基于核密度估计的 K-CFSFDP 聚类算法

董晓君 程春玲
(南京邮电大学计算机学院 南京 210003)

摘 要 快速搜索和发现密度峰值的聚类算法(Clustering by Fast Search and Find of Density Peaks,CFSFDP)是一种新的基于密度的聚类算法,它通过发现密度峰值来有效地识别类簇中心,具有聚类速度快、实现简单等优点。针对 CFSFDP 算法的准确性依赖于数据集的密度估计和截断距离(dc)的人为选择问题,提出一种基于核密度估计的 K-CFSFDP 算法。该算法利用无参的核密度估计分析数据点的分布特征并自适应地选取 dc,从而搜索和发现数据点的密度峰值,并以峰值点数据作为初始聚类中心。基于 4 个典型数据集的仿真结果表明,K-CFSFDP 算法比 CFSFDP,K-means 和 DBSCAN 算法具有更高的准确度和更强的鲁棒性。

关键词 聚类,核密度估计,密度峰值,聚类中心
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.11.038

K-CFSFDP Clustering Algorithm Based on Kernel Density Estimation

DONG Xiao-jun CHENG Chun-ling
(College of Computer,Nanjing University of Posts and Telecommunications,Nanjing 210003,China)

Abstract The CFSFDP (Clustering by Fast Search and Find of Density Peaks) is a new density-based clustering algorithm,it can identify the cluster centers effectively by finding the density peaks,and it has the advantages of fast clustering speed and simple realization. The accuracy of CFSFDP algorithm depends on the density estimation in the dataset and cut off distance (dc) of artificial selection. Therefore,an improved K-CFSFDP algorithm based on kernel density estimation was presented. The algorithm uses non parametric kernel density to analyze distribution of data points and selects the dc adaptively to search and find the peak density of data points,with the peak point data as the initial cluster centers. The simulated results on 4 typical datasets show that the K-CFSFDP algorithm has better performance in accuracy and better robustness than CFSFDP,K-means and DBSCAN algorithm.

Keywords Clustering,Kernel density estimation,Density peak,Cluster center

1 引言

聚类作为传统数据挖掘算法中的一种,是发现数据中潜藏的、事先未知的可能群组,然后根据其相似性分组成多个聚类或簇的过程。聚类结果使得同一簇内的对象具有较高的相似性,不同簇间的对象相似性较低。通过聚类分析,可以从给定的数据集中发现数据间的有价值的联系和知识,进而揭示隐藏的模式和规律。聚类分析已被广泛应用于科学数据分析和工程系统等领域[1]。

传统的聚类算法主要包括划分式聚类算法、层次聚类算法和基于密度的聚类算法[2]。划分式聚类算法需要预先指定聚类数目或聚类中心,通过反复迭代运算,逐步降低目标函数的误差值,当目标函数收敛时,得到最终的聚类结果。K-means[3]是应用得最为广泛的划分式聚类算法。然而,K-means 算法的聚类结果严重依赖于初始类簇中心,对噪声点敏感且类簇数 K 需要事先设定。层次聚类算法按照某种方法对数据进行层次分解,直到满足某种条件为止。按照分类原理的不同,层次聚类算法可以分为凝聚和分裂两种[4]。基于密度的聚类算法通过局部数据密度进行聚类,可以发现任意形状的类簇。其中,DBSCAN[5]算法通过选择一个密度阈值,将数据点分配到用高密度数据值确定的类簇中,从而能够有效去除噪声点和离群点,是基于密度的聚类算法中的经典代表。然而,该算法需要用户在没有先验知识的情况下设定参数 Eps 和 $MinPts$,参数的选择影响着最终的聚类结果,同时它不能完全检测出边界点且不适用于密度变化较大的数据集。

2014 年 6 月,*Science* 发表了自动确定类簇数和聚类中心的新聚类算法 CFSFDP[6](Clustering by Fast Search and Find of Density Peaks),该算法的基本思想是将具有局部极大密度且与其他密度更高点有较远距离的数据点视为聚类中心,通过快速搜索聚类中心,将每一个非聚类中心的数据点沿着密

到稿日期:2017-10-27 返修日期:2018-02-19
董晓君(1993—),男,硕士生,主要研究方向为数据挖掘,E-mail:dongxiaojun_njupt@163.com;**程春玲**(1972—),女,教授,CCF 会员,主要研究方向为数据管理、云计算中的资源管理和优化等,E-mail:chengcl@njupt.edu.cn(通信作者)。

度递增的最近邻方向划分到对应的聚类中心中，实现数据划分。与 K-means 算法相比，CFSFDP 算法可以自动获取类簇的个数且复杂度较低；与 DBSCAN 算法相比，CFSFDP 算法不需要迭代计算距离，可在噪声环境下聚类任意形状的数据集，且适用于大规模数据的聚类分析。然而，CFSFDP 算法的有效性取决于对数据集的密度评估和截断距离 dc 的选择，算法没有采用统一的密度度量准则，而是针对数据集规模采用不同的数据密度评估准则，且在数据规模较小时，dc 的选择对聚类结果的影响较大。

针对截断距离 dc 值的选取和局部密度评估问题，研究者们提出了 CFSFDP 算法的改进方案。文献[7]根据标准偏差可以反映一个数据集的离散程度，将 dc 表示为所有数据属性标准偏差的一个二维向量，然后通过计算 dc 的平均值得到截断距离，其中引入了一个平衡参数 ω 来控制截断距离的大小。文献[8]运用数据场的相关理论，将参数 dc 的选取问题转换成求势函数中影响因子的过程，用势函数的熵来优化影响因子，最后得到最优的截断距离 dc。文献[9]根据分布的簇内数据对象到中心的平均距离得到一个聚类的质量评价，并采用粒子群算法反复寻找最优 dc 值得到较优的聚类质量，其中 dc 作为粒子群优化算法的解，聚类质量作为评价每个 dc 优劣的适应度函数，最终找到最优的 dc 值。文献[10]首先根据数据集的特点，参考近邻距离变化情况来计算截断距离参数 dc 值，使得聚类结果更加准确。文献[11]借鉴 DENCLUE 算法中密度吸引子的思想，通过动态控制核函数的带宽来定义每个数据点的吸引度，从而改进了局部密度的计算方法，解决了 CFSFDP 算法不能聚类非均匀密度的数据集问题。文献[12]引进 K 最近邻的思想来计算数据点的局部密度，使得算法取得了较好的聚类结果，但是聚类的过程与原算法相同，因此，CFSFDP 算法的缺陷同样存在。文献[13]也是根据 K 最近邻提出了一种新的局部密度评估方法，并结合模糊准则给出两种新的数据点分配策略，使得改进算法比原算法更具鲁棒性，但是数据模型变得更加复杂且算法不能满足自适应参数的需求。

综上所述，本文根据现有截断距离 d_c 的选取方法和局部密度评估改进方法的优劣，基于 CFSFDP 算法的核心思想和核密度估计的相关理论，提出了改进的 K-CFSFDP 算法。主要工作包括：1）提出利用非参数的核密度估计来计算数据点的局部密度；2）提出利用改进的核函数窗宽参数确定方法来自动选择合适的截断距离 d_c 值，并利用窗宽参数来有效地检测类簇间的边界区域和噪声点；3）通过实验将本文算法与现有算法进行比较分析。

2　CFSFDP 算法

CFSFDP 算法假设聚类中心是被具有较低局部密度的邻居点包围的数据点，且与具有更高密度的任何其他点有相对较远的距离。该算法首先通过启发式方法确定聚类中心点及其个数，然后将其他非聚类中心点归类到比它们的密度更大且最相近的数据点所属的类簇中，以实现数据的划分。算法中涉及两个基本概念，即数据点的局部密度和与更高密度点的相对距离。

2.1　局部密度和相对距离

给定数据集 $D=\{x_1,x_2,\cdots,x_n\}$，对于任意的数据点 $x_i\in D(1\leqslant i\leqslant n)$，定义局部密度 ρ_i 为：

$$\rho_i=\sum_{j\neq i}\chi(d_{i,j}-d_c) \tag{1}$$

其中，$d_{i,j}$ 为数据点 x_i 和 x_j 间的欧氏距离；d_c 为截断距离，当 $x<0$ 时，$\chi(x)=1$，否则 $\chi(x)=0$。

对于规模较小的数据集，采用指数核来计算局部密度，计算式如式(2)所示：

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

定义与更高密度点的相对距离 δ_i 为数据点 x_i 到任何比其密度大的点的距离的最小值，计算式如下：

$$\delta_i=\min_{j:\rho_j>\rho_i}(d_{i,j}) \tag{3}$$

显然，具有全局最大密度值的数据点不存在更高密度近邻，可令其相对距离 δ_i 等于所有数据点间距离的最大值。

2.2　决策图和数据点的分配

通过计算每个数据点 $x_i(1\leqslant i\leqslant n)$ 的局部密度 ρ_i 和与更高密度点的相对距离 δ_i，CFSFDP 算法将原始数据集 D 映射到由局部密度 ρ 和与更高密度点的相对距离 δ 组成的二维特征空间中，以构造决策图，选择 ρ 和 δ 都较大的数据点为聚类中心。对于剩余数据点，CFSFDP 算法将其分配到有更高密度的最近邻所属的类簇中，类簇分布只需一步即可完成。

为了避免噪声点对聚类结果的影响，CFSFDP 算法定义了类簇的边界区域。一个类簇的边界区域由属于该类簇但与其他类簇数据点的距离小于截断距离 d_c 的数据点构成。以每一个类簇边界区域中密度最大数据点的密度为阈值，定义该类簇中密度大于阈值的数据点为本类簇的核心点，该类簇的其他数据点为噪声点。

在 CFSFDP 算法中，局部密度通过统计由参数 d_c 确定的邻域中的数据对象的个数来计算。这种密度估计对参数非常敏感。文献[6]认为截断距离 d_c 的值应使得数据集中邻域的数量占整个数据集的 1%～2%。同时文献[6]认为截断距离 d_c 的取值虽然会影响数据点的局部密度和相对距离，但不会影响最终的聚类结果，但根据实验结果发现局部密度计算公式的选择和截断距离 d_c 的取值都会严重影响最终的聚类结果。

3　K-CFSFDP 算法的设计

3.1　算法思想

相比于 K-means 和 DBSCAN 算法，CFSFDP 算法能够有效发现不同形状、不同密度的簇，并可以高效地进行数据对象的分配和离群点的剔除。然而，该算法在确定截断距离参数 d_c 值时没有研究原始数据集中数据对象的分布特征，而是由用户基于启发的方式人为定义 d_c 的值，使得聚类过程带有一定的随意性和主观性，不利于算法的实现与应用。

鉴于此，本文提出一种非参数的核密度估计方法来计算

原始数据集中数据点的局部密度；同时利用分析出的数据对象的分布特征，提出利用改进的窗宽参数确定方法来自适应选取截断距离 d_c 的值，并且利用窗宽参数来有效检测类簇的边界区域并去除噪声点，以此来对基于密度峰值的聚类算法进行改进，从而实现自动化确定截断距离 d_c 值。由于核密度估计能够处理任意的密度分布，无需对数据对象分布的形式做出假设，仅以数据点作为概率密度函数估计的依据，因此得到的聚类结果更加准确。

3.2 核密度估计

非参数的密度估计是数据统计分析的重要工具。核密度估计(Kernel Density Estimation，KDE)是非参数密度估计中最为常用的方法[14]之一。标准评估密度的方法是对每个数据点引进一个高斯核函数，并计算整个数据集中核函数的值。假设 $x_1,x_2,\cdots,x_n$ 为取值于 R 的独立分布随机变量，其服从的分布密度函数为 $f(x),x\in R$。定义函数：

$$\hat{f}_h(x;h)=\frac{1}{nh}\sum_{i=1}^{n}K(\frac{x-x_i}{h}) \tag{4}$$

其中，$\hat{f}_h(x;h)$为密度函数 $f(x)$的核密度估计；$K(\frac{x-x_i}{h})$为核函数；h 为窗宽或光滑参数；n 为样本数量。

通常使用高斯核函数来评估密度：

$$K(x,x_i;h)=\frac{1}{\sqrt{2\pi}h}e^{-\frac{(x-x_i)^2}{2h}} \tag{5}$$

本文根据核密度估计的理论，运用式(4)来计算 CFSFDP 算法中的局部密度。但是，KDE 的有效性取决于窗宽参数 h 的合理选择。如果 h 太小，那么密度估计局限于观察数据的附近局部区域，将导致估计的密度函数出现很多错误的峰值，在聚类分析的应用中表现为一个簇被错误地拆分为多个簇。如果 h 过大，那么密度估计会把概率密度分得太稀疏，这样会过滤掉一些重要的特征，在聚类分析应用中表现为噪声被错误地归入簇，若干个自然簇也被错误地合并成一个簇。

3.3 自适应参数选取

依据上文的分析，假如能够通过数学方法确定窗宽参数 h 的大小，则可以确定参数 d_c 的值。统计学上，通常使用积分均方误差(Mean Integrated Square Error，MISE)来判断密度估计的质量，从而优化选择窗宽参数 h。

$$\begin{aligned}MISE(h)&=E\int(\hat{f}_h(x)-f(x))^2\mathrm{d}x\\&=AMISE(h)+o(1/(nh)+h^4)\end{aligned} \tag{6}$$

其中，

$$AMISE(h)=\frac{R(K)}{nh}+\frac{1}{4}m_2(K)^2h^4R(f'') \tag{7}$$

其中，$R(g)=\int g(x)^2\mathrm{d}x$，$m_2(K)=\int x^2K(x)\mathrm{d}x$。

为了使 $AMISE(h)$最小，必须把 h 设在某个中间值，从而避免$\hat{f}_h(x;h)$有过大的偏差，因此将式(7)转化为求极点的问题：

$$\frac{\partial}{\partial h}AMISE(h)=-\frac{R(K)}{nh^2}+m_2(K)^2h^3R(f'')=0 \tag{8}$$

$$h=\frac{R(K)^{1/5}}{m_2(K)^{2/5}R(f'')^{1/5}n^{1/5}} \tag{9}$$

针对最小化 $AMISE(h)$得到的最优窗宽中含有的未知量 $R(f'')$，运用拇指法则使用方差和估计方差相匹配的正态密度来替换 f。这就相当于用 $R(\emptyset'')/\hat{\sigma}^5$ 估计 $R(f'')$，其中 $\emptyset$ 为标准正态密度函数，若取核函数为高斯密度核函数，$\hat{\sigma}$为样本方差，则利用拇指法则可得到：

$$h=(4/3n)^{1/5}\hat{\sigma} \tag{10}$$

最后，本文定义 $h=10\sqrt{h}/33$ 来改进类簇的边界区域，然后指定边界区域中密度最高的点为密度阈值来去除噪声点。

3.4 算法描述与分析

根据上文的算法设计，K-CFSFDP 算法的具体步骤如下：

1)计算数据点间的欧氏距离，并根据式(10)计算窗宽参数 h 的值，得到整体数据密度阈值 d_c；

2)根据式(3)和式(4)计算每个数据点 x_i 的局部密度 ρ_i 和相对距离 δ_i；

3)从由 ρ_i 和 δ_i 构成的决策图中选择出聚类中心个数，即确定聚类个数；

4)分配剩余数据点到高于当前数据点密度的最近数据点的类中；

5)利用窗宽 $h=10\sqrt{h}/33$ 计算类簇的边界区域，然后指定边界区域中密度最高点的密度值为去除噪声数据点的阈值，将密度小于该阈值的数据点划为噪声点。

对于数据规模为 n 的数据集，K-CFSFDP 算法存储距离矩阵的空间复杂度为 $O(n^2)$，这也是该算法空间复杂度的主要来源。K-CFSFDP 算法的时间复杂度主要来自 4 个部分：1)计算数据点间的欧氏距离；2)计算每个数据点的局部密度 ρ_i；3)计算每个数据点的相对距离 δ_i；4)选择最优的窗宽参数。其中，前 3 个部分的过程的时间复杂度均为 $O(n^2)$，最后 1 个部分的时间复杂度为 $O(n)$，因此总的时间复杂度为 $O(n^2)$。

4 仿真实验与结果分析

4.1 实验环境设置

实验采用测试聚类算法性能的经典的 4 个人工数据集(Point[6]，Flame[15]，Aggregation[16] 和 Spiral[17])对本文 K-CFSFDP 算法进行测试和评价。

实验仿真环境为 Matlab 2013a，硬件配置为 Intel(R) Core(TM) i5-2450M CPU@2.50 GHz，4 GB 内存，500 GB 硬盘，操作系统为 Windows 8.1 专业版。

实验从算法的鲁棒性和聚类准确度两个方面对算法的性能进行评估。鲁棒性从聚类效果图和识别核心点数目两个方面进行评价；准确度通过 ACC 指标进行评价，ACC 值越大，算法的聚类准确度越高。

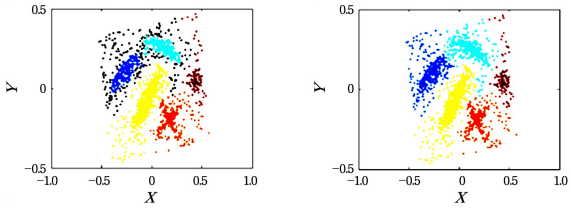
4.2 实验结果及其分析

4.2.1 鲁棒性分析

本节分别将 K-CFSFDP 算法和 CFSFDP 算法在上述 4 个标准的人工数据集中进行实验，以验证算法的鲁棒性，其中 CFSFDP 算法是文献[6]提供的源代码。实验得到的聚类效

果图如图 1 所示。

Point 数据集由 2000 个数据点和 5 个簇组成，这个数据集包含了高重叠的密度峰值，并且是任意形状的簇。图 1(a)显示了 CFSFDP 算法将边界区域的噪声点作为核心点。图 1(b)显示了 K-CFSFDP 算法有更好的聚类效果并能通过参数 h 去除数据集中的噪声点。

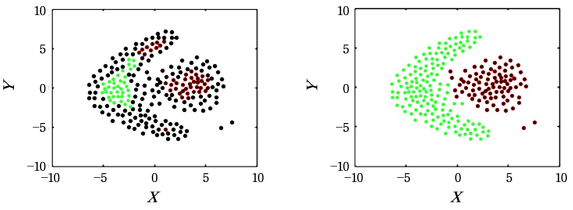


(a)CFSFDP　　(b)K-CFSFDP

图 1　CFSFDP 算法与 K-CFSFDP 算法对 Point 数据集的聚类结果

Fig. 1　Clustering results of CFSFDP algorithm and K-CFSFDP algorithm on Point dataset

Flame 数据集由 240 个数据点和 2 个簇组成。图 2(a)给出了 CFSFDP 算法将大多数核心点识别为噪声点并且不能识别出密度相连簇的紧密关系。图 2(b)给出了由 K-CFSFDP 算法聚类生成的簇，由于采用自适应 d_c 值来准确评估数据集的密度，K-CFSFDP 算法能够生成更好的簇结构并且能发现数据集中密度相连的点。

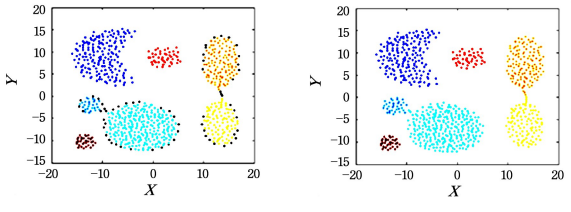


(a)CFSFDP　　(b)K-CFSFDP

图 2　CFSFDP 算法与 K-CFSFDP 算法对 Flame 数据集的聚类结果

Fig. 2　Clustering results of CFSFDP algorithm and K-CFSFDP algorithm on Flame dataset

Aggregation 数据集由 788 个数据点和 7 个簇组成。图 3(a)显示了 CFSFDP 算法将一些边界点作为噪声点。图 3(b)显示了 K-CFSFDP 算法能够正确识别出边界点为核心点，并能发现密度相连的点。

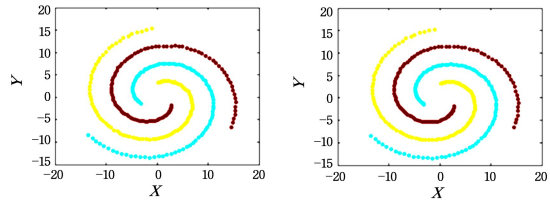


(a)CFSFDP　　(b)K-CFSFDP

图 3　CFSFDP 算法与 K-CFSFDP 算法对 Aggregation 数据集的聚类结果

Fig. 3　Clustering results of CFSFDP algorithm and K-CFSFDP algorithm on Aggregation dataset

Spiral 数据集由 312 个数据点和 2 个簇组成。由于 Spiral 数据集没有重叠的密度区域，因此 CFSFDP 算法和 K-CFSFDP 算法生成的簇结构相同。CFSFDP 算法与 K-CFSFDP 算法对 Spiral 数据集的聚类结果如图 4 所示。



(a)CFSFDP　　(b)K-CFSFDP

图 4　CFSFDP 算法与 K-CFSFDP 算法对 Spiral 数据集的聚类结果

Fig. 4　Clustering results of CFSFDP algorithm and K-CFSFDP algorithm on Spiral dataset

总的来说，CFSFDP 算法不能在类簇的边界区域表达出紧密的密度关系，而 K-CFSFDP 算法得到的聚类结果更精确。K-CFSFDP 算法生成的紧密的簇结构验证了其在大数据集上的鲁棒性，同时，K-CFSFDP 算法在小的数据集上也同样有效，如表 1 所列。

表 1　不同数据集上的 d_c 值和核心点数的比较

Table 1　Comparison of d_c values and core points on different datasets

数据集	CFSFDP		K-CFSFDP	
	d_c 值	核心点数	d_c 值	核心点数
Point	0.033	1618	0.086	2000
Flame	0.930	81	0.417	240
Aggregation	1.860	703	0.538	788
Spiral	1.755	312	0.157	312

从表 1 中的 CFSFDP 算法识别出的核心点数目的统计中可以看出，本文提出的 K-CFSFDP 算法能够更准确地识别出簇中核心点而不依赖于数据集的规模，同时识别出的簇中核心点数更准确和合理。而 CFSFDP 算法会将核心点误判为噪声点，其虽然能够发现密度和边界点，但高度依赖于数据集的规模。因此，K-CFSFDP 算法对于不同规模的数据集更具鲁棒性。

4.2.2　准确度分析

为了进一步验证 K-CFSFDP 算法的准确率，本节将本文方法与 CFSFDP，K-means 和 DBSCAN 算法进行比较。其中，DBSCAN 算法采用作者提供的源码，K-means 则调用 Matlab 中的库函数。实验结果采用聚类准确率(ACC)指标进行性能评价。对于 N 个离散的样本 x_i，y_i 和 c_i 分别表示类别标签和 x_i 的预测簇标签，ACC 的计算式如下：

$$ACC=\sum_{i=1}^{N}\delta(y_i,map(c_i))/N \tag{11}$$

其中，map(·)通过匈牙利算法将每个簇标签映射到一个类别标签。映射是一个最优化过程，当 $y_i=c_i$ 时，$\delta(y_i,c_i)$为 1，否则为 0。ACC 值越大，表示聚类结果越好。4 个算法分别在上述 4 个人工数据集上运行 10 次，得到的平均准确率如表 2 所列。

表 2 K-CFSFDP,CFSFDP,K-means 和 DBSCAN 的平均准确率

Table 2 Average accuracy of K-CFSFDP,CFSFDP,K-means and DBSCAN

聚类算法	Point	Flame	Aggregation	Spiral
K-CFSFDP	0.983	0.997	0.998	1.000
CFSFDP	0.732	0.990	0.996	1.000
K-means	0.723	0.848	0.752	0.426
DBSCAN	0.646	0.986	0.995	1.000

从表 2 中可以看出,改进的 K-CFSFDP 算法和 CFSFDP 算法在整体上的准确度明显高于其他两个算法,这说明了基于密度峰值的聚类算法的优越性。CFSFDP 算法以密度峰值作为聚类中心,根据局部密度和相对距离完成簇的划分。而 K-means 算法是基于划分的聚类算法,可以将数据划分到特定形状的簇,但是不能发现任意形状的簇,导致其准确度较低。同时,DBSCAN 算法需要通过判断核心点、边界点及噪声点来划分簇,密度阈值选取的不合理会造成误判,从而导致聚类的准确度降低。

综上所述,相较于测试的对比算法,本文提出的 K-CFSFDP 算法是更具鲁棒性且更准确的。CFSFDP 算法的截断距离参数 d_c 值难以评估,同时使用不同的方法评估给定数据集的密度,会导致在特定的数据集上生成较多的噪声点。K-CFSFDP算法能够发现任意形状的簇并使用自适应的方式来评估 d_c 值,因此更为有效。综上,K-CFSFDP 算法的聚类结果相较于 CFSFDP 算法是更具鲁棒性的,并且在准确性上比 K-means 算法和 DBSCAN 算法更高。

结束语 针对 CFSFDP 算法聚类的准确性依赖于数据集的密度估计和截断距离 d_c 人为选择的问题,本文结合核密度估计的思想提出了 K-CFSFDP 算法,该算法能够更好地自适应地选择截断距离 d_c 并能更准确地评估数据集的密度,以此来生成更准确的簇并有效识别出簇中的核心点。在 4 个人工数据集上的对比实验验证了所提方法具有更高的准确度和更强的鲁棒性。

然而,CFSFDP 算法和 K-CFSFDP 算法在使用决策图选择聚类中心的过程中也需要人为参与,同时在多个密度峰值时聚类的效果不理想。因此,下一步的工作是研究如何自适应地选取聚类中心以及解决多个密度峰值的问题。

参 考 文 献

[1] MENG X F,CI X. Big Data Management:Concepts,Techniques,and Challenges[J]. Computer Research and Development,2013,50(1):146-169. (in Chinese)
孟小峰,慈祥. 大数据管理:概念、技术与挑战[J]. 计算机研究与发展,2013,50(1):146-169.

[2] SUN J G,LIU J,ZHAO L Y. Study on clustering algorithms [J]. Journal of Software,2008,19(1):48-61. (in Chinese)
孙吉贵,刘杰,赵连宇. 聚类算法研究[J]. 软件学报,2008,19(1):48-61.

[3] MACQUEEN J. Some methods for classification and analysis of multivariate observations[C]// Fifth Berkeley symposium on mathematical statistics and probability. Berkeley: California Press,1967,1(14):281-297.

[4] GELARD R,GOLDMAN O,SPIELER I. Investigating diversity of clustering methods:An empirical comparison[J]. Elsevier Science Publishers B. V,2007,63(1):155-156.

[5] BIRANT D,KUT A. ST-DBSCAN:An algorithm for clustering spatial-temporal data [J]. Data & Knowledge Engineering,2007,60(1):208-221.

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

[7] GAO J,ZHAO L,CHEN Z,et al. ICFS:An Improved Fast Search and Find of Density Peaks Clustering Algorithm[C]// IEEE International Conference on Pervasive Intelligence and Computing. Auckland:IEEE Press,2016:537-543.

[8] WANG S,WANG D,LI C,et al. Clustering by fast search and find of density peaks with data field [J]. Chinese Journal of Electronics,2016,25(3):397-402.

[9] CHEN J Y,HE H H. Research on Clustering Algorithm Based on Density-based Clustering Center for Automatic Determination of Mixed Attribute Data [J]. Journal of Automation,2015,41(10):1798-1813. (in Chinese)
陈晋音,何辉豪. 基于密度的聚类中心自动确定的混合属性数据聚类算法研究[J]. 自动化学报,2015,41(10):1798-1813.

[10] JIANG L Q,ZHANG M X,ZHENG J L,et al. Optimization research of fast searching and finding density peak clustering algorithm [J]. Application Research of Computers,2016,33(11):3251-3254. (in Chinese)
蒋礼青,张明新,郑金龙,等. 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.

[11] ZHANG R,MA H,LIU Q,et al. An Improved Fast Search Clustering Algorithm Based on Kernel Density[C]// IEEE International Conference on Smart City. Chengdu: IEEE Press,2015:689-693.

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

[13] 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(C):19-40.

[14] BOTEV Z I,GROTOWSKI J F,KROESE D P. Kernel density estimation via diffusion [J]. Annals of Statistics,2010,38(5):2916-2957.

[15] FU L,MEDICO E. FLAME,a novel fuzzy clustering method for the analysis of DNA microarray data [J]. BMC Bioinformatics,2007,8(1):3.

[16] TSAPARAS P,MANNILA H,GIONIS A. Clustering aggregation [J]. ACM Transactions on Knowledge Discovery from Data,2007,1(1):4.

[17] CHANG H,YEUNG D Y. Robust path-based spectral clustering [J]. Pattern Recognition,2008,41(1):191-203.