要的时间复杂度来源于相似度矩阵的构建。判断数据对象的近邻点需要的时间复杂度为 $O(n)$;计算所有数据对象到第三方数据点的距离,根据步长 $0.4n$ 分别计算当前点与最近邻点的相似度,以时间复杂度 $O(0.32n^2-0.8n)$ 构建相似度矩阵,需要空间复杂度 $O(0.32n^2-0.8n)$ 存储相似度矩阵。因此 EDPC 算法的计算复杂度包括时间复杂度 $O(0.32n^2+0.2n)$ 和空间复杂度 $O(0.32n^2-0.8n)$。与 DPC 算法相比,EDPC 算法聚类中心的识别和非聚类中心的分配都相同,因此,计算复杂度明显低于 DPC 算法。

EDPC 算法只需要很少的相似度计算,成功降低了计算复杂度。另外,ESM 法构建的相似度矩阵保持了计算局部密度和相对距离的精度,保证了 EDPC 算法的聚类精度。

4　试验与分析

为了验证基于 ESM 改进的密度峰值聚类算法的有效性,将试验主要分为 2 部分:第一,分别从聚类准确率和运行时间 2 个角度在 6 个人工数据集和 6 个 UCI 数据集上验证 EDPC 算法的聚类性能,选取 DPC、SDPC、FKNN-DPC 算法作为对比算法,均使用适合数据集的相同参数 d_{c},SDPC 算法引入新的筛选比例参数,通常设置为 70%;第二,为了证明 ESM 法针对密度峰值聚类的通用性,将其应用于 DPC 改进算法 DPC-KNN 和 FKNN-DPC 的相似度计算,通过分析改进算法的结果证明 ESM 法的有效性。在 DPC-KNN 和 FKNN-DPC 算法的聚类试验中,根据参考文献[14-15]使用合适的参数 d_{c} 并设置合适的近邻参数。

仿真试验在 i5 2.3 GHz 处理器、8GB RAM、macOS 10.14.5 操作系统和 MATLAB 2015 的环境下进行。为了公平起见,算法的运行时间取重复 10 次的平均值,所有算法的相似度矩阵计算均使用原始的迭代循环。

4.1　人工数据集试验

为了证明 EDPC 算法的有效性,通过 6 个人工数据集的可视化,比较 EDPC、DPC、SDPC 及 FKNN-DPC 算法的聚类精度及时间复杂度。数据集的规模由小到大不等,聚类数量不同,如表 1 所示。

表 1 人工数据集

Table 1 Characteristic of artificial datasets

单位:个

数据集	样本数	特征数	类别数
Spiral	312	2	3
Aggregation	788	2	7
Twenty	1 000	2	20
A1	3 000	2	20
S3	5 000	2	15
A3	7 500	2	50

EDPC、DPC、SDPC 和 FKNN-DPC 算法在二维嵌入数据上的聚类结果将通过可视化彩色图展现，如图 3~8 所示。

由图 3~8 可以明显看出:EDPC 算法与 DPC 算法均具有较好的聚类结果,并且上述 4 种算法在大部分数据集上都非常相似;仅在 Spiral 数据集上,SDPC 算法的性能不及 EDPC 和 DPC,这是由于 SDPC 算法采用的网格筛选方法不曾考虑流形数据的全局分布;在 S3 数据集上,SDPC 算法的聚类精度略有降低。为进一步评估算法性能,分别对比各个算法的聚类精度 A_{CC},如表 2 所示,A_{CC}越大,聚类效果越优。

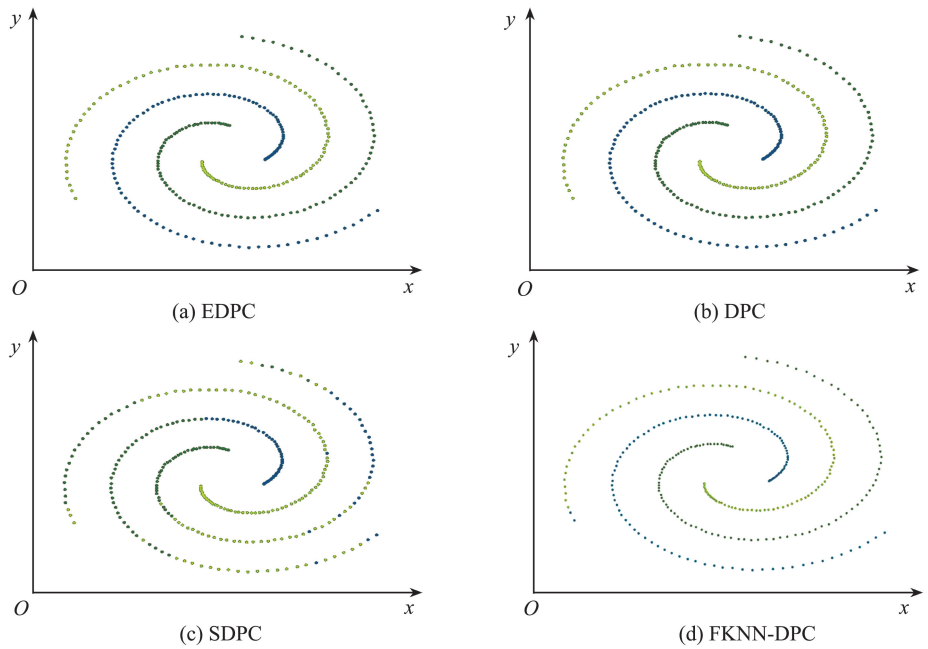


图 3 各算法在 Spiral 数据集上的聚类结果

Fig.3 Clustering results of different algorithms on Spiral dataset

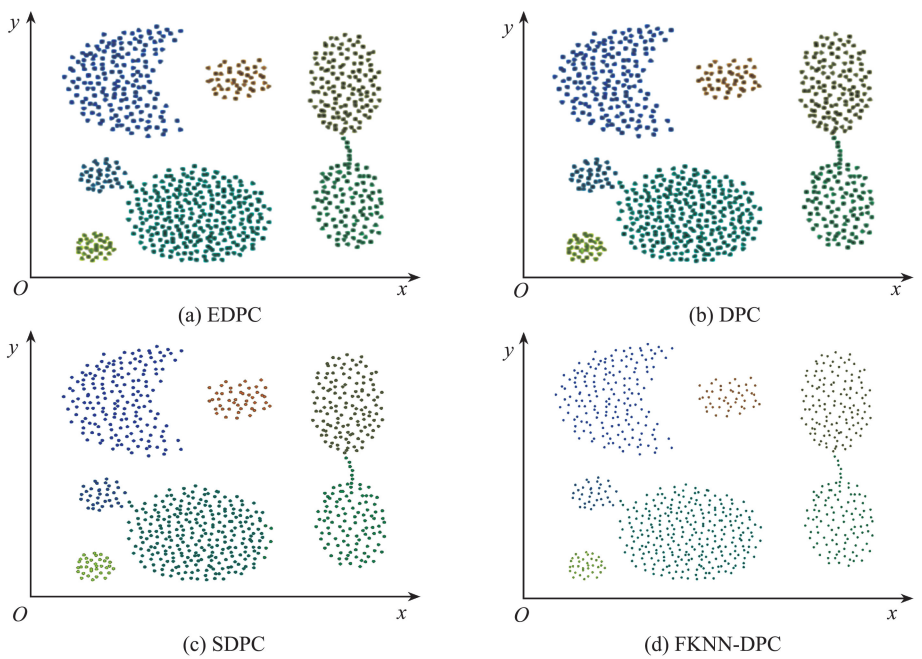
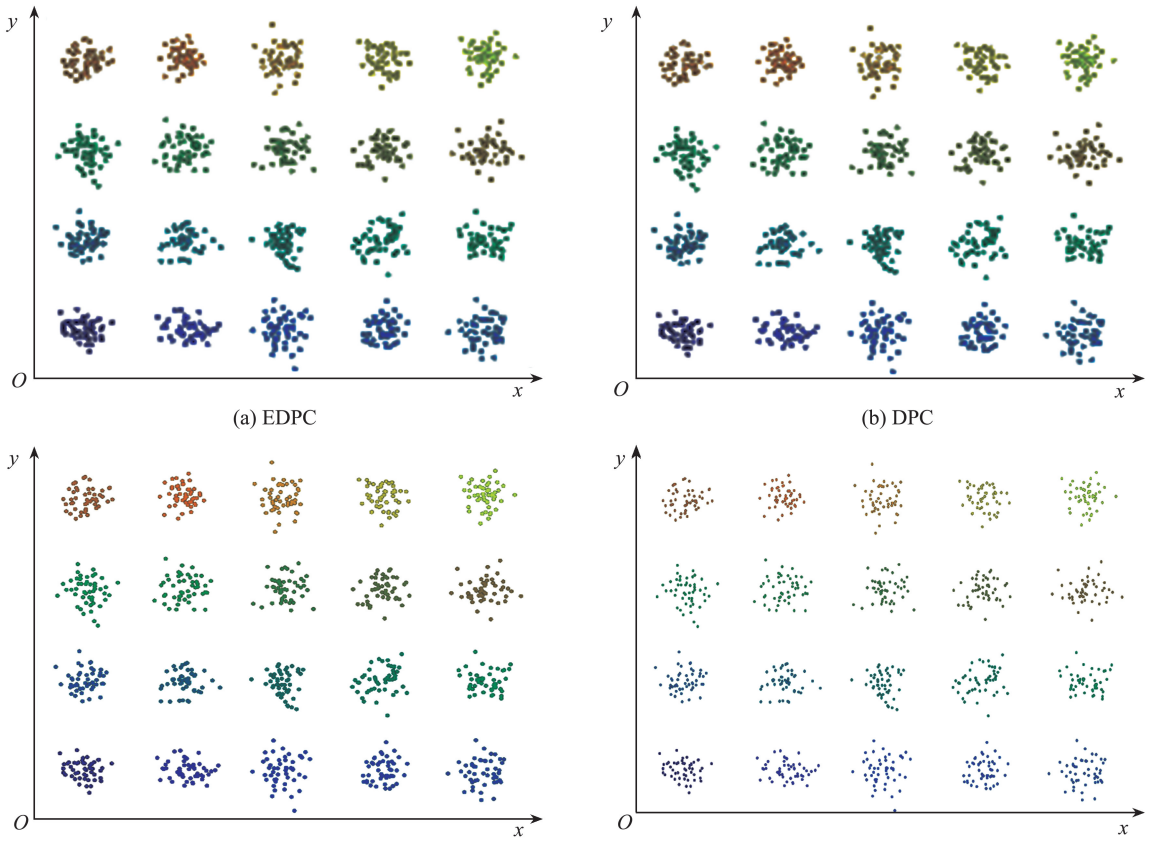


图 4 各算法在 Aggregation 数据集上的聚类结果

Fig.4 Clustering results of different algorithms on Aggregation dataset

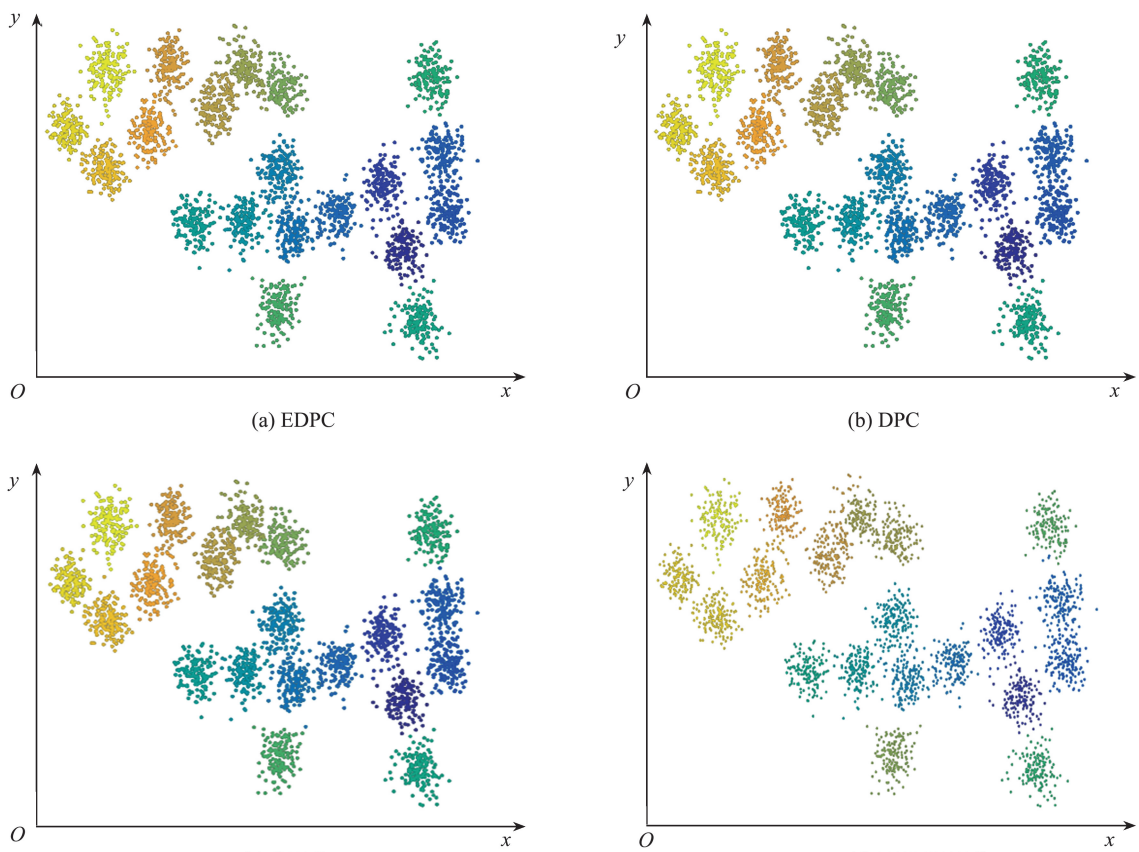


(a) EDPC　(b) DPC

(c) SDPC　(d) FKNN-DPC

图 5　各算法在 Twenty 数据集上的聚类结果

Fig.5　Clustering results of different algorithms on Twenty dataset

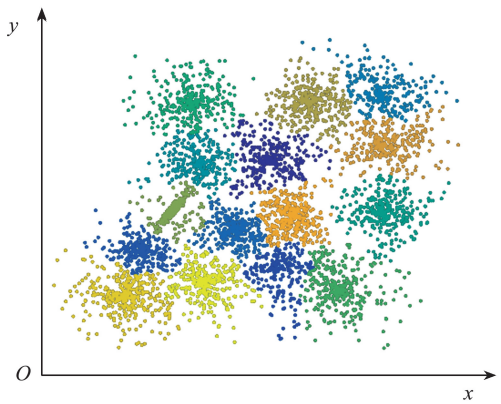


(a) EDPC　(b) DPC

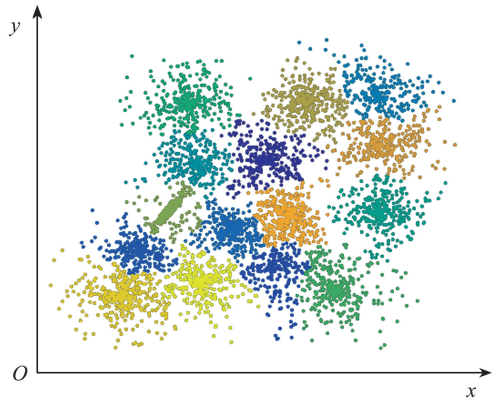
(c) SDPC　(d) FKNN-DPC

图 6　各算法在 A1 数据集上的聚类结果

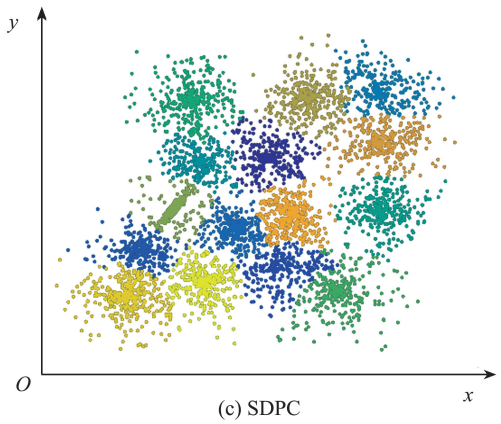
Fig.6　Clustering results of different algorithms on A1 dataset



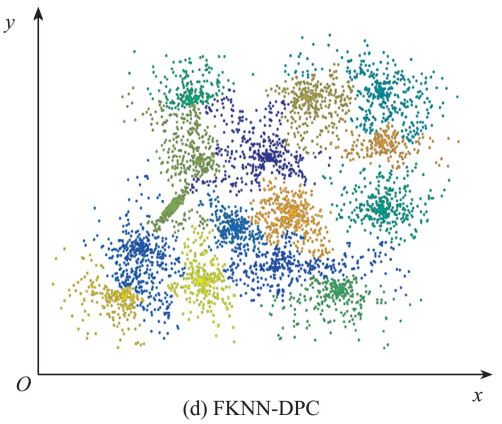
(a) EDPC



(b) DPC



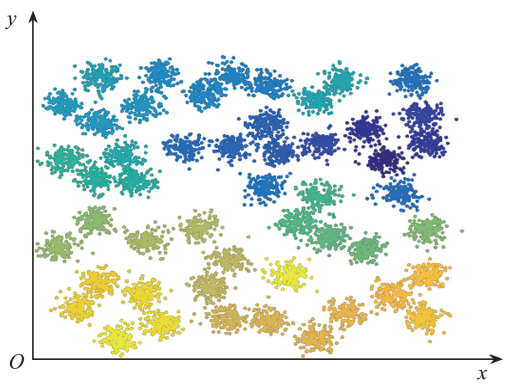
(c) SDPC



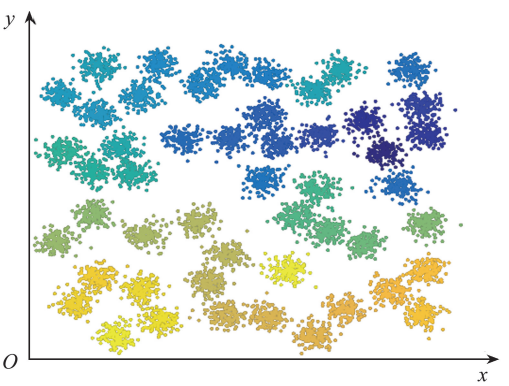
(d) FKNN-DPC

图 7 各算法在 S3 数据集上的聚类结果

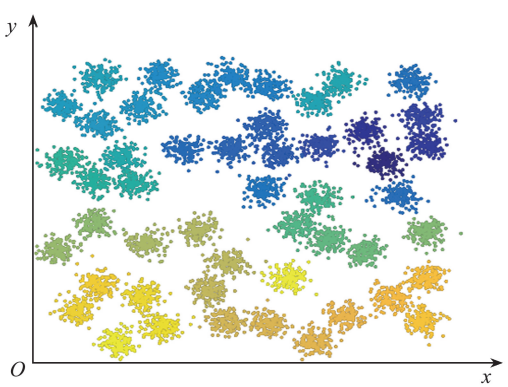
Fig.7 Clustering results of different algorithms on S3 dataset



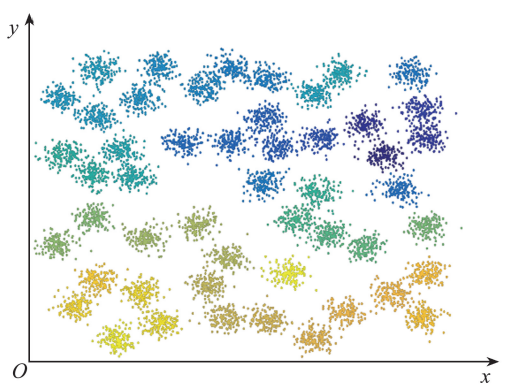
(a) EDPC



(b) DPC



(c) SDPC



(d) FKNN-DPC

图 8 各算法在 A3 数据集上的聚类结果

Fig.8 Clustering results of different algorithms on A3 dataset

表 2　人工数据集上不同算法的 A_{CC}
Table 2　A_{CC} of different algorithms on artificial datasets

数据集	A_{CC}			
	EDPC	DPC	SDPC	FKNN-DPC
Spiral	1.000	1.000	0.561	0.994
Aggregation	0.995	0.995	0.989	0.995
Twenty	1.000	1.000	1.000	1.000
A1	0.983	0.983	0.994	0.890
S3	0.855	0.855	0.852	0.738
A3	0.990	0.990	0.989	0.940

由表 2 可以看出:在这些人工数据集上,EDPC 算法可以保持与 DPC 相同的结果;尽管 SDPC 算法的聚类结果接近 EDPC,但与 EDPC 相比,聚类准确性有所降低,因为 SDPC 算法采用的数据信息较少,无法找到准确的聚类中心;FKNN-DPC 算法由于引入了额外的参数,受参数设置的影响,获取最优结果具有一定的挑战性。EDPC 算法采用 ESM 法的相似度度量方式构建相似度矩阵,满足聚类中心的特点,因此,EDPC 算法保持了 DPC 的聚类准确率。另外,本研究评估了 EDPC、DPC、SDPC 和 FKNN-DPC 算法的聚类效率,分别对比各个算法在这 6 个人工数据集上的运行时间,如表 3 所示。

表 3　各算法在人工数据集上的聚类时间
Table 3　Clustering time of different algorithms on artificial datasets

数据集	聚类时间/s			
	EDPC	DPC	SDPC	FKNN-DPC
Spiral	0.18	0.21	0.40	0.42
Aggregation	0.53	0.68	1.08	1.06
Twenty	0.82	1.15	0.94	1.78
A1	6.28	8.65	5.97	21.98
S3	17.51	25.53	17.87	117.39
A3	38.31	55.90	29.12	441.88

由表 3 明显看出:在不同数据集上,EDPC 算法的效率均比 DPC 和 FKNN-DPC 高;虽然 DPC 算法在这 6 组数据集上也表现出良好的聚类性能,但随着数据规模的增大,其时间消耗呈指数上升;SDPC 算法在 A1 和 A3 数据集上比 EDPC 效率更高,这是由于 A1 和 A3 数据集的分布适合网格筛选方法,SDPC 算法采用网格仅筛选部分数据集进行聚类中心的选择。然而,SDPC 算法引入了需要预先指定网格大小。

综合图 3~8 和表 2、3,就聚类准确性和计算复杂性而言,EDPC 算法是最佳算法,其性能优于 DPC、SDPC 和 FKNN-DPC。EDPC 算法采用 ESM 计算相似度矩阵,得到的相似度信息可以保持 DPC 算法的聚类精度,并且由于只计算最近邻的相似度,因此有效降低了计算复杂度。

4.2　UCI 数据集试验

为了进一步证明 EDPC 算法的有效性,在 6 个大小不同的 UCI 数据集上评估了 EDPC、DPC、SDPC 和 FKNN-DPC 算法,数据特征如表 4 所示。

表 4　UCI 试验数据特征
Table 4　Characteristic of UCI datasets 单位:个

数据集	样本数	特征数	类别数
Iris	150	4	3
Transfusion	784	4	2
Segmentation	2 310	19	7
Twonorm	7 400	20	2
Pendigits	10 992	16	10
Gamma	19 020	10	2

EDPC、DPC、SDPC 和 FKNN-DPC 算法的聚类准确率使用通用的聚类指标 A_{CC} 和调整互信息分数 A_{MI},基准值越大,聚类性能越好。由于 Iris 和 Seeds 数据集的数据规模较小,因此在这 2 个数据集上 SDPC 算法的筛选比例取 30%。EDPC 算法及各对比算法的 A_{CC} 和 A_{MI} 如表 5、6 所示。

表 5　UCI 数据集上不同算法的 A_{CC}
Table 5　A_{CC} of different algorithms on UCI datasets

数据集	A_{CC}			
	EDPC	DPC	SDPC	FKNN-DPC
Iris	0.940	0.940	0.833	0.933
Transfusion	0.767	0.767	0.767	0.767
Segmentation	0.597	0.598	0.550	0.519
Twonorm	0.963	0.963	0.965	0.965
Pendigits	0.650	0.650	0.641	0.560
Gamma	0.649	0.649	0.656	0.649

表 6　UCI 数据集上不同算法的 A_{MI}
Table 6　A_{MI} of different algorithms on UCI datasets

数据集	A_{MI}			
	EDPC	DPC	SDPC	FKNN-DPC
Iris	0.823	0.823	0.694	0.804
Transfusion	0.012	0.012	0.012	0.012
Segmentation	0.564	0.564	0.514	0.400
Twonorm	0.772	0.771	0.783	0.785
Pendigits	0.682	0.683	0.654	0.613
Gamma	0.008	0.008	0.020	0.020

由表 5、6 可以看出:EDPC 算法采用 ESM 法构建的相似度矩阵对聚类中心的选取几乎没有影响,EDPC 算法在这些数据集上保持着和 DPC 几乎一样的聚类精度;但是 SDPC 算法减少了用于聚类中心选择的数据信息,聚类精度在一定程度上受到影响,例如其在 Iris 数据集上影响较大;由于 Gamma 数据集高维稀疏,而 SDPC 算法采用的分配策略比

EDPC 和 DPC 单步分配策略更合适,因此 SDPC 算法精度略高,但差异几乎可以忽略不计;由于分配策略的优化,在 Twonorm 数据集上 FKNN-DPC 与 SDPC 算法优于 EDPC 和 DPC。但是,FKNN-DPC 与 SDPC 算法均引入了额外的近邻参数,保持优越的聚类性能需要不断进行参数的选择,对聚类任务具有一定的挑战性。

总体比较,EDPC 算法聚类结果令人满意。此外,本研究分别比较了 EDPC 与 DPC、SDPC 和 FKNN-DPC 的聚类时间,如表 7 所示。

表 7 各算法在 UCI 数据集上的聚类时间

Table 7 Clustering time of different algorithms on UCI datasets

数据集	聚类时间/s			
	EDPC	DPC	SDPC	FKNN-DPC
Iris	0.11	0.13	0.34	0.31
Transfusion	0.51	0.75	1.11	1.12
Segmentation	4.11	5.35	3.52	11.94
Twonorm	39.13	56.41	48.91	429.95
Pendigits	88.60	121.33	108.62	1.39×10^3
Gamma	302.12	486.17	322.81	7.22×10^3

表 7 中 EDPC 算法和 DPC、SDPC、FKNN-DPC 的运行效率在较小规模的数据集上旗鼓相当。但随着数据规模的增大,EDPC 算法明显快于 DPC、SDPC 和 FKNN-DPC,因为 EDPC 算法只计算了最近邻之间的相似度寻找聚类中心,而 DPC 算法计算了所有数据之间的相似度,时间复杂度和空间复杂度都很高;SDPC 算法虽然优于 DPC,但由于参数的选择,在较大规模的数据集上并没有表现出优于 EDPC 的特性;FKNN-DPC 算法是一个准确率优化算法,因此牺牲了一定的聚类效率。另外,SDPC 和 FKNN-DPC 算法均引入了额外的参数,聚类结果受参数的影响较大。

综上,EDPC 算法在保持 DPC 聚类准确率的同时有效降低了计算复杂度,无需引入敏感参数,因此 EDPC 算法比 DPC 适合大数据环境下的数据挖掘。

4.3 ESM 试验分析

为了验证 ESM 的泛化能力,将 ESM 应用于 DPC 及其改进算法 DPC-KNN 和 FKNN-DPC,并分别选取 3 组人工数据集和 3 组 UCI 数据集进行对比验证,如表 8 所示。根据 ESM 构建相似度矩阵,按 DPC-KNN 和 FKNN-DPC 的方法进行标签分配,完成对数据集的聚类。EDPC 算法的试验结果同表 3、6。

表 8 各算法在不同数据集上的聚类时间

Table 8 Clustering time of different algorithms on different datasets

数据集	聚类时间/s					
	DPC	EDPC	DPC-KNN	ESM-DPC-KNN	FKNN-DPC	ESM-FKNN-DPC
Twenty	1.15	0.82	1.19	0.74	1.78	1.31
S3	25.53	17.51	20.17	16.51	117.39	114.39
A3	55.90	38.31	46.53	38.09	441.88	413.93
Iris	0.13	0.11	0.29	0.11	0.31	0.12
Twonorm	56.41	39.13	52.07	41.75	429.95	415.89
Gamma	486.17	302.12	400.49	359.68	7.22×10^3	7.04×10^3

由表 8 可以看出:在 6 组数据集上,采用 ESM 法构建相似度矩阵提高了原算法的效率。ESM 法通过度量最近邻点间的相似度识别聚类中心,可以有效减少相似度的度量。另外,由于 ESM 法获取了识别聚类中心必要的距离信息,改进后的方法可以保持原算法的聚类精度,因此,ESM 法具有较强的可扩展性,可以在不引入敏感参数的情况下保持聚类精度,降低 DPC 及其相关算法的计算复杂度,ESM 法的提出具有实际意义。

5 结论

本研究提出一种面向密度峰值聚类的高效相似度度量方法 ESM,借助第三方数据点发现最近邻,通过仅度量最近邻的相似度,无需引入额外参数,提高了聚类中心的识别效率。将 ESM 法与 DPC 结合,提出一种高效密度峰值聚类算法 EDPC。EDPC 算法通过 ESM 法构建不完整相似度矩阵,大大降低了聚类任务的时间复杂度,提高了聚类效率。通过理论和试验表明,ESM 法通过减少一定不相似数据对象之间的相似度计算,为 DPC 及相关的改进 DPC 算法构建不完整相似度矩阵,可以保证聚类结果准确率,提高聚类效率,降低参数选择对聚类结果的影响。

无论是 DPC 还是改进的 EDPC,通过决策图选择聚类中心时,依然需要依靠用户的经验。因此,

通过扩展高效相似度度量方法提高识别聚类中心的准确率,需要进一步探索。

参考文献:

[1] CHEN J G, PHILIP S Y. A domain adaptive density clustering algorithm for data with varying density distribution [J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(6):2310-2321.

[2] 贾洪杰, 丁世飞, 史忠植. 求解大规模谱聚类的近似加权核 *k*-means 算法[J]. 软件学报, 2015, 26(11):2836-2846.
JIA Hongjie, DING Shifei, SHI Zhongzhi. Approximate weighted kernel *k*-means for large-scale spectral clustering [J]. Journal of Software, 2015, 26(11):2836-2846.

[3] YAN X Q, YE Y D, QIU X Y, et al. Synergetic information bottleneck for joint multi-view and ensemble clustering[J]. Information Fusion, 2020, 56:15-27.

[4] XIE D Y, GAO Q X, WANG Q Q, et al. Adaptive latent similarity learning for multi-view clustering [J]. Neural Networks, 2020, 121:409-418.

[5] QU H, MA T, TONG X Y, et al. Clustering by centroid drift and boundary shrinkage [J]. Pattern Recognition, 2022, 129:108745.

[6] BARANWAL M, SALAPAKA S. Clustering and supervisory voltage control in power systems [J]. International Journal of Electrical Power & Energy Systems, 2019, 109:641-651.

[7] POTHULA K, SMYRNOVA D, SCHRÖDER G. Clustering cryo-EM images of helical protein polymers for helical reconstructions[J]. Ultramicroscopy, 2019, 203:132-138.

[8] SHI Y C, OTTO C, JAIN A. Face clustering: representation and pairwise constraints [J]. IEEE Transactions on Information Forensics and Security, 2018, 13(7):1626-1640.

[9] LLOBELL F, VIGNEAU E, QANNARI E. Clustering datasets by means of CLUSTATIS with identification of atypical datasets. Application to sensometrics [J]. Food Quality and Preference, 2019, 75:97-104.

[10] 史倩玉, 梁吉业, 赵兴旺. 一种不完备混合数据集成聚类算法[J]. 计算机研究与发展, 2016, 53(9):1979-1989.
SHI Qianyu, LIANG Jiye, ZHAO Xingwang. A clustering ensemble algorithm for incomplete mixed data [J]. Journal of Computer Research and Development, 2016, 53(9):1979-1989.

[11] CHEN Y W, TANG S Y, BOUGUILA N, et al. A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data[J]. Pattern Recognition, 2018, 83:375-387.

[12] BRYANT A, CIOS K. RNN-DBSCAN: a density-based clustering algorithm using reverse nearest neighbor density estimates[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(6):1109-1121.

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

[14] DU M J, DING S F, JIA H J. Study on density peaks clustering based on *K*-nearest neighbors and principal component analysis [J]. Knowledge-Based Systems, 2016, 99:135-145.

[15] XIE J Y, GAO H C, XIE W X, et al. Robust clustering by detecting density peaks and assigning points based on fuzzy weighted *K*-nearest neighbors[J]. Information Sciences, 2016, 354:19-40.

[16] LIU R, WANG H, YU X M. Shared-nearest-neighbor-based clustering by fast search and find of density peaks [J]. Information Sciences, 2018, 450:200-226.

[17] DING S F, DU W, XU X, et al. An improved density peaks clustering algorithm based on natural neighbor with a merging strategy[J]. Information Sciences, 2023, 624:252-276.

[18] WANG M, MIN F, ZHANG Z H, et al. Active learning through density clustering[J]. Expert Systems with Applications, 2017, 85:305-317.

[19] XU J, WANG G Y, LI T R, et al. Fat node leading tree for data stream clustering with density peaks[J]. Knowledge-Based Systems, 2017, 120:99-117.

[20] XU J, WANG G Y, DENG W H. DenPEHC: density peak based efficient hierarchical clustering [J]. Information Sciences, 2016, 373:200-218.

[21] WU B, WILAMOWSKI B. A fast density and grid based clustering method for data with arbitrary shapes and noise [J]. IEEE Transactions on Industrial Informatics, 2017, 13(4):1620-1628.

[22] 巩树凤, 张岩峰. EDDPC:一种高效的分布式密度中心聚类算法[J]. 计算机研究与发展, 2016, 53(6):1400-1409.
GONG Shufeng, ZHANG Yanfeng. EDDPC: an efficient distributed density peaks clustering algorithm [J]. Journal of Computer Research and Development, 2016, 53(6):1400-1409.

[23] XU X, DING S F, DU M J, et al. DPCG: an efficient density peaks clustering algorithm based on grid [J]. International Journal of Machine Learning and Cybernetics, 2018, 9(5):743-754.

(下转第 29 页)