

# 基于麻雀搜索算法改进的密度峰值聚类算法

何婷霏\*, 李 秦

兰州交通大学数理学院, 甘肃 兰州

收稿日期: 2022年9月14日; 录用日期: 2022年10月13日; 发布日期: 2022年10月20日

## 摘 要

针对密度峰值聚类算法(Density Peaks Clustering Algorithm, DPC)用传统距离度量方式不能很好地反映数据分布, 人为选取截断距离参数主观性较强等问题, 设计了一种基于麻雀搜索算法改进的密度峰值聚类算法(Improved Density Peak Clustering Algorithm Based on Sparrow Search Algorithm, SSA-DPC)。该算法从两个方面进行改进: 改变数据间的距离度量方式, 用标准欧氏距离替代原算法中的欧氏距离; 利用麻雀搜索算法(Sparrow Search Algorithm, SSA)较强的全局寻优能力, 搜寻最佳截断距离值。通过对7个数据集进行仿真测试, 证明SSA-DPC算法在3个评价指标上均优于其他聚类算法, 提升了聚类性能, 说明了算法的有效性。

## 关键词

密度峰值聚类算法, 麻雀搜索算法, 截断距离, 标准欧氏距离

# Improved Density Peak Clustering Algorithm Based on Sparrow Search Algorithm

Ting'ai He\*, Qin Li

College of Mathematics and Physics, Lanzhou Jiaotong University, Lanzhou Gansu

Received: Sep. 14<sup>th</sup>, 2022; accepted: Oct. 13<sup>th</sup>, 2022; published: Oct. 20<sup>th</sup>, 2022

## Abstract

Aiming at the problems that density peaks clustering algorithm (DPC) cannot well reflect the data distribution with traditional distance measurement, and the artificial selection of truncation distance parameters is highly subjective, an improved density peak based on sparrow search algorithm was designed—Clustering algorithm (Improved Density Peak Clustering Algorithm Based on

\*通讯作者。

**Sparrow Search Algorithm, SSA-DPC). The algorithm is improved from two aspects: change the distance measurement method between data, and replace the Euclidean distance in the original algorithm with the standard Euclidean distance; using the strong global optimization ability of the Sparrow Search Algorithm (SSA), the best cutoff distance value was searched. Through the simulation test of 7 data sets, it is proved that the SSA-DPC algorithm is superior to other clustering algorithms in 3 evaluation indicators, and the clustering performance is improved, which shows the effectiveness of the algorithm.**

## Keywords

**Density Peak Clustering Algorithm, Sparrow Search Algorithm, Truncation Distance, Standard Euclidean Distance**

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

当前世界正处在一个科学技术飞速发展, 信息量指数型增加的状态, 与之而来的是繁多且复杂的数据, 这些数据包括人们的日常活动信息, 比如文字数据, 视频数据, 语音数据等, 如何处理分析这些数据成为当下研究的热点。聚类作为数据挖掘的核心技术之一, 其目的是将待处理的数据按照相似性规则聚成几组各异的簇, 这些簇的特征是簇内相似度高但簇间相似性低。根据原理的不同可将聚类分为五类 [1] [2] [3] [4] [5]: 划分聚类 (division-based clustering)、层次聚类 (hierarchical clustering)、网格聚类 (grid-based clustering)、密度聚类 (density-based clustering) 和模型聚类 (model-based clustering)。

密度峰值聚类算法于 2014 年被意大利学者 Rodriguez 等人 [6] 首次提出, 相比于其他聚类算法, 该算法原理简单, 参数唯一且聚类高效, 这些优势让 DPC 算法成功应用到各个领域, 包括人工智能、模式识别、图像处理 [7] 等多个方面。自此算法提出至今, 诸多国内外学者对其进行了改进: 蒋礼青等人 [8] 针对截断参数人为确定, 一个簇中存在多密度峰值数据等问题, 利用近邻距离曲线的变化和类合并优化的方法, 实现了参数自适应确定和多密度峰值数据的正确聚类。Jian 等人 [9] 利用 K 近邻思想重新定义了局部密度的计算, 并分析了 DPC 算法中核密度估计方法对聚类结果的影响, 通过使用距离归一化原理设计了新的聚类中心选择标准。这些改进算法在一定程度上解决了 DPC 算法的部分问题。

麻雀搜索算法是 2020 年由 Xue 等人 [10] 提出的一种新型群智能优化算法, 此算法被提出的契机来自于麻雀群体的觅食与反捕食行为。SSA 的数学模型可看作是发现者 - 加入者模型, 并且加入了预警机制。此算法不仅具备良好的鲁棒性和收敛速度, 而且有着较强的全局探索能力和局部开发能力, 这使得麻雀群体会自发向全局最优值移动, 有效避免了早熟现象。

本文对于人工选取截断距离参数主观性强这一问题, 考虑将密度峰值聚类算法与麻雀搜索算法相结合, 提出基于麻雀搜索算法改进的密度峰值聚类算法: 首先更改数据间距离度量方式, 用标准欧氏距离代替传统的欧氏距离来计算数据样本间的相似性, 其次选择经过变形的聚类评价指标作为 SSA 算法的适应度函数对 DPC 算法中的截断距离参数进行迭代寻优, 最后对所提算法进行仿真实验, 分别选取人工数据集和真实数据集对其进行算法有效性评价。第二部分对 DPC 算法和 SSA 算法进行简要介绍, 第三部分详细阐述所提出的 SSA-DPC 算法, 第四部分对新算法进行实验分析, 第五部分为总结。

## 2. 相关算法简述

### 2.1. 密度峰值聚类算法(DPC)

密度峰值聚类算法建立在两个基本假设上：1) 类簇中心点的局部密度相较于周围点的密度要高；2) 类簇中心点与比其密度更高的点之间的距离相对较远。

假设有数据集  $S = \{x_1, x_2, \dots, x_n\}$ ,  $d_{ij} = \text{dist}(x_i, x_j)$  是数据点  $x_i$  与  $x_j$  之间的欧氏距离。定义局部密度  $\rho_i$  为:

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

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases}, \quad (1)$$

其中,  $d_c$  是截断距离, 需要人为指定, 一般选取整个数据集的 1%~2% 作为截断阈值,  $\chi(x)$  是指示函数。定义相对距离  $\delta_i$  为:

$$\delta_i = \min_{j: \rho_j > \rho_i} (d_{ij}), \quad (2)$$

当数据点  $x_i$  的局部密度达到最大时, 相对距离变为:  $\delta_i = \max_j (d_{ij})$ 。

局部密度和相对距离确定完成后, 通过绘制决策图选取聚类中心, 一般选择决策图右上方的点, 这些点既有较高的局部密度, 也有较大的相对距离。最后采用一步分配策略进行非聚类中心点的分配: 若数据点  $x_j$  不是中心点, 则将其归入密度比  $x_j$  大且距离  $x_j$  最近的数据点  $x_i$  所在的类, 该过程只执行一次, 不用进行迭代更新[11]。

### 2.2. 麻雀搜索算法(SSA)

根据麻雀种群的生物特性, 将其行为理想化, 制定相应规则, 建立数学模型, 从而得到新颖且有潜力的算法——麻雀搜索算法。SSA 将麻雀种群分为两大类: 发现者和加入者, 发现者拥有较高的能量, 它们的职责是搜索食物并提供相应的觅食区域和方向, 这类群体占总种群的 20%; 加入者相对来说能量较少, 它们利用发现者来获取食物, 占总种群的 80%。当麻雀意识到危险时, 会即刻表现出反捕食状态[12]。

假设麻雀种群为:  $X = (x_{i1}, x_{i2}, \dots, x_{id})^T, i = 1, 2, \dots, n$ ,  $n$  为麻雀总数,  $d$  为变量维数, 麻雀个体对应的适应度值为:  $F_x = [f([x_{i1}, x_{i2}, \dots, x_{id}])]$ 。在 SSA 中, 有较高适应度值的发现者会优先获得食物, 故发现者的位置更新公式为:

$$X_{i,j}^{t+1} = \begin{cases} X'_{i,j} \cdot \exp\left(-\frac{i}{\alpha \cdot \text{iter}_{\max}}\right), & R_2 < ST \\ X'_{i,j} + Q \cdot L, & R_2 \geq ST \end{cases}, \quad (3)$$

其中,  $X'_{i,j}$  表示第  $i$  个麻雀在第  $j$  维的位置,  $t$  为当前迭代次数,  $\alpha \in (0, 1]$  是随机数,  $R_2 \in [0, 1]$  是预警值,  $ST \in [0.5, 1]$  是安全阈值,  $Q$  是服从正态分布的随机数,  $L$  是元素全部为 1 的  $1 \times d$  矩阵。当  $R_2 < ST$  时, 说明觅食环境周围是安全的, 发现者可以大范围搜索食物; 当  $R_2 \geq ST$  时, 说明麻雀感应到了捕食者, 需要撤往安全地带[12]。

对加入者来说, 它们会优先选择和发现者争夺食物资源, 公式如下:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst} - X_{i,j}^t}{i^2}\right), & i > \frac{n}{2} \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^T (AA^{-1})^T \cdot L, & i \leq \frac{n}{2} \end{cases}, \quad (4)$$

其中,  $X_{worst}$  为当前全局最差位置,  $X_p$  为发现者所占最佳位置,  $A$  中每个元素被随机赋值为 1 或 -1。当  $i > \frac{n}{2}$  时, 表示第  $i$  个加入者的适应度值较低, 因未获得食物而处于饥饿状态, 所以将飞往其他可以觅食的地方; 当  $i \leq \frac{n}{2}$  时, 表示第  $i$  个加入者将在当前最优位置附近寻找食物[12]。

为保证其整个觅食过程的安全性, 选取麻雀种群的 10%~20% 作为可以感知危险的麻雀, 其位置更新表达式为:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta |X_{i,j}^t - X_{best}^t|, & f_i > f_g \\ X_{i,j}^t + K \left( \frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon} \right), & f_i = f_g \end{cases}, \quad (5)$$

其中,  $X_{best}$  为当前全局最优位置,  $\beta$  为控制参数的步长, 是一个服从标准正态分布的随机数,  $K \in [-1, 1]$  是调整移动方向的随机数,  $f_i$  是当前麻雀的适应度值,  $f_g$  与  $f_w$  分别表示当前全局最佳与最差适应度值,  $\varepsilon$  为最小常数。当  $f_i > f$  时, 表示麻雀处于种群边缘, 易被捕食者袭击; 当  $f_i = f_g$  时, 表示麻雀处于种群中间且感知到了危险, 为了免受攻击需要向其他麻雀靠近[12]。

### 3. 基于麻雀搜索算法的密度峰值聚类(SSA-DPC)

本文主要从以下两个方面进行改进: 首先更改数据间的距离度量函数来更新距离矩阵; 其次, 利用麻雀搜索算法较强的全局寻优能力对 DPC 算法中的截断距离  $d_c$  进行优化。

#### 3.1. 距离度量函数的改进

密度峰值聚类中, 局部密度和相对距离的计算与所采用的距离度量方式有着密切联系, 距离矩阵构造的好坏很大程度上影响着聚类的性能。传统 DPC 算法采用欧氏距离进行聚类, 但在处理一些较为复杂的数据集, 比如数据各维度的分量不一致时, 欧氏距离不能准确地反映数据分布情况, 不能很好地表现出数据间的特征。因此, 考虑将数据间的距离度量方式改用标准欧氏距离来替代。

标准欧氏距离是对传统欧氏距离的缺点而做的一种改进方案, 可将其看成一种加权欧氏距离。他利用数据样本的标准差, 综合各维度的差异程度, 使得各个维度都满足标准正态分布, 这种处理方式更能体现数据间的分布情况, 且能很好地反映数据间的特征, 其计算公式为:

$$d_{ij} = \sqrt{\sum_{k=1}^d \left( \frac{x_{1k} - x_{2k}}{s_k} \right)^2}, \quad (6)$$

其中,  $d$  为维度,  $s_k$  为标准差。

#### 3.2. 截断距离 $d_c$ 的优化

密度峰值聚类中有且只有一个参数——截断距离  $d_c$ , 但该参数的确定方式是人们根据经验来估计, 主观性很强, 因此利用麻雀搜索算法较强的全局寻优能力对参数进行自适应选取。

本文提出的 SSA-DPC 算法将聚类结果的标准互信息(Normalized Mutual Information, NMI) [13] 指标进行变形作为适应度值目标函数, 即  $d_c$  求解的收敛条件。NMI 指标利用互信息来评价聚类质量, 取值范

围为 $[0,1]$ , 其值越接近 1, 说明聚类结果越接近于真实结果。假设  $U$  为真实标签向量,  $V$  为预测标签向量,  $MI$  为互信息,  $H$  为信息熵, 那么 NMI 指标的数学表达式为:

$$NMI = \frac{MI(U,V)}{\sqrt{H(U)H(V)}}, \quad (7)$$

取 NMI 指标的倒数作为 SSA 中的适应度值函数, 表达式如下:

$$f_i = \frac{1}{NMI(X_{i,j})}, \quad (8)$$

在优化过程中, 通过 SSA 算法进行多次迭代寻优, 找出使 DPC 算法中 NMI 指标最大的  $d_c$  作为当前数据集最优的  $d_c$ , 从而实现算法的聚类。

### 3.3. SSA-DPC 算法描述

SSA-DPC 算法将数据间的距离度量方式用标准欧氏距离来表示, 把密度峰值聚类算法和麻雀搜索算法通过 NMI 指标函数有效地结合在一起, 优化了截断参数  $d_c$ 。记采用标准欧氏距离改进后的 DPC 为 DPC\*。算法具体步骤如下:

输入: 数据集  $S = \{x_1, x_2, \dots, x_n\}$ 。

输出: 聚类结果  $C = \{c_1, c_2, \dots, c_k\}$ ,  $k$  为类簇数目。

步骤 1. 计算数据点间的距离, 构造距离矩阵, 确定  $d_c$  取值范围。

步骤 2. 运行 DPC 算法, 根据聚类结果得到 NMI 指标的值。

步骤 3. 根据数据集大小设置种群大小为  $S$ , 最大迭代次数为  $T$ 。

步骤 4. 初始化种群位置, 得到初始  $d_c$  值。

步骤 5. 根据式(8)运行 SSA 算法, 迭代更新  $d_c$  的值。

步骤 6. 判断算法是否满足终止条件, 若是, 则结束迭代转至下一步; 若否, 则继续迭代。

步骤 7. 将最优的  $d_c$  值代入 DPC\* 算法, 得到聚类结果, 完成聚类。

## 4. 仿真实验及分析

### 4.1. 实验数据集

为了验证 SSA-DPC 算法的有效性, 本文采用 4 个人工数据集和 3 个真实数据集对该算法进行验证, 7 个数据集的具体描述如表 1 所示:

Table 1. Dataset description

表 1. 数据集描述

数据集	样本数量	维数	类别数
Spiral	312	2	3
Jain	373	2	2
Flame	240	2	2
Aggregation	788	2	7
Iris	150	4	3
Wine	178	13	3
Seeds	210	7	3



## 4.2. 性能评估指标

本文采用调整互信息(Adjusted Mutual Information, AMI) [14]、调整兰德系数(Adjusted Rand Index, ARI) [15]和 FM 指数(Fowlkes and Mallows Index, FMI) [16]三个评价指标对算法进行性能评估, 定义如下:

定义 1 调整互信息(AMI) [14]: 假设  $U$  为真实标签向量,  $V$  为预测标签向量,  $MI$  为互信息,  $E$  为期望,  $H$  为信息熵, 那么 AMI 指标的数学表达式为:

$$AMI(U, V) = \frac{MI(U, V) - E[MI(U, V)]}{\max[H(U), H(V)] - E[MI(U, V)]}, \quad (9)$$

定义 2 调整兰德系数(ARI) [15]: 设  $a$  为同属于  $U$  与  $V$  的数据对个数;  $b$  为在  $U$  中属在同类, 在  $V$  中属不同类的数据对个数;  $c$  为在  $U$  中属不同类, 在  $V$  中属同类的数据对个数;  $d$  为在  $U$  与  $V$  中都属不同类的数据对个数[11], 则有:

$$ARI = \frac{2(ad - bc)}{(a+b)(b+d) + (a+c)(c+d)}, \quad (10)$$

定义 3 FM 指数(FMI) [16]: FMI 是平衡了精确度和召回率的调和平均值, 表达式为:

$$FMI = \frac{a}{\sqrt{(a+b)(a+c)}}, \quad (11)$$

## 4.3. 实验结果及分析

使用 SSA-DPC 算法、DPC 算法、DBSCAN 算法以及 K-Means 算法对表 1 中的四个人工数据集 Spiral、Jain、Flame 以及 Aggregation 进行聚类, 这四个数据集的总体分布情况各不相同, 因而可以更清晰直观地反映出四种算法的聚类效果和性能。图 1 为四种算法在 Spiral 数据集上的聚类效果图。

从图中可以看出, Spiral 数据集是由三条螺旋线组成, DPC 算法和 K-Means 算法的聚类效果不是很好, 它们错误地将一个类簇的数据点划分到三个类簇, 而 SSA-DPC 算法和 DBSCAN 算法可以正确地将数据聚集在一起, 聚类效果很好。

图 2 为四种算法在 Jain 数据集上的聚类效果图。

从图中可以看出, Jain 数据集由两个月牙型的数据簇组成, DPC 算法、DBSCAN 算法以及 K-Means 算法在此数据集上的表现效果很差, 尤其是 DBSCAN 算法, 未能将第二个数据簇的数据表现出来, 只有 SSA-DPC 算法能够准确确定类簇的个数, 同时聚类效果十分优秀。

图 3 为四种算法在 Flame 数据集上的聚类效果图。

从图中可以看出, Flame 数据集由两个形状不同的数据簇组成, SSA-DPC 算法、DPC 算法以及 DBSCAN 算法的聚类效果都很好, 只存在细微差别, 而 K-Means 算法将火焰型的数据集用直线割裂成了两部分, 错误地将一个类簇的数据点划分到两个类簇。

图 4 为四种算法在 Aggregation 数据集上的聚类效果图。

从图中可以明显看出, 四种算法都未能精准地将七个类簇识别, 但相对来说, SSA-DPC 算法将数据集下方难以分辨的三个球形类簇很好地分离开来, 聚类效果较好, 其他三种算法要么错误的将不同类簇的数据聚集到了一起, 要么不能准确识别类簇数目。

为了更好地对比四种算法的聚类效果, 表 2 给出了各算法 7 个数据集上的聚类评价指标值:

对比表中数据指标值可以发现: SSA-DPC 算法对七个数据集进行聚类所得的 AMI、ARI 和 FMI 指标值均得到了更好的结果, 聚类性能也有了很大的提升。在四个人工数据集中, SSA-DPC 算法的三个

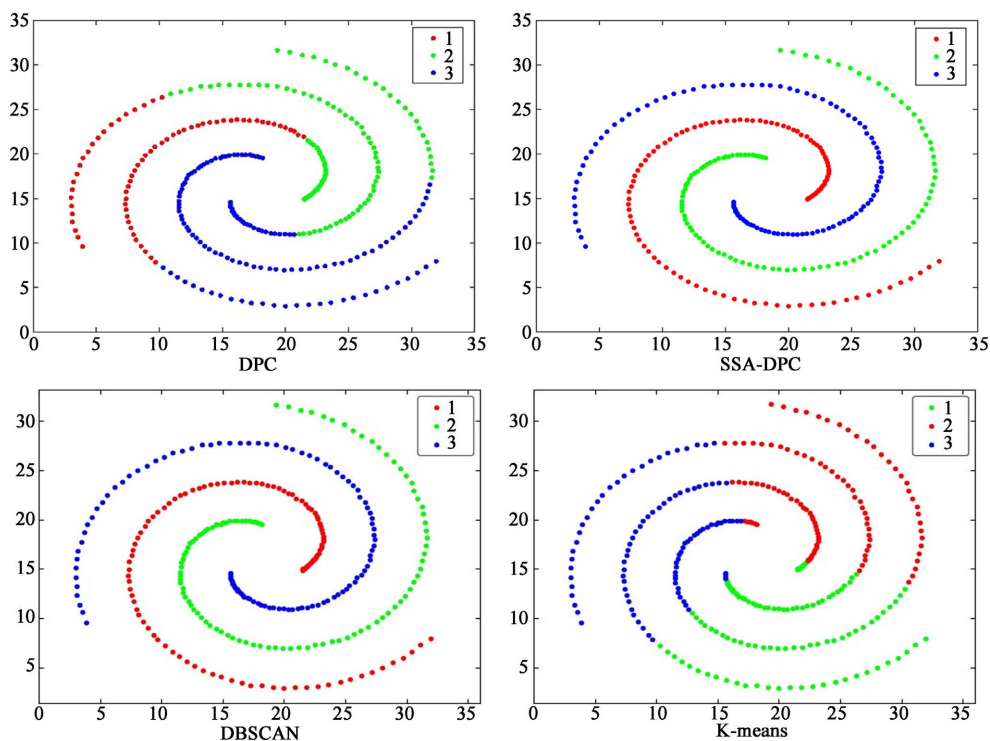


Figure 1. The renderings of the four algorithms in the Spiral dataset

图 1. 四种算法在 Spiral 数据集的效果图

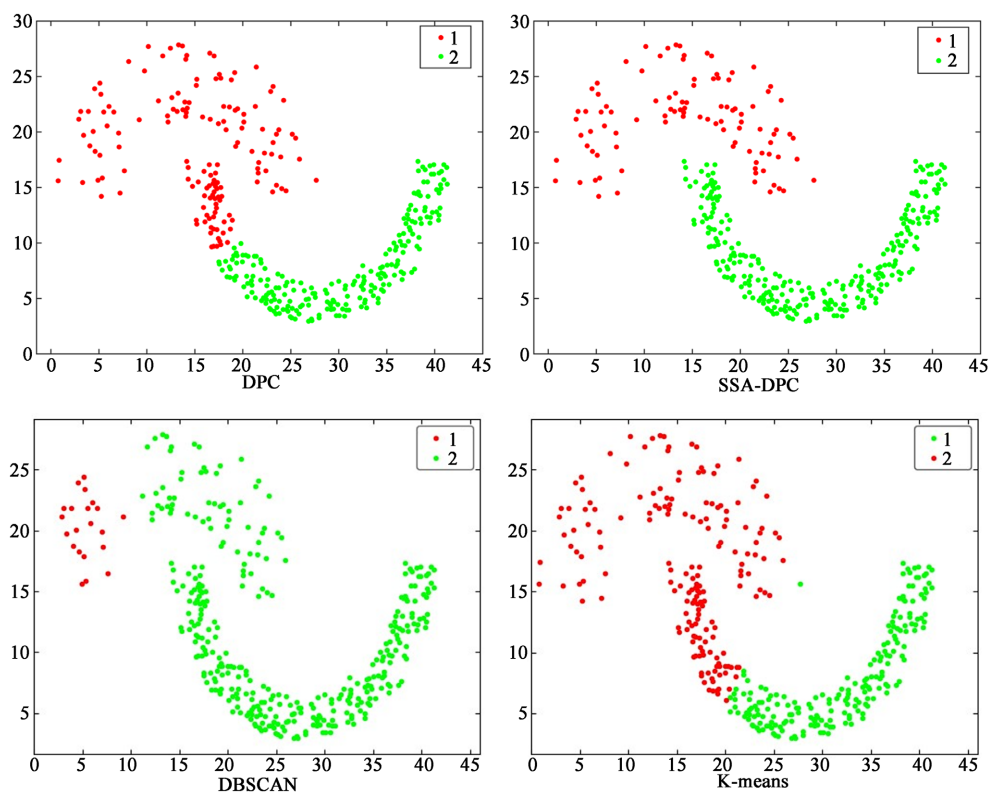


Figure 2. The renderings of the four algorithms in the Jain dataset

图 2. 四种算法在 Jain 数据集的效果图

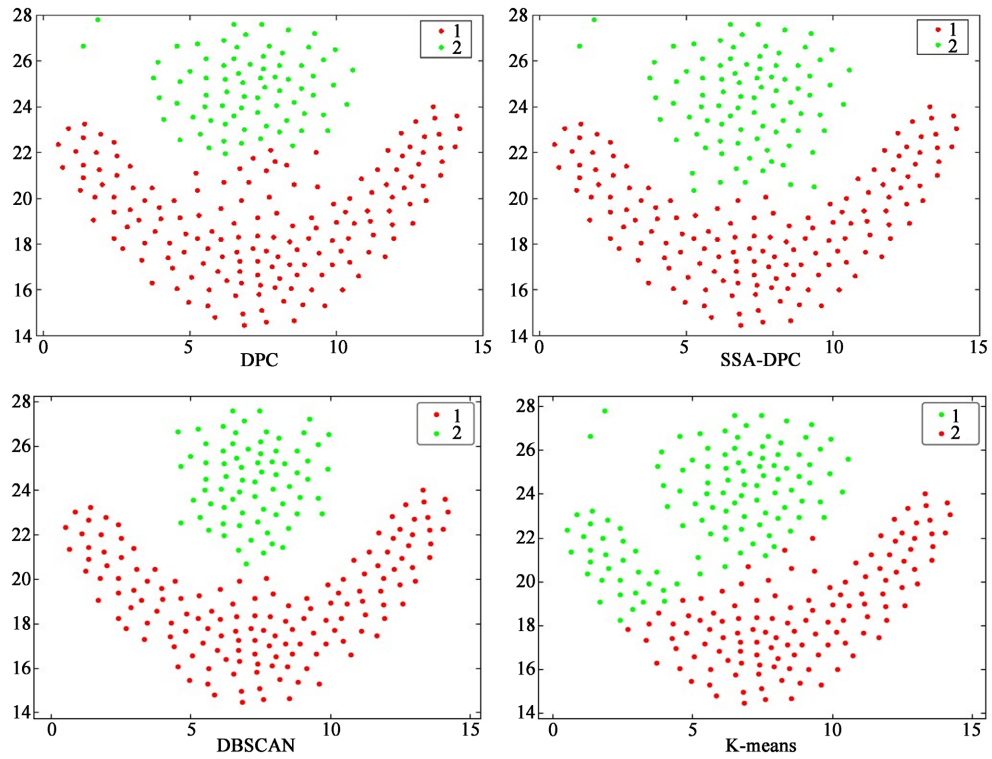


Figure 3. The renderings of the four algorithms in the Flame dataset

图 3. 四种算法在 Flame 数据集的效果图

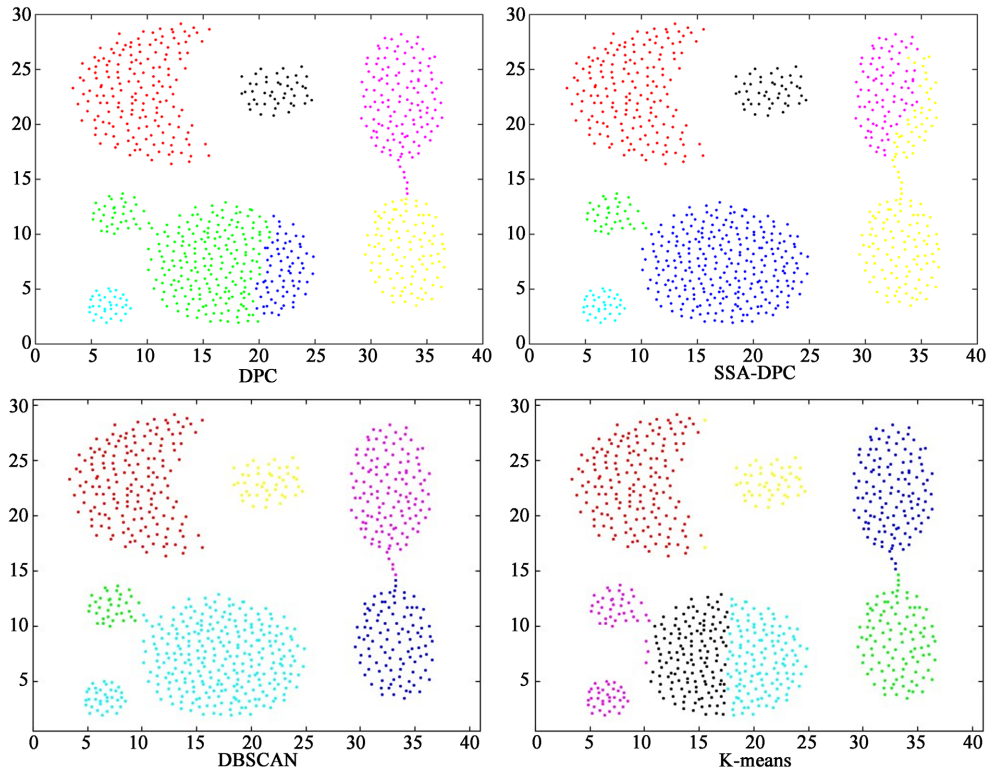


Figure 4. The effect of the four algorithms in the Aggregation dataset

图 4. 四种算法在 Aggregation 数据集的效果图



**Table 2.** Experimental results of the four algorithms on 7 datasets**表 2.** 四种算法在 7 个数据集上的实验结果

算法	DPC			SSA-DPC			DBSCAN			K-Means		
	AMI	ARI	FMI	AMI	ARI	FMI	AMI	ARI	FMI	AMI	ARI	FMI
Spiral	0.165	0.140	0.143	1.000	1.000	1.000	1.000	1.000	1.000	0.001	0.000	0.327
Jain	0.505	0.514	0.790	1.000	1.000	1.000	0.273	0.290	0.806	0.370	0.324	0.700
Flame	0.761	0.822	0.920	0.935	0.966	0.984	0.718	0.778	0.889	0.416	0.464	0.741
Aggregation	0.899	0.785	0.830	0.926	0.913	0.931	0.959	0.915	0.937	0.892	0.774	0.825
Iris	0.704	0.560	0.756	0.716	0.701	0.799	0.627	0.541	0.675	0.758	0.730	0.820
Wine	0.404	0.292	0.619	0.566	0.489	0.676	0.576	0.425	0.596	0.428	0.371	0.583
Seeds	0.514	0.410	0.665	0.576	0.597	0.734	0.622	0.518	0.658	0.772	0.790	0.859

指标值几乎都等于或者接近于 1, 这表示此算法在类似数据集中都有较好的聚类效果。在 Iris 和 Seeds 数据集中, SSA-DPC 的聚类效果略差于 K-Means 算法, 但仍高于 DPC 算法和 DBSCAN 算法。

## 5. 结语

为了更好地体现密度峰值聚类算法的数据分布, 减弱手动确定截断距离的主观性, 本文结合群智能优化算法中的麻雀搜索算法, 提出一种基于麻雀搜索算法改进的密度峰值聚类算法。该算法用标准欧氏距离取代传统的欧式距离, 更改数据间距离度量方式; 将 NMI 指标作为适应度值目标函数, 设定取值范围, 对截断距离进行寻优。通过在人工数据集和真实数据集上的实验验证, 可以得出结论: 本文所提出的 SSA-DPC 算法可以有效反映数据分布, 能自动确定截断参数, 是一种有效的自适应聚类算法, 并且对任意形状类簇都有着较好的聚类效果。下一步的研究目标是对剩余点的分配策略进行改进, 尝试将 DPC 算法应用于实际问题。

## 基金项目

国家自然科学基金(11461039)。

## 参考文献

- [1] Han, J., Pei, J. and Kamber, M. (2011) Data Mining: Concepts and Techniques. Elsevier, Amsterdam.
- [2] Xu, R. and Wunsch, D. (2005) Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks*, **16**, 645-678. <https://doi.org/10.1109/TNN.2005.845141>
- [3] Bai, L., Cheng, X.Q., Liang, J.Y., Shen, H.W. and Guo, Y.K. (2017) Fast Density Clustering Strategies Based on the *k*-Means Algorithm. *Pattern Recognition*, **71**, 375-386. <https://doi.org/10.1016/j.patcog.2017.06.023>
- [4] Xu, D. and Tian, Y. (2015) A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, **2**, 165-193. <https://doi.org/10.1007/s40745-015-0040-1>
- [5] Xie, J.Y., Gao, H.C., Xie, W.X., Liu, X.H. and Grant, P.W. (2016) Robust Clustering by Detecting Density Peaks and Assigning points Based on Fuzzy Weighted K-Nearest Neighbors. *Information Sciences*, **354**, 19-40. <https://doi.org/10.1016/j.ins.2016.03.011>
- [6] Rodriguez, A. and Laio, A. (2014) Clustering by Fast Search and Find of Density peaks. *Science*, **344**, 1492-1496. <https://doi.org/10.1126/science.1242072>
- [7] Morris, K. and McNicholas, P.D. (2016) Clustering, Classification, Discriminant Analysis, and Dimension Reduction via Generalized Hyperbolic Mixtures. *Computational Statistics & Data Analysis*, **97**, 133-150. <https://doi.org/10.1016/j.csda.2015.10.008>

- [8] 蒋礼青, 张明新, 郑金龙, 戴娇, 尚赵伟. 快速搜索与发现密度峰值聚类算法的优化研究[J]. 计算机应用研究, 2016, 33(11): 3251-3254.
- [9] Jian, H. and Xu, E. (2017) An Improved Density Peak Clustering Algorithm. *International Conference on Intelligent Data Engineering and Automated Learning*, Guilin, 30 October-1 November 2017, 211-221. [https://doi.org/10.1007/978-3-319-68935-7\\_24](https://doi.org/10.1007/978-3-319-68935-7_24)
- [10] Xue, J. and Shen, B. (2020) A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. *Systems Science & Control Engineering*, **8**, 22-34. <https://doi.org/10.1080/21642583.2019.1708830>
- [11] 陈俊芬, 张明, 赵佳成. 复杂高维数据的密度峰值快速搜索聚类算法[J]. 计算机科学, 2020, 47(3): 79-86.
- [12] 薛建凯. 一种新型的群智能优化技术的研究与应用[D]: [硕士学位论文]. 上海: 东华大学, 2020.
- [13] Strehl, A. and Ghosh, J. (2002) Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research*, **3**, 583-617.
- [14] Viola, P. and Wells III, W.M. (1997) Alignment by Maximization of Mutual Information. *International Journal of Computer Vision*, **24**, 137-154. <https://doi.org/10.1023/A:1007958904918>
- [15] Hubert, L. and Arabie, P. (1985) Comparing Partitions. *Journal of Classification*, **2**, 193-218. <https://doi.org/10.1007/BF01908075>
- [16] Fowlkes, E.B. and Mallows, C.L. (1983) A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, **78**, 553-569. <https://doi.org/10.1080/01621459.1983.10478008>