计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2023/17(08)-1879-14
doi: 10.3778/j.issn.1673-9418.2205032

开放科学(OSID)

融合相对密度和最近邻关系的密度峰值聚类

王威娜[1+],朱　钰[1],任　艳[2]
1. 吉林化工学院 信息与控制工程学院,吉林 132022
2. 沈阳航空航天大学 人工智能学院,沈阳 110136
+ 通信作者 E-mail: wangweina@jlict.edu.cn

摘　要:密度峰值算法在处理密度不均匀的数据时对中心点的选取不准确,并在样本分配时易产生连带错误,导致聚类效果不佳。针对上述问题,提出一种融合相对局部密度和最近邻关系的密度峰值聚类算法。在局部密度的定义中引入稀疏平和权重,提出相对局部密度的定义,根据相对局部密度寻找密度峰值,避免稀疏差异较大的数据集在选取密度峰值时出现的错误,确保中心点选择的正确性;针对分配策略,结合最邻近点准则和阈值限制,提出最近邻分配策略,根据阈值条件有效抑制分配连带错误;基于类内距离均值定义距离比例,提出修正分配策略,提升算法对边界点聚类的准确性。在5个合成数据集和5个UCI数据集上,将提出算法与DPC、DPC-MND、FKNN-DPC、DBSCAN、OPTICS、AP、*K*-means算法进行比较,实验结果表明,所提算法在调整互信息、调整兰德系数和Fowlkes-Mallows指数上均表现出良好的聚类效果,并通过Friedman检验表明该算法具有最优的性能。

关键词:聚类算法;密度峰值;相对局部密度;最近邻关系;分配策略
文献标志码:A　　**中图分类号:**TP301

Density Peaks Clustering Combining Relative Local Density and Nearest Neighbor Relationship

WANG Weina[1+], ZHU Yu[1], REN Yan[2]
1. College of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin 132022, China
2. College of Artificial Intelligence, Shenyang Aerospace University, Shenyang 110136, China

Abstract: When the density peaks algorithm deals with datasets with different densities, the wrong center points may be selected, and the problem of associated errors may occur in the sample allocation process. To solve the above problems, a density peaks clustering algorithm based on the relative local density and nearest neighbor relationship is proposed. The weights of sparse balance are introduced into the definition of local density, and the definition of relative local density is proposed. The density peak can be found according to the relative local density, which avoids the error of selecting the density peak in the dataset with large sparse differences, and ensures the accuracy of the center point selection. The nearest neighbor allocation strategy is proposed by combining the nearest neighbor criterion and threshold limit to suppress the allocation error effectively. The modified allocation strategy based on the mean value of the distance within the class is proposed to enhance the accuracy of the algorithm for boundary point clustering. The proposed algorithm is compared with DPC, DPC-MND, FKNN-DPC, DBSCAN,

基金项目:国家自然科学基金(61602321);吉林省自然科学基金(YDZJ202201ZYTS603);辽宁省自然科学基金(2020MS235)。
This work was supported by the National Natural Science Foundation of China (61602321), the Natural Science Foundation of Jilin Province (YDZJ202201ZYTS603), and the Natural Science Foundation of Liaoning Province (2020MS235).
收稿日期:2022-05-09　　**修回日期:**2022-08-09

OPTICS, AP, and *K*-means algorithms on 5 synthetic datasets and 5 UCI datasets, and the experimental results demonstrate that the proposed algorithm has sound clustering performance in metrics of adjusted mutual information, adjusted Rand index, and Fowlkes-Mallows index. Friedman test shows that the algorithm has the best performance.

Key words: clustering algorithm; density peaks; relative local density; nearest neighbor relations; allocation strategy

聚类分析[1]根据数据集各点之间的相似性把样本点划分为不同的类簇,其作为数据挖掘的一种有效方法已在众多领域得到广泛应用,如图像处理[2]、工业信息[3]、生物学[4]和计算机视觉[5]等领域。正因其应用的广泛性,使其受到众多学者的青睐,大量的聚类算法先后被提出,依据对样本的处理方式可分为基于划分的聚类算法[6]、基于层次的聚类算法[7]、基于密度的聚类算法[8]、基于网格的聚类算法[9]以及基于模型的聚类算法[10]等。

K-means[11]算法是最常见的基于划分的聚类算法,该算法原理简单易于实现,并具有高效且伸缩性好的优点,但该算法聚类效果严重依赖于初始类簇中心的选择、对噪声和离群点敏感、不适于发现非凸形状的类簇等问题严重限制了其应用。基于层次的典型聚类算法为BIRCH(balanced iterative reducing and clustering using hierarchies)[12],它的优点是聚类速度快,可有效识别噪音点,但该算法时间复杂度高且对高维或非凸数据的聚类效果不佳。DBSCAN(density-based spatial clustering of applications with noise)[13]是基于密度的聚类算法,该算法克服了噪声敏感的缺点,能够对任意形状的数据进行聚类,但是算法中的数量阈值和圆半径的设置对聚类结果的影响较大,若参数选取不当会导致聚类错误甚至失效,并存在对高维数据聚类效果差的问题。基于网格的STING(statistical information grid)算法[14]将样本空间分割成矩阵单元,以单元为处理对象,有效地提高了算法的效率,但其存在对不规则数据处理失效和维数灾难的问题。基于模型的典型方法是高斯混合模型(Gaussian mixture model,GMM)[15],该算法对类簇进行“软”划分,并将其以概率的形式表现,提高了算法的精度,但是每个类簇的特征均需用相应的参数来表达,从而产生大量参数,增加了模型的运行负担,降低了算法的执行效率。

Rodriguez等[16]在*Science*上提出密度峰值聚类算法(density peaks clustering,DPC),该算法基于两点假设:(1)类簇中心的局部密度相对较大,并被密度不超过它的样本包围;(2)类簇中心与密度超过它的样本的距离相对较远。与其他聚类算法相比,该算法的优点在于类簇数由密度峰值确定,能快速发现任意形状的类簇。DPC算法仍存在一些不足:(1)定义的局部密度未考虑数据内部的结构差异,当类簇间的数据密集程度差异较大时不能获得合理的聚类效果;(2)样本分配策略存在分配连带错误,当某一点分配错误后会导致后续聚类的整体错误。

针对DPC算法只考虑密度的全局结构导致中心点选取错误的问题,丁世飞等[17]利用块不相似性度量计算样本的K近邻信息,并以此给出局部密度的不相似度量方法,增强了样本的局部信息,提升了算法对密度不均匀数据以及高维数据的聚类性能。针对分配策略设计存在的不足,Xie等[18]提出基于模糊加权K近邻分配点的密度峰值聚类(robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors,FKNN-DPC)算法,该算法将样本分为核心样本和离群样本,首先根据密度峰值对样本的K近邻进行广度优先搜索实现对核心样本的分配,再利用加权K近邻技术对第一次未分配样本和离群样本进行分配,此分配策略能够有效缓解DPC算法一步分配导致的分配错误传递问题。纪霞等[19]提出相对邻域与剪枝策略优化的密度峰值聚类(relative neighborhood and pruning strategy optimized density peaks clustering algorithm,RP-DPC)算法,该算法引入相对距离,将样本距离和密度的计算缩小到相对邻域中,从而有效提升了算法的效率。上述方法虽然在一定程度上改进了DPC算法的不足,但是仍存在对凹形数据集和密度差异较大的数据聚类效果不佳的问题。赵嘉等[20]提出基于相互邻近度的密度峰值聚类(density peaks clustering based on mutual neighborhood degrees,DPC-MND)算法,该算法融入K近邻思想,以类簇中心及其*K*个近邻点建立类簇,再根据相互邻近度最高原则对其余样本点进行归类,但是该算法中的参数*K*需人为指定,降低了算法的自适应力。孙林等[21]提出基于K近邻和优化分配策略的密度峰值聚类(density peak clustering algorithm based on K-nearest neighbors and optimized allocation strategy,

DPCKS)算法，首先根据构建的中心信任度确定簇中心，并结合不同相似度给出分层的分配策略，实验结果表明该算法具有较好的聚类性能，但由于算法在初始阶段选取候选簇中心时排除了大量数据点，易将可能的聚类中心剔除，影响数据集的全局结构。

基于上述分析，提出一种融合相对局部密度和最近邻关系的密度峰值聚类(density peak clustering based on relative density and nearest neighbor relationship，RN-DPC)算法。算法的主要贡献包括：(1)在局部密度的定义中引入权重限制，提出相对局部密度的定义，根据相对局部密度寻找密度峰值，避免稀疏差异较大的数据集在选取密度峰值时出现的错误，确保中心点选择的准确性；(2)针对分配策略，结合最邻近点准则和阈值限制，提出最近邻分配策略，根据阈值条件有效避免分配的连带错误；(3)基于类内距离均值定义距离比例，提出针对未分配点的修正分配策略，提升算法对边界点聚类的准确性；(4)在合成和UCI数据集的实验表明，RN-DPC与DPC以及改进的DPC算法相比具有最优的聚类效果。

1 DPC算法及缺陷

1.1 DPC算法

DPC算法通过计算密度峰值能够快速有效地找到类簇中心并利用分配策略实施聚类。设给定数据集 $X=\{x_i\}$，DPC算法以两个概念为基础，即：(1)样本点 x_i 的局部密度 ρ_i；(2)样本点 x_i 的相对距离 δ_i。

局部密度根据截断核和高斯核可定义为两种形式，其中截断核度量形式如下：

$$\rho_i=\sum_{i\neq j}X(d_{ij}-d_c)$$

$$X(x)=\begin{cases}1,x<0\\0,x\geqslant 0\end{cases} \tag{1}$$

高斯核度量形式如下：

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

其中，d_{ij} 为样本点 x_i 与 x_j 之间的距离，d_c 为截断距离。截断核适用于样本点较少的数据集，高斯核适用于样本点相对较多的数据集。

样本点 x_i 的相对距离定义为：

$$\delta_i=\max_{i\neq j}(\delta_j) \tag{3}$$

对于密度最高的样本，相对距离定义为：

$$\delta_i=\min_{j\rho_j>\rho_i}(d_{ij}) \tag{4}$$

DPC算法的具体流程如下：

步骤1 根据式(1)~(4)计算出样本点 x_i (i=1，2，…，n)的局部密度 ρ_i 和相对距离 δ_i；

步骤2 计算样本点 x_i (i=1，2，…，n)的决策值：

$$\gamma_i=\rho_i\cdot\delta_i \tag{5}$$

步骤3 根据得到的决策值，选择较大者对应的样本点作为密度峰值(类簇中心)；

步骤4 根据高密度临近原则，即样本点属于密度比它高的最近样本点所在的类簇，完成对数据集的聚类。

1.2 DPC算法的缺陷

DPC算法虽然能快速发现任意形状数据的密度峰值(类簇中心)，但存在如下缺陷：

(1)DPC算法定义的样本局部密度未考虑数据集密度稀疏程度的不同，当类簇的密度差异较大时，DPC算法并不能获得较好的聚类效果。根据DPC算法的假设，类簇中心与密度峰值之间存在一一对应关系，局部密度和相对距离的乘积是密度峰值的数值体现，进而通过确定密度峰值可获得类簇中心，但当数据集各类簇的密集程度差异较大时，相对距离在密度峰值中的作用被削弱，这导致密度峰值的选择由局部密度起决定性作用。当存在类簇密度相差较大的情况时，密度较小类簇的密度峰值会被忽略或者难以被发现，由此产生密度峰值选取错误，将会导致聚类错误或失效。针对Jain数据集(图1)，可看出根据DPC定义的局部密度无法获取正确的类簇中心，从而导致聚类效果不理想，如图2所示。

(2)DPC算法的分配策略是将非类簇中心的样本点归为距离最近且密度超过它的样本所在的类簇中，该分配策略极易产生分配连带错误，即一旦某一个样本分配错误，会导致后续其他样本的分配错误，最终无法得到理想的聚类效果。因为当密度峰值即类簇中心确定后，其他数据点的分配完全取决于密度比它高的最近数据点，这单一条件的强制限制，导致中间任何点出现错误，都会对后续的分配造成巨大的影响。针对Pathbased数据集(图3)，DPC算法虽然获取了正确的类簇中心，但存在样本点分配错误，导致出现连带错误的问题，从而无法得到理想的聚类效果，如图4所示。

2 RN-DPC算法

DPC算法能够快速、高效地对数据集进行聚类，但其对密集程度差异较大的数据集聚类效果不佳，

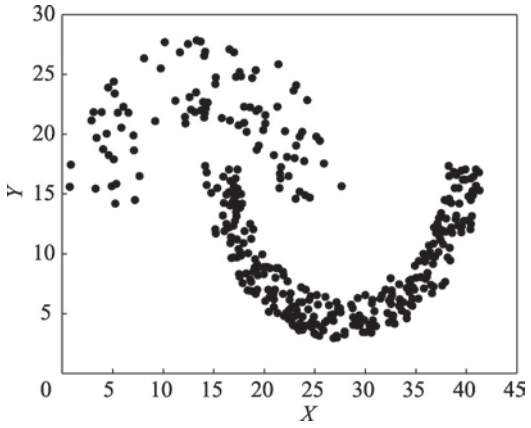

图1 Jain数据集

Fig.1 Jain dataset

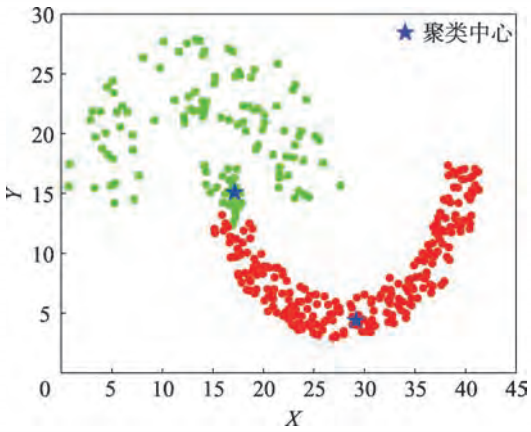

图2 DPC算法的Jain数据集聚类效果

Fig.2 Jain dataset clustering result of DPC

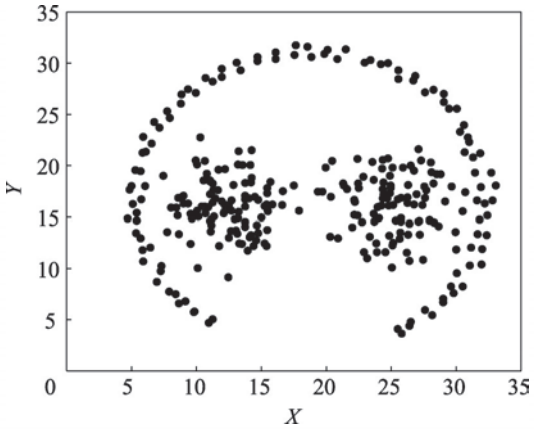

图3 Pathbased数据集

Fig.3 Pathbased dataset

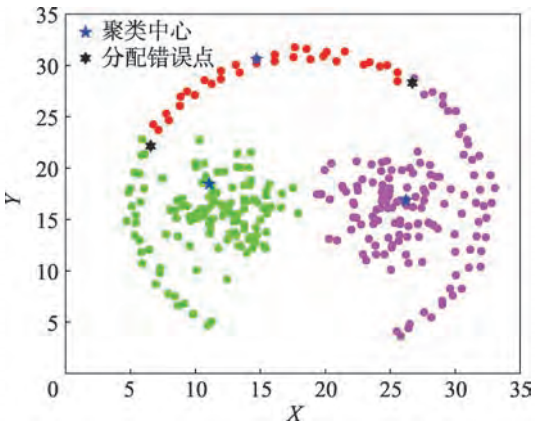

图4 DPC算法的Pathbased数据集聚类效果

Fig.4 Pathbased dataset clustering result of DPC

且分配制度存在缺陷,易发生连带错误。针对上述问题,提出融合相对密度和最近邻关系的密度峰值聚类(RN-DPC)算法。首先,提出相对局部密度的定义,在局部密度的定义中引入稀疏平和权重,将不同稀疏程度的数据赋予不同的权重,并根据相对局部密度寻找密度峰值,进而确定最优的类簇中心。然后,结合最邻近点准则和阈值限制,提出最近邻分配策略,将非类簇中心点有效地分配到相应类别。最后,提出修正分配策略,进一步优化剩余样本点的分配,从而得到最终的聚类,算法示意图如图5所示。

2.1 相对局部密度

DPC算法定义的局部密度和相对距离对于密度峰值的选取起着决定性作用,但是当样本点的局部密度相差较大时,会削弱甚至吞噬相对距离的作用。针对上述问题,提出相对局部密度,根据数据集局部密度的差异,对其赋予不同的权重调节局部密度的比重,得到更为合理的密度峰值,从而确定最优的类簇中心。根据DPC算法中定义的密度峰值,当数据集的局部密度差异较大而又存在相对距离较远的数据点时,此数据点更容易被选为密度峰值。以Jain数据集为例,根据DPC定义的密度峰值计算方法,选取的聚类中心集中在密度较大处(图6(a)),导致获取到错误的聚类中心(图7(a))。针对此种情况,应增加较小局部密度的比重,协调相对距离和局部密度在决策值选取中的作用,从而得到更为合理的密度峰值。

通过对不同数据集局部密度和相对距离的计算和分析得到,当数据点的局部密度达不到最大峰值的一半时,其作用会被削弱。因此选择以最大局部密度的一半作为增加权重的阈值,对局部密度的比重进行调节。此时定义的相对局部密度 ρ_i^* 计算如下:

$$\rho_i^*=\begin{cases}\alpha\cdot\rho_i, & \rho_i<\dfrac{\rho_{\max}}{2}\\ \rho_i, & \rho_i\geqslant\dfrac{\rho_{\max}}{2}\end{cases} \tag{6}$$

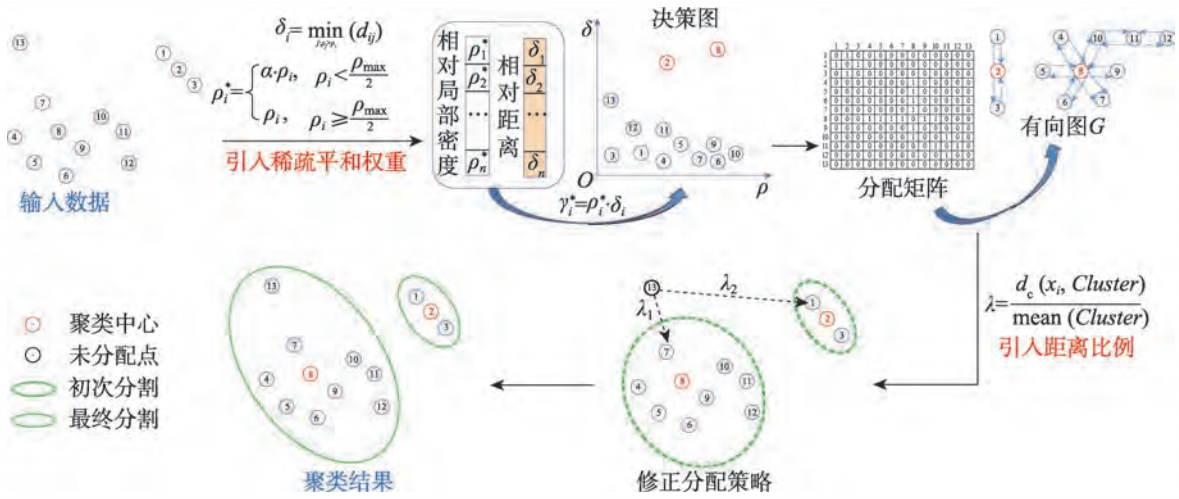

图5 RN-DPC 算法示意图

Fig.5 Illustration of RN-DPC algorithm

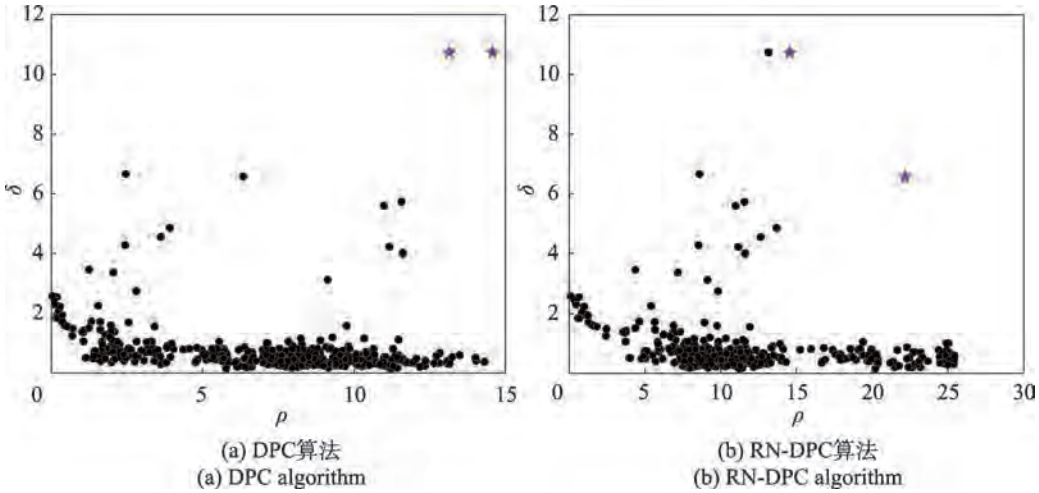

图6 两种算法对 Jain 数据集的聚类决策图

Fig.6 Clustering decision graphs for Jain dataset obtained by two methods

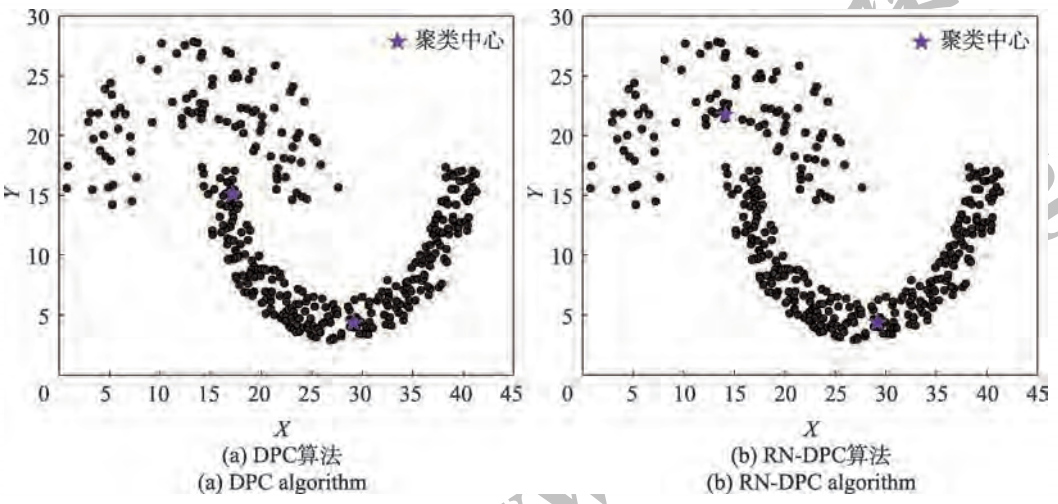

图7 两种算法在 Jain 数据集的聚类中心

Fig.7 Clustering centers for Jain dataset obtained by two methods

其中，ρ_{max} 为局部密度 ρ_i 的最大值，α 为权重参数。当局部密度差异较大时，权重系数大于1，增加局部密度的作用，而当局部密度差异不大时，权重系数接近1，保持局部密度的作用。当数据密度差异越大，局部密度较小时，其作用越容易被削弱，因此应增大其权重数值，同时正因密度差异大，局部密度较大者更集中在密度排序的前面。因此，在设置局部密度的权重时，以1/3为密度划分的节点，获得的权重更能有效提升局部密度的作用。计算公式如下：

$$\alpha=\frac{M_{1/3}}{M_{2/3}} \tag{7}$$

将局部密度按从大到小排序，得到局部密度的新排序 $\{\rho_i'|i=1,2,\cdots,n\}$，$M_{1/3}$ 为排在前1/3局部密度的均值，$M_{2/3}$ 为后2/3局部密度的均值，具体计算如下：

$$M_{1/3}=\frac{1}{\left\lceil\frac{1}{3}n\right\rceil}\sum_{i=1}^{\left\lceil\frac{1}{3}n\right\rceil}\rho_i' \tag{8}$$

$$M_{2/3}=\frac{1}{n-\left\lceil\frac{1}{3}n\right\rceil}\sum_{i=\left\lceil\frac{1}{3}n\right\rceil+1}^{n}\rho_i' \tag{9}$$

其中，$\lceil\cdot\rceil$ 为向上取整函数。

根据式(4)得到相对距离 δ_i，进一步给出决策值 γ_i^* 的定义如下：

$$\gamma_i^*=\rho_i^*\cdot\delta_i \tag{10}$$

根据上述定义得到的密度峰值如图6(b)所示，协调了相对距离 δ_i 和局部密度 ρ_i 在决策值选取中的作用，进而获取到最优的聚类中心(图7(b))。相对于DPC算法的局部密度定义，提出的相对局部密度考虑到数据集的稀疏差异，对稀疏差异较大数据集的局部密度赋予不同权重，用以平衡不同稀疏程度数据所产生的差异，进而获取更为合理的密度峰值。

2.2 分配策略

确定类簇中心后，根据分配策略将非类簇中心样本点进行归类。DPC算法的分配策略是将非密度峰值样本分配给距其最近且密度比其大的样本所在的类簇，此方法仅取决于邻近样本，极易产生分配连带错误，导致无法实现正确的聚类。根据样本最近邻原则将分配策略进行改进，同时增加距离阈值限定条件防止分配连带错误。首先，以确定的类簇中心为起始点，计算样本间的距离矩阵 $\boldsymbol{D}=[d(i,j)]$，并定义样本点 x_i 的距离阈值，如下：

$$d_t(i)=\beta\cdot d(p_{C_i}^1,C_i) \tag{11}$$

其中，β 为阈值参数，C_i 为样本点 x_i 所在类簇的聚类中心，p_x^1 表示距离样本点 x_i 最近的第一个未分配点。

然后，结合最近邻原则和距离阈值计算分配矩阵 $\boldsymbol{A}=[A(i,j)]$，如下：

$$A(i,j)=\begin{cases}1,\text{在}x_j=p_{x_i}^1\text{且}d(i,j)\leqslant d_t(i)\\ \quad\text{且}i=\arg\min_k d(x_i,C_k)\\ 0,\text{其他情况}\end{cases} \tag{12}$$

分配矩阵 $\boldsymbol{A}$ 为对称稀疏矩阵，其结合1-近邻图和共享近邻图确定有向图 $G=(V;E)$，V 为待聚类的样本点，E 为连接边，$A(i,j)=1$，图的连通部分即为确定的类簇。实施此分配策略后仍可能存在未分配样本点，接下来针对剩余未分配点提出修正策略。

2.3 修正分配策略

未分配点通常为距离所有类簇都较远的点或类簇的边界点，前者只需将其划分到最邻近类即可，后者不仅要考虑与最邻近类的关系，还需考虑其与次邻近类的关系，即未分配点与最邻近类的距离如果大于最邻近类的平均距离，而与次邻近类的距离小于次邻近类的平均距离，则未分配点不应属于最邻近类而应属于次邻近类，如图8所示。

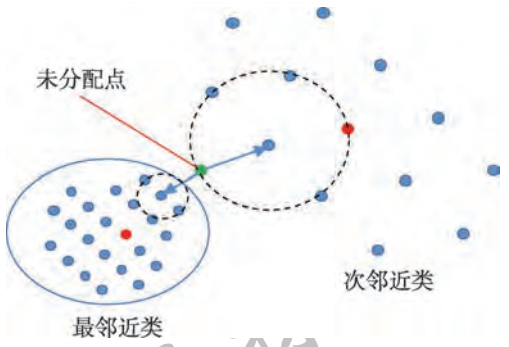

图8 未分配点距离关系

Fig.8 Unassigned point distance relationship

基于上述分析，定义未分配点 x_i 相对于类簇 *Cluster* 的距离比例 λ，即：

$$\lambda=\frac{d_c(x_i,Cluster)}{\text{mean}(Cluster)} \tag{13}$$

其中，$d_c(x_i,Cluster)$ 为未分配点到类簇 *Cluster* 的距离，mean(*Cluster*) 表示类簇 *Cluster* 的平均距离。则 λ_1 和 λ_2 分别表示 $Cluster_1$ 未分配点 x_i 相对于最邻近类和次邻近类 $Cluster_2$ 的距离比例。

最终得到未分配点 x_i 的修正分配策略为：(1)如果 $\lambda_1\leqslant\lambda_2$，则 x_i 属于最邻近类；(2)如果 $\lambda_1>\lambda_2$，则 x_i 属于次邻近类。

2.4 整体聚类算法

RN-DPC算法首先引入稀疏平和权重，提出的相对局部密度增加了较小局部密度的比重，协调了相对距离和局部密度在决策值选取中的作用，避免稀疏差异较大的数据集在选取密度峰值时出现的错误，从而得到更为合理的密度峰值。然后，提出的最近邻分配策略结合了最邻近点准则和阈值限制，准确地确定了最邻近点的类别，并将可能造成分配错误的点排除在外，有效抑制分配连带错误的发生。最后，基于类内距离均值提出修正分配策略，同时考虑数据距离与类内密度，提升了算法对边界点聚类的准确性。算法流程如下：

输入：数据集 $X=\{x_i|i=1,2,\cdots,n\}$。

输出：聚类结果 C。

步骤1 计算样本点间欧氏距离，根据式(1)~(4)和式(6)~(7)分别计算样本点 $x_i(i=1,2,\cdots,n)$ 的局部密度 ρ_i^* 和相对距离 δ_i；

步骤2 根据式(10)计算改进的决策值 γ_i^*；

步骤3 根据计算出的决策值，选择较大的样本作为密度峰值(类簇中心)；

步骤4 根据类簇中心和距离矩阵，根据式(12)计算分配矩阵，划分有向图 G 得到初次聚类结果；

步骤5 针对所有未分配点，根据式(13)计算距离比例，再由修正分配策略确定其所属类别，得到最终的聚类结果。

2.5 算法复杂度分析

DPC算法的时间复杂度主要由计算样本间距离矩阵的复杂度、计算样本局部密度的复杂度和计算样本相对距离的复杂度组成。每个部分的时间复杂度均为 $O(n^2)$，因此总的时间复杂度为 $O(n^2)$。RN-DPC算法的时间复杂度主要由以下五部分组成：(1)计算样本间距离矩阵的复杂度 $O(n^2)$；(2)计算每个样本相对局部密度的复杂度 $O(n^2)$；(3)计算样本相对距离的复杂度 $O(n^2)$；(4)第一次分配的时间复杂度为 $O(n^2)$；(5)修正分配策略时假设剩余未分配点个数为 $m,m<n$，时间复杂度为 $O(m^2)<O(n^2)$，则RN-DPC算法的时间复杂度为 $O(n^2)$。因此，RN-DPC算法与DPC算法具有相同的时间复杂度。

3 实验结果与分析

实验基于合成数据集和UCI数据集[22]实施算法的测试与分析，将提出的RN-DPC算法与FKNN-DPC[18]、DPC[16]、DBSCAN[13]、*K*-means[11]、OPTICS[23]和AP(affinity propagation)[24]算法进行比较。DPC、DBSCAN和*K*-means算法的实验结果基于作者提供的源代码在Matlab2019a上获得；AP算法的实验结果基于Python的Sklearn库获得；OPTICS算法的实验结果基于Python的PyClustering库获得。

3.1 评价指标

实验使用3个评价指标评估聚类效果，分别为调整互信息(adjusted mutual information，AMI)[25]、调整兰德系数(adjusted Rand index，ARI)[25]和Fowlkes-Mallows指数(Fowlkes-Mallows index，FMI)[26]。以上指标的最大取值均为1，指标的数值越大，意味着聚类结果越好。

AMI用来衡量数据分布的重叠程度。假设 U 和 V 分别为 N 个数据的真实和实验类标，二者的熵如下：

$$Ent(U)=\sum_{i=1}^{|U|}P(i)\log P(i) \tag{14}$$

$$Ent(V)=\sum_{i=1}^{|V|}P'(j)\log P'(j) \tag{15}$$

其中，$|\cdot|$ 表示样本数量，$P(i)=|U_i|/N, P(j)=|V_j|/N$。$U$ 和 V 之间的互信息为：

$$\mathrm{MI}(U,V)=\sum_{i=1}^{|U|}\sum_{j=1}^{|V|}P(i,j)\log\frac{P(i,j)}{P(i)P'(j)} \tag{16}$$

调整互信息AMI的定义为：

$$AMI=\frac{\mathrm{MI}(U,V)}{\sqrt{Ent(U)Ent(V)}} \tag{17}$$

ARI通过统计正确决策对的数量来评价算法的性能，其定义如下：

$$ARI=\frac{2(A\cdot D-B\cdot C)}{(B+D)\cdot(A+C)+(C+D)\cdot(A+B)} \tag{18}$$

FMI通过计算成对准确率与召回率的几何平均值来衡量聚类性能，定义为：

$$FMI=\frac{A}{\sqrt{(A+B)\cdot(A+C)}} \tag{19}$$

其中，A 和 D 分别表示在 U 和 V 均为同一类和不同类的样本点对数量，B 和 C 分别表示在 U 或 V 中同类而在另一类不同类的样本点对数量。

3.2 数据集介绍

实验选用的合成数据集和UCI数据集被广泛应用于聚类算法有效性的测试。这些数据集的特点各异，数据集的样本规模、属性个数、类簇个数如表1和表2所示。

3.3 合成数据集实验结果分析

首先，将提出的RN-DPC算法与3个著名的聚类

表1 合成数据集

Table 1 Synthetic dataset

数据集	样本规模	属性个数	类簇个数
Aggregation	788	2	7
Flame	240	2	2
Jain	373	2	2
Pathbased	300	2	3
Spiral	312	2	3

表2 UCI数据集

Table 2 UCI dataset

数据集	样本规模	属性个数	类簇个数
Iris	150	4	3
Seeds	210	7	3
Libras	360	90	15
Ionosphere	351	34	2
Balance Scale	625	4	3

算法DPC、DBSCAN和*K*-means算法在合成数据集上实施验证，并将聚类结果可视化显示在图9~图13中，图中不同颜色的点代表不同的类簇，DBSCAN算法中的噪声点用“×”代表。图9~图13表明，RN-DPC算法在聚类效果上均优于其他算法。从图9可以看出，Jain数据集由于两个月牙形类簇的密集程度相差较大，并存在交叠部分，使得DPC、DBSCAN和*K*-means算法均无法实现准确的聚类，而由于RN-DPC算法引入稀疏平和权重，从而使其能够获得正确的密度峰值和聚类结果。从图10可以看出，Pathbased数据集中圆环形状数据与其中两个圆形数据距离较近，使得DPC、DBSCAN和*K*-means算法均产生不同程度的分配连带错误，而由于RN-DPC算法在分配策略中增加了阈值限制，有效地避免了连带错误的发生，从而获得正确的聚类效果。从图11可以看出，Aggregation数据集由形状不同的7个类簇组成，其中有两个部分均存在类簇连接现象，这导致*K*-means算法出现严重的分割错误，而RN-DPC、DPC和DBSCAN算法均考虑到密度因素，使其获得理想的聚类效果。从图12可以看出，Spiral数据集由3组螺旋数据组成，RN-DPC、DPC和DBSCAN算法中均融入数据邻接关系，使其获得合理的聚类结果，而*K*-means算法仅考虑数据之间的欧式距离，导致其出现聚类错误。从图13可以看出，Flame数据集出现明显的数据连接现象，这导致DBSCAN和*K*-means算法均无法实现准确的聚类，而RN-DPC和DPC均获得准确的聚类结果。

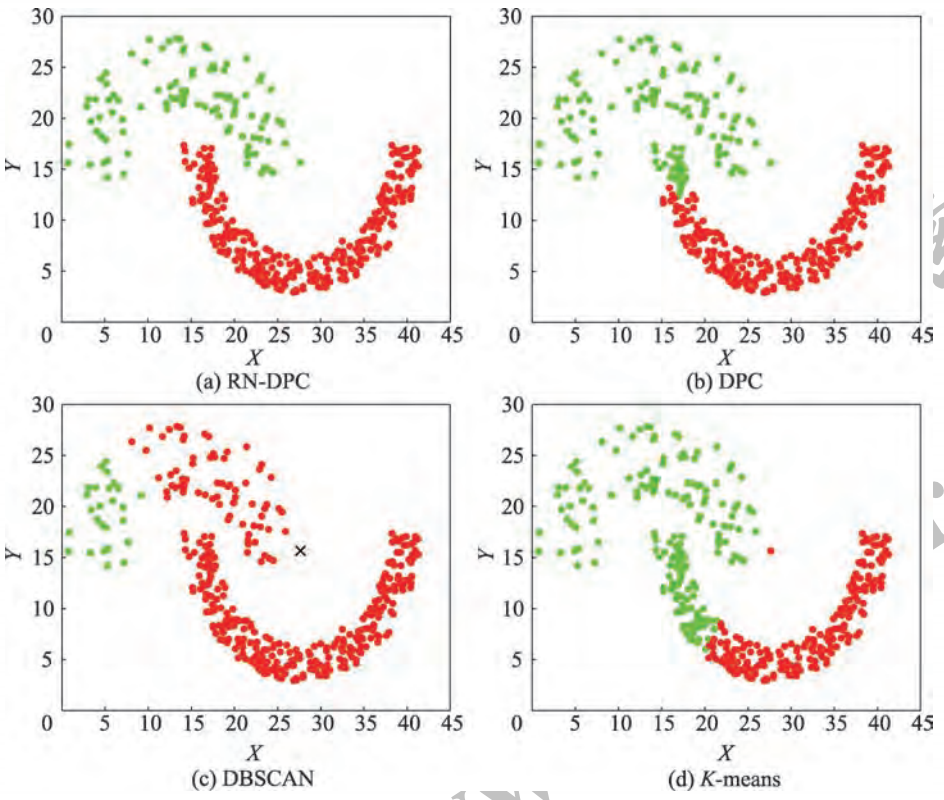

图9 4种算法对Jain数据集的聚类结果

Fig.9 Clustering results of Jain dataset by 4 methods

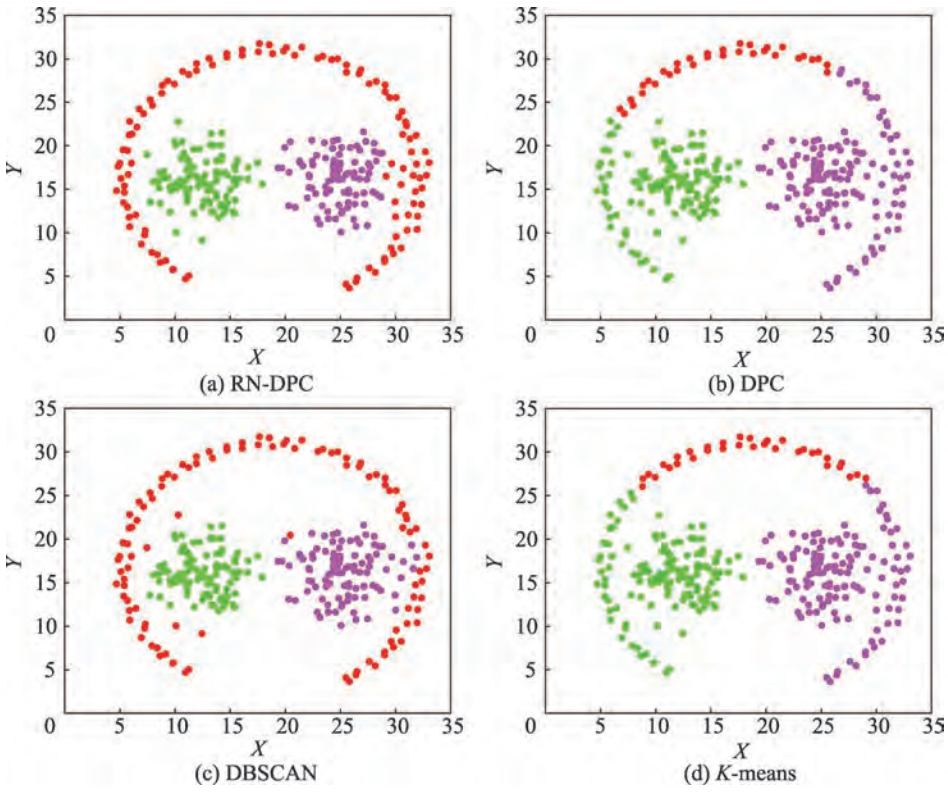

图 10　4 种算法对 Pathbased 数据集的聚类结果

Fig.10　Clustering results of Pathbased dataset by 4 methods

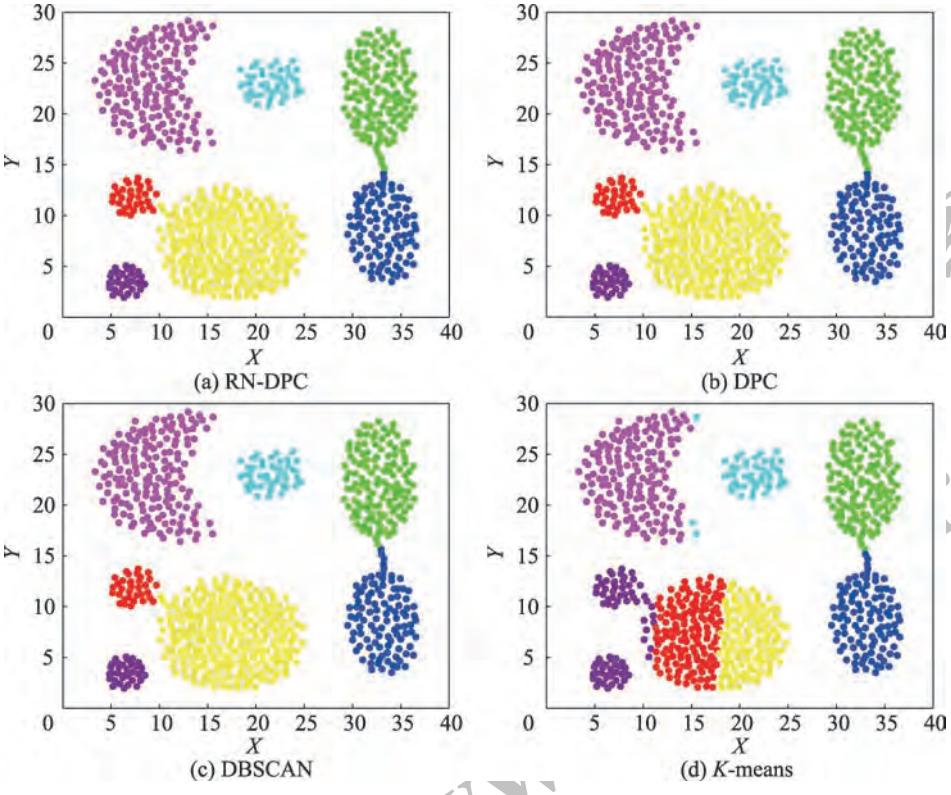

图 11　4 种算法对 Aggregation 数据集的聚类结果

Fig.11　Clustering results of Aggregation dataset by 4 methods

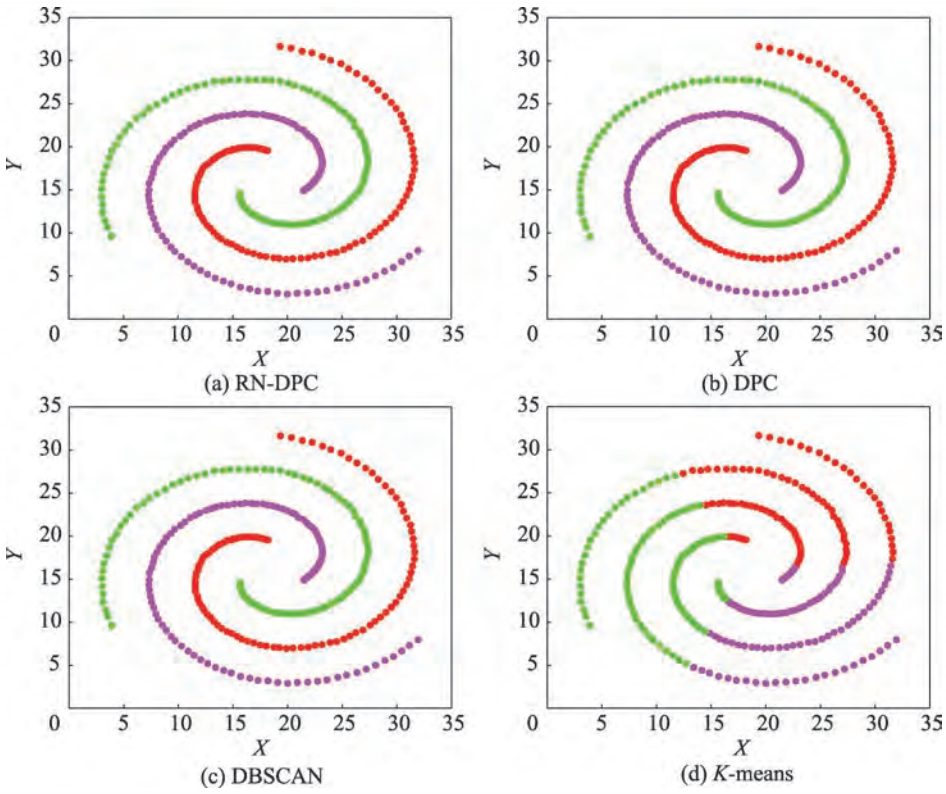

图12 4种算法对Spiral数据集的聚类结果

Fig.12 Clustering results of Spiral dataset by 4 methods

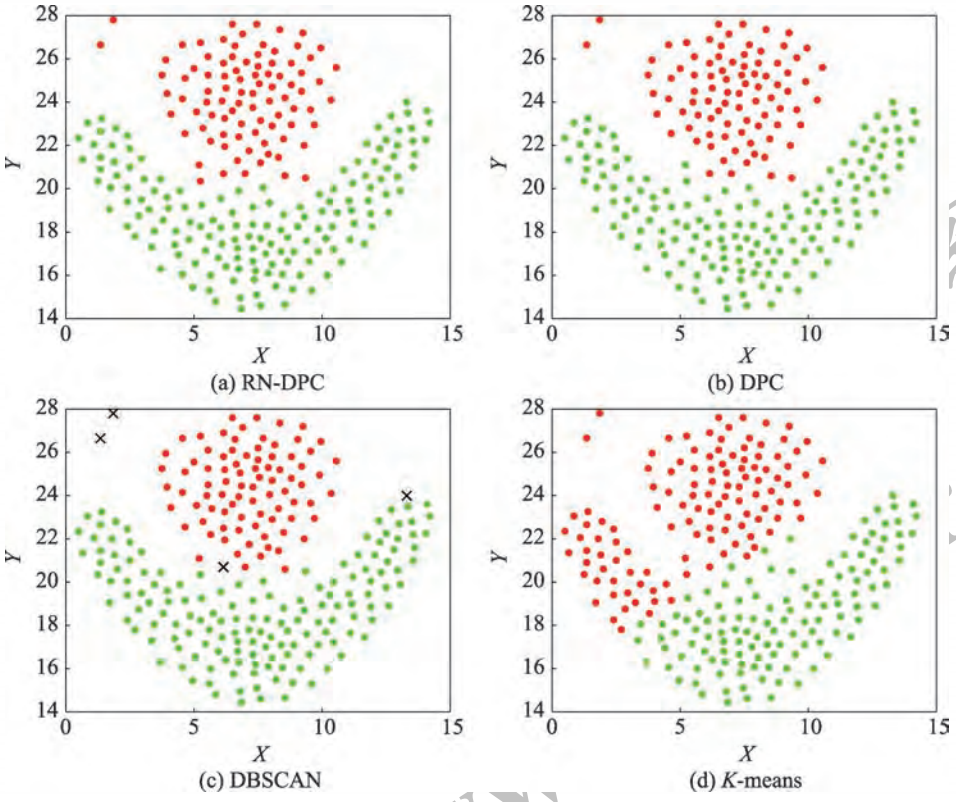

图13 4种算法对Flame数据集的聚类结果

Fig.13 Clustering results of Flame dataset by 4 methods

然后，为验证RN-DPC算法的优越性，将其与DPC-MND、FKNN-DPC、DPC、DBSCAN、OPTICS、AP、*K*-means算法针对5个合成数据集进行比较，并利用评价指标AMI、ARI和FMI评估聚类性能，如表3所示。在表3中，Arg-表示算法获得最优结果时对应的参数取值，其中RN-DPC所用的距离阈值在选取时，以距离中心点最近数据点的距离为最小取值，逐渐增长其倍数作为参数取值进行实验训练，最大取值则为任一中心点到数据点最远距离的最大值。对所得实验结果进行比较，将最佳实验结果的取值进行取整后作为最终的距离阈值。表3表明，针对Pathbased数据集提出的分配策略避免了连带错误问题，获得最优的聚类结果；针对Jain、Spiral和Aggregation数据集，RN-DPC算法根据提出的相对局部密度获取到合理的聚类中心，并采用提出的最邻近点分配和修正分配策略，使其得到最优的聚类结果；针对Flame数据集，RN-DPC算法的聚类效果仅次于DPC算法和FKNN-DPC算法，优于其他算法。

最后，从统计角度验证RN-DPC算法的优越性，实验引入显著性差异检验方法——Friedman检验，其秩均值能够体现算法的综合性能，获得的秩均值越大，表明对应算法的综合性能越好。分别对各算法在合成数据集的AMI、ARI、FMI评价指标进行秩均值检验，如表4所示。表4表明，RN-DPC算法在3种聚类评价指标上均获得最大的秩均值。因此，RN-DPC算法与其他算法相比具有最优的综合性能。

表4　3种评价指标在合成数据集的Friedman检验值

Table 4　Friedman values of 3 indexes on synthetic datasets

聚类算法	秩平均值		
	AMI	ARI	FMI
RN-DPC	**6.90**	**6.90**	**6.90**
DPC-MND	6.20	6.20	6.20
FKNN-DPC	4.80	4.80	4.80
DPC	5.50	5.10	5.30
DBSCAN	5.10	5.10	5.10
OPTICS	3.50	4.10	4.10
AP	2.40	2.60	2.20
K-means	1.60	1.20	1.40

3.4　UCI数据集实验结果分析

为进一步验证RN-DPC算法的性能，将其与DPC-MND、FKNN-DPC、DPC、DBSCAN、OPTICS、AP、*K*-means算法针对UCI数据集中的5个数据集进行比较，并利用评价指标AMI、ARI和FMI评估聚类性能，如表5所示。表5表明，RN-DPC算法在Iris、

表3　8种聚类算法在合成数据集上的性能

Table 3　Performance of 8 clustering methods on synthetic datasets

聚类算法	Aggregation				Flame				Jain			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
RN-DPC	**1.000 0**	**1.000 0**	**1.000 0**	3	0.935 9	0.966 7	0.985 6	4	**1.000 0**	**1.000 0**	**1.000 0**	2
DPC-MND	0.995 5	0.997 8	0.998 3	28	0.931 8	0.966 6	0.984 6	43	**1.000 0**	**1.000 0**	**1.000 0**	5
FKNN-DPC	0.977 5	0.985 5	0.988 6	20	**1.000 0**	**1.000 0**	**1.000 0**	6	0.056 2	0.131 8	0.643 0	10
DPC	**1.000 0**	**1.000 0**	**1.000 0**	3.4	**1.000 0**	**1.000 0**	**1.000 0**	2.8	0.618 3	0.714 6	0.881 9	0.9
DBSCAN	0.952 9	0.977 9	0.982 7	0.04/6	0.823 4	0.938 8	0.971 2	0.09/8	0.865 0	0.975 8	0.990 6	0.08/2
OPTICS	0.922 1	0.975 3	0.980 7	0.06/10	0.689 8	0.896 8	0.950 8	0.10/8	0.854 2	0.975 6	0.990 5	0.08/1
AP	0.787 3	0.765 8	0.815 0	1.21	0.498 7	0.540 3	0.749 8	6.36	0.658 2	0.795 2	0.921 2	−1.77
K-means	0.793 5	0.730 0	0.788 4	7	0.386 3	0.453 4	0.736 4	2	0.491 6	0.576 7	0.820 0	2

聚类算法	Pathbased				Spiral			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
RN-DPC	**0.965 8**	**0.979 5**	**0.985 5**	3	**1.000 0**	**1.000 0**	**1.000 0**	5
DPC-MND	0.952 4	0.969 6	0.979 7	10	**1.000 0**	**1.000 0**	**1.000 0**	1
FKNN-DPC	0.834 4	0.874 4	0.916 5	9	**1.000 0**	**1.000 0**	**1.000 0**	5
DPC	0.521 2	0.471 7	0.666 4	3.8	**1.000 0**	**1.000 0**	**1.000 0**	1.8
DBSCAN	0.871 0	0.901 1	0.934 0	0.08/10	**1.000 0**	**1.000 0**	**1.000 0**	0.04/2
OPTICS	0.436 4	0.636 4	0.751 7	0.06/4	**1.000 0**	**1.000 0**	**1.000 0**	0.04/1
AP	0.519 9	0.477 5	0.657 7	−4.1	0.293 2	0.156 9	0.340 9	−0.19
K-means	0.509 8	0.461 3	0.661 7	3	−0.006 0	−0.006 0	0.327 4	3

表5 8种聚类算法在UCI数据集上的性能

Table 5 Performance of 8 clustering methods on UCI datasets

聚类算法	Iris				Balance Scale				Ionoshpere			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
RN-DPC	**0.940 5**	**0.960 3**	**0.939 8**	5	**0.161 9**	**0.264 1**	**0.557 9**	3	0.488 0	0.507 7	0.769 9	3
DPC-MND	0.878 6	0.903 7	0.935 4	22	0.142 5	0.198 5	0.514 1	35	0.402 1	0.527 2	0.797 7	44
FKNN-DPC	0.883 1	0.903 8	0.935 5	22	0.035 1	0.023 6	0.554 8	9	0.131 4	0.132 1	0.584 1	26
DPC	0.860 6	0.885 7	0.923 3	0.2	0.115 4	0.139 4	0.502 4	1.1	0.150 4	0.235 7	0.649 1	0.5
DBSCAN	0.569 2	0.612 0	0.729 1	0.12/5	0.090 2	0.139 4	0.151 0	0.03/1	**0.594 7**	**0.722 6**	**0.874 0**	0.78/9
OPTICS	0.451 3	0.688 6	0.786 8	0.15/5	0.063 3	0.106 2	0.116 5	0.03/2	0.097 0	0.338 3	0.608 5	0.58/1
AP	0.547 9	0.570 1	0.709 9	0.23	0.090 2	0.142 0	0.155 3	0.97	0.136 7	0.077 3	0.513 7	1.92
K-means	0.733 1	0.716 3	0.811 2	2	0.013 2	0.001 5	0.044 0	3	0.129 4	0.177 6	0.605 3	2

聚类算法	Libras				Seeds			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
RN-DPC	**0.575 2**	**0.376 0**	**0.516 3**	2	0.700 8	0.766 5	0.696 5	3
DPC-MND	0.564 6	0.373 6	0.432 8	16	**0.756 6**	**0.801 1**	**0.867 0**	7
FKNN-DPC	0.475 4	0.318 4	0.397 6	11	0.697 1	0.742 2	0.827 6	9
DPC	0.535 8	0.319 3	0.371 7	0.3	0.729 9	0.767 0	0.844 4	0.7
DBSCAN	0.418 3	0.196 5	0.257 0	0.90/2	0.530 2	0.529 1	0.671 1	0.24/16
OPTICS	0.137 7	0.082 8	0.212 6	0.59/1	0.380 2	0.419 0	0.635 0	0.81/5
AP	0.148 7	0.205 6	0.197 1	4.31	0.446 5	0.393 6	0.693 3	−2.07
K-means	0.523 2	0.309 4	0.361 2	15	0.670 5	0.704 9	0.802 6	3

Libras和Balance Scale数据集上均获取到正确的聚类中心,并且分配时策略合理,使其得到最优的聚类效果;在Ionoshpere和Seeds数据集上,RN-DPC算法虽然获取到正确的聚类中心,但数据点交叉紧密,导致分配时产生部分数据点的分配错误,聚类效果不佳,此问题将作为本研究处理复杂数据的改进方向和未来的研究目标。UCI数据集上针对3种聚类评价指标Friedman检验值如表6所示。由表6可知,RN-DPC算法在3种聚类评价指标上的秩均值均为最大值,再次验证RN-DPC算法与其他算法相比具有最优的聚类性能。

表6 3种评价指标在UCI数据集的Friedman检验值

Table 6 Friedman values of 3 indexes on UCI datasets

聚类算法	秩平均值		
	AMI	ARI	FMI
RN-DPC	**7.40**	**6.92**	**6.80**
DPC-MND	6.80	6.75	6.80
FKNN-DPC	4.20	4.42	5.60
DPC	5.80	5.33	5.40
DBSCAN	4.30	4.17	3.60
OPTICS	1.40	3.25	2.40
AP	2.90	2.33	2.00
K-means	3.20	2.83	3.40

3.5 消融实验

为进一步验证所提创新点的有效性,在6个数据集上对提出的算法进行消融实验。DPC+LC表示DPC与提出的相对局部密度融合的算法,DPC+AS表示DPC与提出的分配策略融合的聚类算法,消融实验结果如表7所示。表7表明,在Jain和Libras数据集上,DPC+LC的聚类效果优于DPC,在Pathbased、Spiral、Aggregation和Iris数据集上,DPC+LC的聚类效果和DPC算法相同,因此DPC+LC的聚类性能优于DPC算法,验证了提出的相对局部密度的有效性。在Jain、Pathbased和Libras数据集上,DPC+AS的聚类效果优于DPC,表明了提出的分配策略的有效性。在Spiral、Aggregation和Iris数据集上,由于未加修正分配,存在未分配点,导致DPC+AS的聚类效果略低于DPC。由于DPC算法在数据集分配时不存在未分配点,修正策略无法与DPC算法进行融合。通过RN-DPC算法和未加修正分配的DPC+LC+AS算法在6个数据集上的结果对比,表明带有分配策略的算法具有更好的聚类性能,验证了提出的分配策略的有效性。

表7 RN-DPC算法在6个数据集上的消融实验结果

Table 7 Experimental results of RN-DPC ablation on 6 datasets

聚类算法	Jain			Pathbased			Spiral		
	AMI	ARI	FMI	AMI	ARI	FMI	AMI	ARI	FMI
DPC	0.618 3	0.714 6	0.881 9	0.521 2	0.471 7	0.666 4	**1.000 0**	**1.000 0**	**1.000 0**
DPC+LC	0.772 3	0.773 8	0.892 6	0.521 2	0.471 7	0.666 4	**1.000 0**	**1.000 0**	**1.000 0**
DPC+AS	0.663 2	0.763 2	0.813 7	0.857 2	0.865 3	0.909 4	0.956 6	0.967 4	0.987 5
DPC+LC+AS	0.750 5	0.796 1	0.838 3	0.857 2	0.865 3	0.909 4	0.956 6	0.967 4	0.987 5
RN-DPC	**1.000 0**	**1.000 0**	**1.000 0**	**0.965 8**	**0.979 5**	**0.985 5**	**1.000 0**	**1.000 0**	**1.000 0**
聚类算法	Aggregation			Iris			Libras		
	AMI	ARI	FMI	AMI	ARI	FMI	AMI	ARI	FMI
DPC	**1.000 0**	**1.000 0**	**1.000 0**	0.863 6	0.887 5	0.923 3	0.115 4	0.139 4	0.502 4
DPC+LC	**1.000 0**	**1.000 0**	**1.000 0**	0.863 6	0.887 5	0.923 3	0.129 1	0.206 2	0.416 6
DPC+AS	0.959 4	0.970 8	0.977 2	0.762 6	0.804 1	0.866 2	0.121 1	0.099 4	0.324 5
DPC+LC+AS	0.959 4	0.970 8	0.977 2	0.762 6	0.804 1	0.866 2	0.192 3	0.150 1	0.363 8
RN-DPC	**1.000 0**	**1.000 0**	**1.000 0**	**0.940 5**	**0.960 3**	**0.939 8**	**0.161 9**	**0.264 1**	**0.557 9**

4 结束语

针对DPC算法存在的不足，提出融合相对局部密度和最近邻关系的密度峰值聚类，该算法在类簇中心选择时引入相对局部密度，避免了DPC算法在处理稀疏程度差异较大数据集时无法选出正确的聚类中心导致效果不佳的问题；在分配时结合最邻近点准则和阈值限制，提出最近邻分配和修正分配策略，通过设定阈值条件约束数据点的分配，避免了DPC算法中一次分配连带错误，并利用修正分配避免由于密度不均导致的分配错误问题。通过对多个合成和UCI数据集的实验表明，RN-DPC算法可以较好地处理各种形状的数据集，且算法的聚类性能优异。在处理复杂数据集时，面对数据点密集交叉程度严重的数据集仍无法实现准确的分配，并且距离阈值的设定上需要有足够的经验，因此如何更好地处理复杂和高维数据集及距离阈值的自适应确定成为本研究的改进方向。

参考文献：

[1] LI X, ZHANG H, WANG R, et al. Multiview clustering: a scalable and parameter-free bipartite graph fusion method [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(1): 330-344.

[2] 徐金东, 赵甜雨, 冯国政, 等. 基于上下文模糊C均值聚类的图像分割算法[J]. 电子与信息学报, 2021, 43(7): 2079-2086.
XU J D, ZHAO T Y, FENG G Z, et al. Image segmentation algorithm based on context fuzzy C-means clustering[J]. Journal of Electronics & Information Technology, 2021, 43(7): 2079-2086.

[3] 邢海燕, 刘超, 徐成, 等. 基于粒子群优化模糊C焊缝等级磁记忆定量识别模型[J]. 吉林大学学报(工学版), 2022, 52(3): 525-532.
XING H Y, LIU C, XU C, et al. Quantitative metal magnetic memory classification model of weld grades based on particle swarm optimization fuzzy C-means[J]. Journal of Jilin University (Engineering and Technology Edition), 2022, 52(3): 525-532.

[4] CHEN H, LIANG M, LIU W, et al. An approach to boundary detection for 3D point clouds based on DBSCAN clustering[J]. Pattern Recognition, 2022, 124: 108431.

[5] 王芙银, 张德生, 张晓. 结合鲸鱼优化算法的自适应密度峰值聚类算法[J]. 计算机工程与应用, 2021, 57(3): 94-102.
WANG F Y, ZHANG D S, ZHANG X. Adaptive density peak clustering algorithm combining whale optimization algorithm[J]. Computer Engineering and Applications, 2021, 57(3): 94-102.

[6] LIU N, XU Z, ZENG X J, et al. An agglomerative hierarchical clustering algorithm for linear ordinal rankings[J]. Information Sciences, 2021, 557: 170-193.

[7] XU T, JIANG J. A graph adaptive density peaks clustering algorithm for automatic centroid selection and effective aggregation[J]. Expert Systems with Applications, 2022, 195: 116539.

[8] 彭启慧, 宣士斌, 高卿. 分布的自动阈值密度峰值聚类算法[J]. 计算机工程与应用, 2021, 57(5): 71-78.
PENG Q H, XUAN S B, GAO Q. Distribution automatic threshold density peak clustering algorithm[J]. Computer Engineering and Applications, 2021, 57(5): 71-78.

[9] MELNYKOV V, SARKAR S, MELNYKOV Y. On finite

mixture modeling and model-based clustering of directed weighted multilayer networks[J]. Pattern Recognition, 2020, 112: 107641.

[10] REZAEE M J, ESHKEVARI M, SABERI M, et al. GBK-means clustering algorithm: an improvement to the K-means algorithm based on the bargaining game[J]. Knowledge-Based Systems, 2021, 213: 106672.

[11] LIKAS A, VLASSIS N, VERBEEK J J. The global k-means clustering algorithm[J]. Pattern Recognition, 2003, 36(2): 451-461.

[12] ZHANG T, RAMAKRISHNAN R, LIVNY M. BIRCH: an efficient data clustering method for very large databases[J]. ACM SIGMOD Record, 1996, 25(2): 103-114.

[13] SCHUBERT E, SANDER J, ESTER M, et al. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN [J]. ACM Transactions on Database Systems, 2017, 42(3): 19.

[14] BUREVA V, SOTIROVA E, POPOV S, et al. Generalized net of cluster analysis process using STING: a statistical information grid approach to spatial data mining[C]//Proceedings of the 12th International Conference on Flexible Query Answering Systems, London, Jun 21-22, 2017. Cham: Springer, 2017: 239-248.

[15] ANDRIYANOV N, TASHLINSKY A, DEMENTIEV V. Detailed clustering based on Gaussian mixture models[C]// Proceedings of the 2020 Intelligent Systems Conference, London, Sep 3-4, 2020. Cham: Springer, 2020: 437-448.

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

[17] 丁世飞, 徐晓, 王艳茹. 基于不相似性度量优化的密度峰值聚类算法[J]. 软件学报, 2020, 31(11): 3321-3333.
DING S F, XU X, WANG Y R. Optimized density peaks clustering algorithm based on dissimilarity measure[J]. Journal of Software, 2020, 31(11): 3321-3333.

[18] XIE J, GAO H, XIE W, 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.

[19] 纪霞, 姚晟, 赵鹏. 相对邻域与剪枝策略优化的密度峰值聚类算法[J]. 自动化学报, 2020, 46(3): 562-575.
JI X, YAO C, ZHAO P. Relative neighborhood and pruning strategy optimized density peaks clustering algorithm[J]. Acta Automatica Sinica, 2020, 46(3): 562-575.

[20] 赵嘉, 姚占峰, 吕莉, 等. 基于相互邻近度的密度峰值聚类算法[J]. 控制与决策, 2021, 36(3): 543-552.
ZHAO J, YAO Z F, LV L, et al. Density peaks clustering based on mutual neighbor degree[J]. Control and Decision, 2021, 36(3): 543-552.

[21] 孙林, 秦小营, 徐久成, 等.基于K近邻和优化分配策略的密度峰值聚类算法[J]. 软件学报, 2022, 33(4): 1390-1411.
SUN L, QIN X Y, XU J C, et al. Density peak clustering algorithm based on K-nearest neighbors and optimized allocation strategy[J]. Journal of Software, 2022, 33(4): 1390-1411.

[22] BLAKE C L, MERZ C J. UCI repository of machine learning database[EB/OL]. (2016-12-28) [2022-04-20]. http://archive.ics.uci.edu/ml/index.php.

[23] ANKERST M, BREUNIG M M, KRIEGEL H P, et al. OPTICS: ordering points to identify the clustering structure [J]. ACM SIGMOD Record, 1999, 28(2): 49-60.

[24] FREY B J, DUECK D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976.

[25] VINH N X, EPPS J, BAILEY J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance[J]. The Journal of Machine Learning Research, 2010, 11(1): 2837-2854.

[26] FOWLKES E B, MALLOWS C L. A method for comparing two hierarchical clusterings[J]. Journal of the American Statistical Association, 1983, 78(383): 553-569.

王威娜(1981—),女,吉林人,博士,副教授,主要研究方向为数据挖掘、人工智能。
WANG Weina, born in 1981, Ph.D., associate professor. Her research interests include data mining and artificial intelligence.

朱钰(1997—),男,山东枣庄人,硕士研究生,主要研究方向为数据挖掘、智能计算。
ZHU Yu, born in 1997, M.S. candidate. His research interests include data mining and intelligent computing.

任艳(1981—),女,辽宁沈阳人,博士,副教授,主要研究方向为数据挖掘、模式识别。
REN Yan, born in 1981, Ph.D., associate professor. Her research interests include data mining and pattern recognition.