面向流形数据的测地距离与余弦互逆近邻密度峰值聚类算法

赵 嘉,王 刚,吕 莉,樊棠怀
(南昌工程学院信息工程学院,江西南昌 330099)

摘 要: 密度峰值聚类算法倾向在球形分布数据中选择密度峰值,而流形数据多呈非球形分布,导致不能准确找到数据的类簇中心.该算法的分配策略优先对类簇中心附近的样本进行链式分配,而流形数据大量样本远离其类簇中心,导致本应属于同一类簇的样本被错误分配.为此,本文提出一种面向流形数据的测地距离与余弦互逆近邻密度峰值聚类算法.将K近邻与测地距离结合并重新定义局部密度,凸显密度峰值与非密度峰值的差异,准确找到类簇中心;将互逆近邻和余弦相似性相结合,得到基于余弦互逆近邻的样本相似度矩阵,为流形类簇准确分配样本.实验结果表明,本算法能有效发现流形数据集的几何形状并准确聚类,对真实数据集和图像数据集的聚类效果优秀.

关键词: 密度峰值; 聚类; K近邻; 互逆近邻; 局部密度; 分配策略
中图分类号: TP182;TP391 **文献标识码:** A **文章编号:** 0372-2112(2022)11-2730-08
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20211273

Density Peaks Clustering Algorithm Based on Geodesic Distance and Cosine Mutual Reverse Nearest Neighbors for Manifold Datasets

ZHAO Jia, WANG Gang, LÜ Li, FAN Tang-huai
(*School of Information Engineering*, *Nanchang Institute of Technology*, *Nanchang*, *Jiangxi* 330099, *China*)

Abstract: The density peaks clustering algorithm tends to select the density peaks in the spherical distribution data, while the manifold data are mostly non spherical distribution, resulting in the inability to accurately find the cluster centers. The allocation strategy of the algorithm gives priority to the chain allocation of samples near the cluster centers, while a large number of samples of manifold data are far away from the cluster centers, resulting in the wrong allocation of samples that should belong to the same cluster. Therefore, this paper proposes a density peaks clustering algorithm based on geodesic distance and cosine mutual reverse nearest neighbors for manifold datasets. Combining K-nearest neighbors with geodesic distance and redefining local density, highlighting the difference between density peaks and non density peaks, accurately find the cluster centers; combining the mutual reverse nearest neighbors and cosine similarity, the sample similarity matrix based on cosine mutual reverse nearest neighbors is obtained, which can accurately allocate samples for manifold clusters. The experimental results show that the algorithm can effectively find the geometry structure of manifold datasets, and has excellent clustering effect on real datasets and picture datasets.

Key words: density peaks; clustering; K-nearest neighbors; mutual reverse nearest neighbors; local density; allocation strategy

1 引言

聚类是数据挖掘的一类重要方法,它以最大化类簇内相似性、最小化类簇间相似性的方式对数据分组.作为一种无监督学习方法,聚类可以辅助解释数据分布特征,并为其他数据挖掘方法提供基础支撑,已被广泛应用于自动驾驶、市场研究、文档检索、入侵检测和医学影像等多个领域[1~5].

迄今为止,学者们已经提出了多种类型的聚类算法.基于划分的聚类算法中,K-means算法[6]将样本分成互不重叠的子集,使得所有样本正好归属于一个子

收稿日期:2021-09-17;修回日期:2022-01-01;责任编辑:覃怀银
基金项目:国家自然科学基金(No.52069014, No.62066030);江西省重点研发计划项目(No.20192BBE50076, No.20203BBGL73225)

集. 基于层次的聚类算法通常将样本映射为层次化的树图,切割树图产生期望数量的类簇[7,8]. 基于模型的聚类算法需要构建一种概率分布模型,然后使用EM(Expectation Maximization, EM)算法[9]对该模型进行计算并聚类. 但是,上述的聚类算法对数据分布类型敏感,在没有先验知识的情况下确定算法参数十分困难.

Alex Rodriguez和Alessandro Laio[10]于2014年提出了密度峰值聚类(Density Peaks Clustering, DPC)算法. DPC算法基于两点假设:类簇中心具有较高的局部密度,并且被局部密度较低的邻居包围;类簇中心间具有较大的相对距离. DPC算法能快速寻找到类簇中心并完成样本分配,对不同类型数据具有较好的聚类效果.

流形结构数据普遍存在于人类的生产活动中,如图像分割、网页分析和手写数字识别等[11]. 流形数据有两个本质特征,一方面其几何形状复杂,通常包含一些线条状和圆环状的类簇;另一方面其样本大部分分布于流形骨干网络(核心样本)附近,骨干网络间的连通性是其样本相似性度量的核心要素[12].

DPC及其改进算法当前已在多个领域得到应用,但是在面对流形数据时,依旧存在如下不足:(1) DPC算法过于依赖样本的密度特征[13],而流形数据的样本分布与密度相关性较弱,易导致部分流形类簇没有密度峰值;(2) DPC算法沿梯度下降的方向进行链式样本分配,并未考虑类簇样本的连通性,导致大量样本被分配到错误的流形类簇.

为解决上述问题,本文提出一种面向流形数据的测地距离与余弦互逆近邻密度峰值聚类算法(Density Peaks Clustering algorithm based on Geodesic Distance and Cosine mutual reverse nearest Neighbors for manifold datasets, DPC-GDCN). 本文的主要贡献如下:结合K近邻和测地距离,定义一种基于测地距离的局部密度;融合互逆近邻和余弦相似性,提出一种基于余弦互逆近邻相似度的分配策略;实验结果表明,DPC-GDCN算法在流形数据集、真实数据集和图像数据集上的聚类准确度较高.

2　密度峰值聚类算法

密度峰值聚类算法是一种简洁而高效的基于密度的聚类算法,可以快速寻找到类簇中心,发现多种形状类簇. 密度峰值聚类算法认为:类簇中心附近的样本局部密度较高且不同类簇中心之间相距较远. DPC算法为样本赋予了局部密度ρ_i,同时依据局部密度给出了样本的相对距离δ_i.

DPC算法给出了截断核与高斯核的局部密度计算方式,截断核如式(1)所示,高斯核如式(2)所示:

$$\rho_i=\sum_{i\neq j}\chi\left(d_{ij}-d_c\right),\ \chi(x)=\begin{cases}1, & x<0\\ 0, & x\geq 0\end{cases} \tag{1}$$

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

其中,d_{ij}为样本间的欧氏距离,d_c是样本的截断距离,需预先设定.

取局部密度大于样本i且距其最近点的距离,作为样本i的相对距离δ_i,相对距离δ_i如式(3)所示,把局部密度最高样本的相对距离设定为最大值,如式(4)所示:

$$\delta_i=\min_{j:\rho_j>\rho_i} d_{ij} \tag{3}$$

$$\delta_i=\max_{j} d_{ij} \tag{4}$$

计算所有样本的ρ_i和δ_i,用ρ_i作为横坐标,δ_i作为纵坐标,建立决策图,选取ρ_i和δ_i都较大的点作为类簇中心. 另外,类簇中心也可以通过决策值选取,其定义如式(5)所示:

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

选取类簇中心后,DPC算法依次将剩余点分配给距离其最近的高密度样本所在类簇,完成聚类.

3　DPC-GDCN算法

相较于DPC算法,DPC-GDCN算法主要在局部密度和分配策略两方面进行改进. 首先,结合K近邻和测地距离,定义了一种基于测地距离的局部密度,增强同一类簇样本间的联系,使流形类簇中心更易区分. 其次,融合互逆近邻和余弦相似性,提出了一种基于余弦互逆近邻相似度的分配策略,利用距离和方向信息为流形类簇准确分配样本.

3.1　局部密度

DPC算法采用的欧氏距离只能衡量局部样本相似性,为了体现流形数据的全局结构特性,引入一种基于近邻图的测地距离[11],如式(6)所示:

$$d_G(i,j)=\begin{cases}d(i,j), & j\in \mathrm{KNN}(i)\\ \min\sum_{L=1}^{m-1} d(P_L,P_{L+1}), & j\notin \mathrm{KNN}(i)\end{cases} \tag{6}$$

当样本j为样本i的K近邻时,两样本间的测地距离与欧氏距离相等;当样本j不属于样本i的K近邻时,通过式(6)中第二项计算样本j和样本i的最短路径,得到两样本间的测地距离.

定义1　基于测地距离的局部密度

$$\omega=\frac{1}{N}\sum_{i=1}^{N} d\left(i,\mathrm{Nt}(K)\right) \tag{7}$$

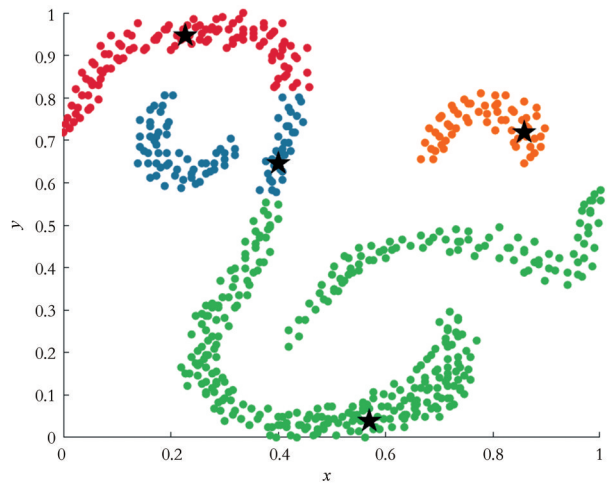
$$\rho(i)=\sum_{j=1}^{N}\exp\left(\frac{-d_{\mathrm{G}}(i,j)^2}{\omega^2}\right)+\frac{1}{N-1}\sum_{v=1}^{N}\exp\left(-d_{\mathrm{G}}(i,v)^2\right) \quad (8)$$

其中,N为样本总数量,ω为离样本i最近的第K个近邻的距离均值. 在新的局部密度中,为了更加突出流行数据的全局结构特征,采用测地距离表征样本间的关联程度;由带宽为ω与带宽为1的两种高斯核计算流形数据的局部密度贡献.

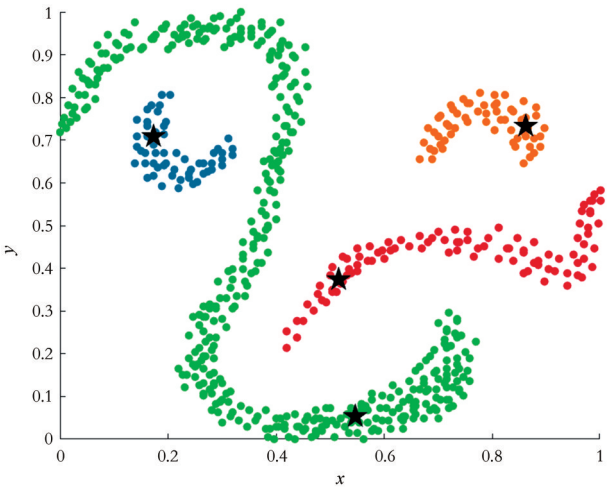
图1为基于截断核和测地距离的局部密度在Db数据集上选择的类簇中心(用五角星表示). 图1中,x、y表示样本的平面位置. 图1(a)的两个流形类簇中没有类簇中心,而最长的"Z"字形流形类簇中出现了三个类簇中心. 如图1(b)所示,本文基于测地距离的局部密度改善了类簇中心判断错误的问题,准确找到四个流形类簇的类簇中心.

3.2 分配策略

DPC算法的分配策略优先对类簇中心附近的样本进行链式分配,而流形数据中大量样本距离其类簇中心较远,标签传播错误会在样本分配过程中持续积累. 因此,本文在考虑样本的空间位置信息的基础上,将互逆近邻和余弦相似性融合到样本分配过程中,提出基于余弦互逆近邻的样本分配策略.



(a) 基于截断核的局部密度



(b) 基于测地距离的局部密度

图1 两种局部密度在Db数据集上选择的类簇中心

定义2 逆近邻 若样本i在样本j的K近邻集合中,则样本j为样本i的逆近邻.

$$\mathrm{RNN}(i)=\left\{i,j\in \mathrm{X}\middle|\, i\in \mathrm{KNN}(j)\right\} \quad (9)$$

定义3 相互逆近邻 若样本i是样本j的逆近邻,样本j同时也是样本i的逆近邻,则将它们称为相互逆近邻.

$$\mathrm{MRNN}(i,j)=\left\{i\in \mathrm{RNN}(j),\, j\in \mathrm{RNN}(i)\right\} \quad (10)$$

定义4 余弦接近度 将样本视作D维空间向量,基于式(11)与(12)计算样本间的余弦接近度.

$$\cos(i,j)=\frac{\sum_{m=1}^{D}\boldsymbol{X}_{im}\boldsymbol{X}_{jm}}{\sqrt{\sum_{m=1}^{D}\boldsymbol{X}_{im}^2}\sqrt{\sum_{m=1}^{D}\boldsymbol{X}_{jm}^2}} \quad (11)$$

$$\varphi(i,j)=\exp\left(-\frac{\cos(i,j)}{K}\right)\cdot\exp\left(-d(i,j)\right) \quad (12)$$

若只以欧氏距离计算样本的相似性,可能有多个样本的欧氏距离相等,欧氏距离一致的样本会降低分配准确度,因此引入夹角余弦以进一步区分样本的相似性.

定义5 余弦互逆近邻相似度 基于相互逆近邻和余弦接近度,得到余弦互逆近邻相似度,如式(13)所示.

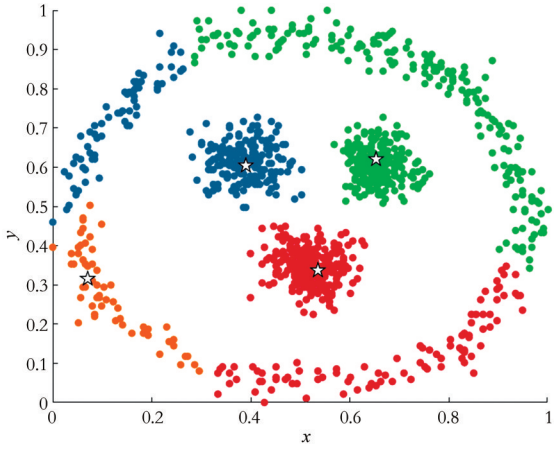
$$\mathrm{SIM}(i,j)=\begin{cases}\sum\limits_{v\in \mathrm{RNN}(j)}\varphi(i,v)+\sum\limits_{u\in \mathrm{RNN}(i)}\varphi(j,u), & i,j\in \mathrm{MRNN}(i,j)\\ 0, & i,j\notin \mathrm{MRNN}(i,j)\end{cases} \quad (13)$$

若样本之间不是互逆近邻,则将其余弦互逆近邻相似度设置为0;若样本之间为互逆近邻,那么分别将其余弦接近度累加得到样本之间的余弦互逆近邻相似度.

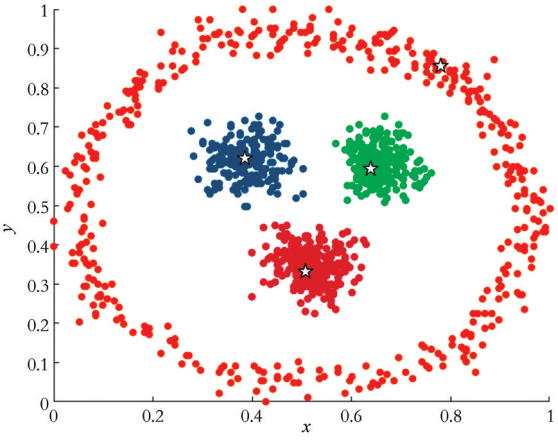
图2为组合截断核与两种分配策略在流形数据集Cth上的对比实验. 图2中,x、y表示样本的平面位置. 图2(a)为采用截断核与DPC算法分配策略的聚类结果,数据集外侧的环状流形类簇被错误切割成四段. 而图2(b)为采用截断核与余弦互逆近邻分配策略的聚类结果,能够正确分配环状流形类簇的所有样本.

3.3 算法步骤

DPC-GDCN算法的执行流程主要包括:计算局部密度和相对距离,寻找类簇中心;构造相似度矩阵,分配核心样本与非核心样本. DPC-GDCN算法的详细步骤如算法1所示.



(*a*)　截断核与DPC算法分配策略组合



(*b*)　截断核与余弦互逆近邻分配策略组合

图2　两种组合在Cth数据集上的聚类结果

算法1　DPC-GDCN

输入:数据集,近邻个数K

输出:聚类结果C

Step1:数据归一化;

Step2:根据式(6)计算测地距离;

Step3:根据式(8)计算样本的局部密度,并根据式(3)、式(4)计算样本的相对距离;

Step4:根据式(5)计算样本的决策值并选取类簇中心;

Step5:根据式(9)~(13)计算样本相似度,得到基于余弦互逆近邻的相似度矩阵;

Step6:在相似度矩阵中找到与已分配样本相似度最大的核心样本,归类到相同类簇;

Step7:当所有已分配样本与未分配样本相似度达到零时,转至Step8,否则,转至Step6再次寻找相似度最大的样本,继续进行分配;

Step8:非核心样本使用DPC算法的分配策略进行分配.

4　实验结果与分析

4.1　实验设置

本文选择两类DPC的改进算法进行对比.第一类是对常规数据集聚类性能优异的算法,包括DPC[10]、FNDPC[14]、DPCSA[15]、FKNN-DPC[16]和IDPC-FA[17]算法;第二类是对流形数据集聚类性能优异的算法,包括DPC-GD[11]和RDPC-DSS[18]算法.本文实验采用的软件环境为Matlab R2020a编程环境.采用调整互信息(Adjusted Mutual Information,AMI)[19]、调整兰德系数(Adjusted Rand Index,ARI)[19]和Fowlkes-Mallows指数(Fowlkes-Mallows Index,FMI)[20]对聚类结果进行评价.AMI、ARI和FMI的指标值越接近1表示聚类结果越好.

4.2　流形和真实数据集实验结果及分析

为验证DPC-GDCN算法的有效性,将8种算法在流形数据集和真实数据集[1]上进行实验.数据集的基本特征如表1所示,表2~9展示了实验结果.表2~4为8种算法在流形数据集上的FMI、AMI、ARI值.表6~8为8种算法在真实数据集上的FMI、AMI、ARI值.表5、表9分别为8种算法在流行数据集和真实数据集上获得最佳聚类效果时的参数取值,表中最优的实验结果用加粗字体表示.

表1　流形和真实数据集的基本特征

数据集	样本规模	样本维度	类簇个数
Db	630	2	4
Jain	373	2	2
Spiral	312	2	3
Pathbased	300	2	3
Lineblobs	266	2	3
Sticks	512	2	4
Cth	1016	2	4
Circle	1897	2	3
Iris	150	4	3
Seeds	210	7	3
Wine	178	13	3
WDBC	569	30	2
Libras	360	90	15
Ecoli	336	8	8
Dermatology	366	33	6
Glass	214	11	39

表2~4中,以Circle数据集为例,对聚类结果进行分析.DPC算法的AMI值略高于0.27,ARI值接近0.06,FMI值略高于0.50,说明DPC算法对Circle数据集的聚类性能较差;FKNN-DPC、DPCSA、FNDPC和RDPC-DSS算法聚类结果依然不佳;而DPC-GDCN和DPC-GD算法能够对Circle数据集准确聚类.综合比

表2 8种算法在流形数据集上的AMI值

数据集	DPC-GDCN	RDPC-DSS	DPC-GD	IDPC-FA	FKNN-DPC	DPCSA	FNDPC	DPC
Db	**1**	0.6675	**1**	0.6526	0.5107	0.4136	0.5098	0.5185
Spiral	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Jain	**1**	0.5398	0.6183	**1**	0.7092	0.2167	0.5961	0.6183
Pathbased	0.9027	0.537	0.5294	0.8442	**0.9305**	0.7073	0.5751	0.5212
Lineblobs	**1**	**1**	**1**	0.8375	**1**	**1**	0.6386	0.7799
Sticks	**1**	**1**	**1**	**1**	**1**	0.7634	0.7634	0.8094
Cth	**1**	0.6555	**1**	0.8482	**1**	0.7891	0.8758	0.682
Circle	**1**	0.7788	**1**	0.1927	0.7063	0.295	0.4236	0.2747

表3 8种算法在流形数据集上的ARI值

数据集	DPC-GDCN	RDPC-DSS	DPC-GD	IDPC-FA	FKNN-DPC	DPCSA	FNDPC	DPC
Db	**1**	0.5368	**1**	0.5033	0.2718	0.1096	0.2714	0.2794
Spiral	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Jain	**1**	0.6728	0.7146	**1**	0.8224	0.0442	0.7257	0.7146
Pathbased	0.9292	0.5329	0.4797	0.8593	**0.9499**	0.6133	0.5067	0.4717
Lineblobs	**1**	**1**	**1**	0.8237	**1**	**1**	0.5769	0.721
Sticks	**1**	**1**	**1**	**1**	**1**	0.636	0.639	0.7534
Cth	**1**	0.671	**1**	0.7729	**1**	0.6538	0.8327	0.5017
Circle	**1**	0.8497	**1**	0.1319	0.6139	0.0833	0.2732	0.0554

表4 8种算法在流形数据集上的FMI值

数据集	DPC-GDCN	RDPC-DSS	DPC-GD	IDPC-FA	FKNN-DPC	DPCSA	FNDPC	DPC
Db	**1**	0.725	**1**	0.6999	0.5793	0.4689	0.5803	0.5853
Spiral	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Jain	**1**	0.8896	0.8819	**1**	0.9359	0.5924	0.9051	0.8819
Pathbased	0.9529	0.7538	0.6703	0.9067	**0.9665**	0.7511	0.7065	0.6664
Lineblobs	**1**	**1**	**1**	0.8842	**1**	**1**	0.7218	0.8166
Sticks	**1**	**1**	**1**	**1**	**1**	0.7443	0.7467	0.8235
Cth	**1**	0.7868	**1**	0.835	**1**	0.7547	0.8786	0.6397
Circle	**1**	0.9152	**1**	0.4857	0.779	0.5242	0.5863	0.5005

表5 8种算法在流形数据集上获得最优聚类结果的参数取值

数据集	DPC-GDCN	RDPC-DSS	DPC-GD	IDPC-FA	FKNN-DPC	DPCSA	FNDPC	DPC
Db	6	-	6/2.0	-	19	-	0.09	4
Spiral	11	-	10/2.0	-	6	-	0.07	1.8
Jain	27	-	10/0.5	-	43	-	0.47	0.8
Pathbased	7	-	6/6.0	-	9	-	0.01	3.8
Lineblobs	7	-	10/6.0	-	12	-	0.21	3.7
Sticks	15	-	10/2.0	-	6	-	0.16	2.1
Cth	10	-	10/2.0	-	22	-	0.45	1.1
Circle	13	-	10/2.0	-	32	-	0.29	3.3

较所有算法在8个流形数据集的聚类结果可以发现：DPC-GDCN算法Db、Jain、Spiral、Pathbased、Lineblobs、Sticks、Cth和Circle数据集都取得了最佳的聚类结果，说明DPC-GDCN算法在流形数据集上聚类效果十分优秀.

由表6~8可知，鸢尾花卉数据集Iris上，RDPC-DSS、FKNN-DPC、DPCSA和FNDPC算法的聚类效果最佳，而DPC-GDCN与IDPC-FA算法的聚类结果略低于前4种算法，DPC-GD和DPC算法的聚类效果最差. 红酒数据集Wine上，DPC-GDCN算法虽然未取得最好聚类效果，但是与FKNN-DPC算法非常接近. 大肠杆菌数据集Ecoli上，IDPC-FA算法取得较好的聚类效果，DPC-

表6　8种算法在真实数据集上的AMI值

数据集	DPC-GDCN	RDPC-DSS	DPC-GD	IDPC-FA	FKNN-DPC	DPCSA	FNDPC	DPC
Iris	0.8828	**0.8831**	0.7277	0.8623	**0.8831**	**0.8831**	**0.8831**	0.7247
Seeds	**0.7441**	0.5452	0.7172	0.7299	0.7118	0.6609	0.7136	0.7298
Wine	0.8210	0.6455	0.7575	0.7675	**0.8481**	0.748	0.7898	0.7065
WDBC	**0.6575**	0.1345	0.6518	0.6237	0.6423	0.3361	0.6076	0.4366
Libras	**0.6171**	0.4367	0.5609	0.5733	0.5554	0.5388	0.5494	0.5358
Ecoli	0.5236	0.5603	0.5103	**0.6638**	0.5878	0.4406	0.4833	0.4978
Dermatology	**0.9401**	0.7468	0.8029	0.8638	0.8066	0.7451	0.7898	0.7615
Glass	**0.5345**	0.2666	0.4337	0.4511	0.4308	0.2404	0.4701	0.5260

表7　8种算法在真实数据集上的ARI值

数据集	DPC-GDCN	RDPC-DSS	DPC-GD	IDPC-FA	FKNN-DPC	DPCSA	FNDPC	DPC
Iris	0.886	**0.9038**	0.6634	0.8857	**0.9038**	**0.9038**	**0.9038**	0.7037
Seeds	**0.788**	0.555	0.7341	0.767	0.7303	0.6873	0.7545	0.767
Wine	0.8516	0.6332	0.7562	0.7713	**0.8839**	0.7414	0.8025	0.6724
WDBC	**0.7736**	0.1634	0.7607	0.7423	0.7613	0.3771	0.7305	0.4964
Libras	**0.4187**	0.2428	0.4174	0.3816	0.3459	0.3095	0.329	0.3193
Ecoli	0.5929	0.5551	0.4551	**0.7561**	0.5894	0.4593	0.5618	0.4465
Dermatology	**0.9437**	0.7376	0.784	0.8772	0.8361	0.6062	0.7995	0.6923
Glass	0.4705	0.2316	0.3573	0.4049	0.437	0.1982	**0.4711**	0.3562

表8　8种算法在真实数据集上的FMI值

数据集	DPC-GDCN	RDPC-DSS	DPC-GD	IDPC-FA	FKNN-DPC	DPCSA	FNDPC	DPC
Iris	0.9237	**0.9355**	0.7824	0.9233	**0.9355**	**0.9355**	**0.9355**	0.8032
Seeds	**0.8581**	0.7101	0.8231	0.8444	0.8029	0.7918	0.8361	0.8444
Wine	0.9013	0.7672	0.838	0.8478	**0.9229**	0.8283	0.8686	0.7835
WDBC	**0.8956**	0.7164	0.8914	0.8829	0.8894	0.7595	0.8758	0.7941
Libras	**0.4654**	0.3397	0.4581	0.4247	0.4044	0.3791	0.3869	0.3717
Ecoli	0.7198	0.6724	0.5843	**0.8284**	0.7027	0.6467	0.7178	0.5775
Dermatology	**0.955**	0.8077	0.8311	0.9018	0.8709	0.6896	0.8418	0.7545
Glass	**0.6055**	0.5554	0.5403	0.5552	0.5924	0.5448	0.5973	0.5267

表9　8种算法在真实数据集上获得最优聚类结果的参数取值

数据集	DPC-GDCN	RDPC-DSS	DPC-GD	IDPC-FA	FKNN-DPC	DPCSA	FNDPC	DPC
Iris	21	-	8/2.0	-	22	-	0.11	0.2
Seeds	5	-	10/2.0	-	4	-	0.07	0.7
Wine	24	-	10/6.0	-	8	-	0.26	4
WDBC	11	-	6/6.0	-	2	-	0.05	0.5
Libras	8	-	10/2.0	-	10	-	0.17	0.3
Ecoli	30	-	10/0.5	-	2	-	0.35	0.4
Dermatology	7	-	6/6.0	-	35	-	0.17	1.6
Glass	20	-	6/0.2	-	14	-	0.16	4

GDCN和FKNN-DPC算法的聚类结果略逊于前者. 玻璃数据集Glass上,DPC-GDCN算法的AMI和FMI值最优,而ARI值略低于FNDPC算法. 小麦种子数据集Seeds、乳腺癌数据集WDBC、手势数据集Libras、皮肤数据集Dermatology上,DPC-GDCN算法的聚类效果好于其他7种算法.

4.3　流形特征图像数据集实验结果及分析

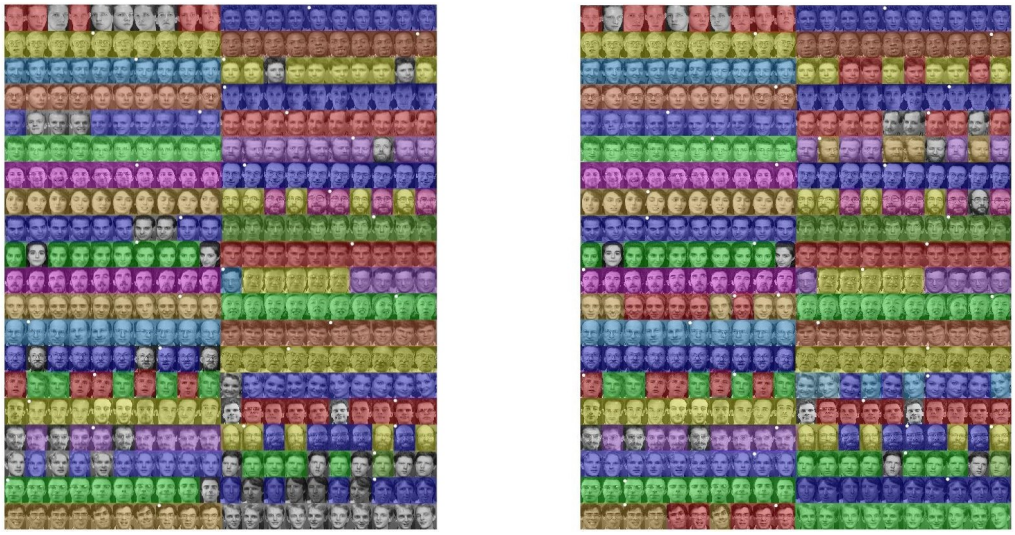
为验证DPC-GDCN算法在流形特征图像数据集中的聚类效果,8种算法在Olivetti Faces数据集[1]中进行实验. Olivetti Faces数据集是机器学习领域广泛使用的一个数据集,其中包含40个人的脸部图像,分别从10个角度拍摄,共计400张人脸图像. 由表10可知,DPC-

GDCN算法在人脸数据集中的AMI、ARI和FMI值高于其他算法.

由于8种算法对Olivetti Faces数据集的聚类结果较为接近,本文选择DPC-GDCN与DPC算法在图像数据集的聚类结果上进行分析.如图3所示,40个人的脸部图像中,左侧从上到下分别为第1~20号,右侧为第21~40号.从图3(a)可以发现,第1、5、9、10、14、17、18、23、36、38、39和40号的图像中,每个人至少有2张人脸未被DPC算法识别,说明DPC算法在复杂数据中寻找类簇中心具有一定困难,即使算法在正确选择类簇中心后,依然会错误分配样本.从图3(b)发现,相较DPC算法,DPC-GDCN算法成功识别出第5、9、14、18、39和40号的10张人脸,提高了人脸识别准确度.由表10和图3可以发现,相较DPC算法,DPC-GDCN算法在面对图像数据集时,能够克服其复杂的数据分布特征,得到理想的聚类结果.

表10 8种算法在Olivetti Faces数据集的聚类结果

算法	AMI	ARI	FMI	参数值
DPC-GDCN	**0.8308**	**0.7035**	**0.7159**	4
RDPC-DSS	0.6612	0.4354	0.4845	-
DPC-GD	0.8260	0.7029	0.7132	6/1.0
IDPC-FA	0.8093	0.6924	0.7029	-
FKNN-DPC	0.7989	0.6784	0.6885	5
DPCSA	0.8254	0.7025	0.7127	-
FNDPC	0.7965	0.6652	0.6778	0.44
DPC	0.7891	0.6549	0.6701	3.0



(a) DPC算法 (b) DPC-GDCN算法

图3 两种算法在Olivetti Faces数据集的聚类结果

5 结论

针对DPC算法聚类流形数据易误判类簇中心和类簇样本错误分配的问题,本文提出了一种面向流形数据的测地距离与余弦互逆近邻密度峰值聚类算法.实验结果表明,DPC-GDCN算法有效改善了DPC算法对流形类簇易误判类簇中心和样本易错误分配的问题,对流形数据集的聚类准确度高,并且在真实数据集和图像数据集上的聚类效果优于对比算法.

参考文献

[1] 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.

[2] DUAN X Y, LIU Y N, WANG X B. SDN enabled 5G-VANET: Adaptive vehicle clustering and beam formed transmission for aggregated traffic[J]. IEEE Communications Magazine, 2017, 55(7): 120-127.

[3] YOUCEF D, ASMA B, PHILIPPE F V, et al. Fast and effective cluster based information retrieval using frequent closed itemsets[J]. Information Sciences, 2018, 453: 154-167.

[4] CARCILLO F, BORGNE Y A L, CAELEN O, et al. Combining unsupervised and supervised learning in credit card fraud detection[J]. Information Sciences, 2021, 557: 317-331.

[5] HU Z L, TANG J S, WANG Z M, et al. Deep learning for image based cancer detection and diagnosis-A survey[J]. Pattern Recognition, 2018, 83: 134-149.

[6] ZHAO W L, DENG C H, NGO C W. *K*-means: a revisit[J]. Neurocomputing, 2018, 291: 195-206.
[7] KARYPIS G, HAN E H, KUMAR V. Chameleon: hierarchical clustering using dynamic modeling[J]. Computer, 1999, 32(8): 68-75.
[8] GUHA S, RASTOGI R, SHIM K. CURE: An efficient clustering algorithm for large databases[J]. Information Systems, 2001, 26(1): 35-58.
[9] DEMPSTER A P, LAIRD N M, RUBIN D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the Royal Statistical Society, 1977, 39(1): 1-22.
[10] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.
[11] DU M J, DING S F, XU X, et al. Density peaks clustering using geodesic distances[J]. International Journal of Machine Learning and Cybernetics, 2017, 9(8): 1335-1349.
[12] SUN L L, CHEN G Q, XIONG H, et al. Cluster analysis in data-driven management and decisions[J]. Journal of Management Science and Engineering, 2017, 2(4): 227-251.
[13] CHENG Y Z. Mean shift, mode seeking, and clustering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(8): 790-799.
[14] DU M J, DING S F, XUE Y. A robust density peaks clustering algorithm using fuzzy neighborhood[J]. International Journal of Machine Learning and Cybernetics, 2018, 9 (7): 1131-1140.
[15] YU D H, LIU G J, GUO M Z, et al. Density peaks clustering based on weighted local density sequence and nearest neighbor assignment[J]. IEEE Access, 2019, 7: 34301-34317.
[16] 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.
[17] ZHAO J, TANG J J, SHI A Y, et al. Improved density peaks clustering based on firefly algorithm[J]. International Journal of Bio-Inspired Computation, 2020, 15(1): 24-42.
[18] XU X, DING S F, WANG L J, et al. A robust density peaks clustering algorithm with density-sensitive similarity[J]. Knowledge-Based Systems, 2020, 200: 106028
[19] NGUYEN X V, JULIEN E, JAMES B. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance[J]. Journal of Machine Learning Research, 2010, 11(1): 2837-2854.
[20] 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.

作者简介

赵　嘉　男, 1981年9月出生于江西省九江市. 现为南昌工程学院教授、硕士生导师. 主要研究方向为机器学习、数据挖掘和智能计算.
E-mail: zhaojia925@163.com

王　刚　男, 1995年8月出生于江西省赣州市. 现为南昌工程学院在读硕士研究生. 主要研究方向为机器学习和数据挖掘.
E-mail: wang691630202@163.com

吕　莉　女, 1982年5月出生于江西省贵溪市. 现为南昌工程学院教授、硕士生导师. 主要研究方向为大数据分析和目标跟踪.
E-mail: lvli@nit.edu.cn

樊棠怀　男, 1962年11月出生于江西省九江市. 现为南昌工程学教授、硕士生导师. 主要研究方向为传感器信息获取与处理、机器学习和数据挖掘.
E-mail: fantanghuai@163.com