面向地理定位大数据的新型密度山峰聚类算法

万静意，软件工程6班，20164872
杜语嘉，软件工程6班，20161752

An Efficient Clustering Algorithm by Fast Search and Find of Density Peaks for Location Big Data

摘要：

本文通过实现和改进在《Science》杂志上发表的一种新型快速密度山峰聚类算法[1],与现有的常用聚类方法进行分析比较，发现其优点和不足的地方，并实际应用于地理定位大数据的分析中,以快速发现位置数据中任意形状的聚类簇模式和噪声。首先，基于类簇中心被具有较低局部密度的邻居点包围，且与具有更高密度的任何点有相对较大的距离这样的假设提出基本算法。其次，通过 Python 实现该算法并采用基本聚类测试数据集进行效果测试和算法评估分析。最后，用在基准测试数据上验证了所提算法的聚类效果,以及在位置大数据上的实验结果统计和原先普遍所采用的的方法，如基于密度的方法 DBSCAN[2]、基于划分的方法 K-means[3]、基于网格的方法（如 GDCA[4]和 GG[5]）基于路径的谱聚类算法[6]等进行比较。

一．引言

聚类分析旨在依赖所要聚类元素之间的相似度以期将他们分成不同的类。该方法的应用领域范围包括航空航天、生物化学、文献计量学、模式识别等。基于聚类中心具有比相邻点密度更高以及它们之间具有相当远的距离的特点,作者依此提出了一种新的聚类方法。该种方法最大的突破就是跳出了 K-means[3]方法的桎梏，采用了一种较新的思想，且群的数目直观可见、噪声点自动标出并排除在分析之外、群的区分不依赖于它们的形状和所处的空间维度。

随着智能手机和各种智能终端等位置感知定位追踪设备的广泛普及，各类基于 GPS、北斗卫星导航系统或者其它定位服务的应用层出不穷，早已在生活、商业、交通、医疗、军事等方面广泛应用，如用户签到数据、车辆 GPS 轨迹、带有位置的图片、微博等海量的位置大数据被实时采集,而基于这些位置大数据的服务俨然已成为一大独具特色的新兴产业,尤其在交通调度与控制、推荐系统、广告投放、道路规划、公共设施选址与评估、商业决策以及恐怖主义分析等领域有着巨大应用价值.由于这些位置数据采集越来越便利,数据量正以爆炸式趋势增长.因此,如何有效存储、快速挖掘出有意义的潜在模式成为了快速处理位置大

数据的主要挑战。海量位置数据的分析可以定量描述和估计人们的社会活动特征,发现人们在不同时空粒度下的行为规律,洞察群体整体移动趋势,识别人们感兴趣的路线和区域等,因此,通过对位置数据挖掘,可帮助我们理解和发现位置大数据中所隐含的巨大价值[7]. 例如 Zheng 等[8]通过对用户的 GPS 轨迹挖掘来发现有关联的兴趣点模式,进而发现受欢迎的旅游路线;Zheng 等[8]还对个人历史位置数据挖掘,实现对用户的好友推荐和景点推荐.在位置大数据分析处理应用中,聚类技术常常被用于对位置大数据进行预处理分析,以发现位置数据中的空间或时空分簇模式[9][10].

本文针对位置大数据的高效密度聚类问题,从密度聚类算法内在特性优化视角,实现和改进了一种新型的快速查找密度山峰的聚类算法,可在一般计算平台实现堆位置数据的密度蕨类任务。最后,在多个基准测试数据和大规模位置大数据上进行了综合实验评估,验证了所提算法的聚类效果和性能,并与常用聚类方法进行了对比分析。

二. 算法描述

1. 基本思想

该算法基于以下假设:簇的中心被局部密度更小的点包围而且这些中心距离局部密度比它们大的点相当远。即聚类中心同时具有以下两个特点:

- 本身的密度大,即它被密度均不超过它的邻居包围;
- 与其它密度更大的数据点之间的“距离”相对更大;

对于每一个数据点,我们计算两个数值,:局部密度P_i以及它与更高密度点的距离δ_i.这两个值只依赖与数据点之间的距离,我们假定该距离满足三角不等式。我们将采用 Cut-off kernel 和 Gaussian kernel 两种计算方式数据点的局部密度定义为:

- Cut-off kernel

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{1.1}$$

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

其中,d_c为截断距离,ρ_i为到点i的距离比d_c小的点的数目。该算法支持在不同的点出P_i的相对大小敏感。这就意味着对于大型数据集,该分析的结果对于d_c的选择鲁棒性很强。

δ_i是点i到与比它密度高的点的最小距离:

$$\delta_i = \min_{j:\ pj>pi} (d_{ij}) \tag{2}$$

对于密度最高的点,我们一般记为$\delta_i = max_j(d_{ij})$。我们注意到局部或者全局密度最高的点的δ_i会比周围的邻居点大。因此,那些δ_i的值异常答的点被视作簇的

中心。

2 聚类方法

2.1 找出聚类中心

如图 1[11]所示，所有点的密度值按照由高到低排列，“1”表示密度最高的点。B 图中画图每个点和的函数关系，从中可以看出“9”和“10”号点拥有相近的密度值但是其δ_i不同，这里“9”属于“1”号类别。“26”，“27”和“28”号点有一个相对较大的δ_i，但是其ρ_i太小，这主要是因为它们是孤立点。

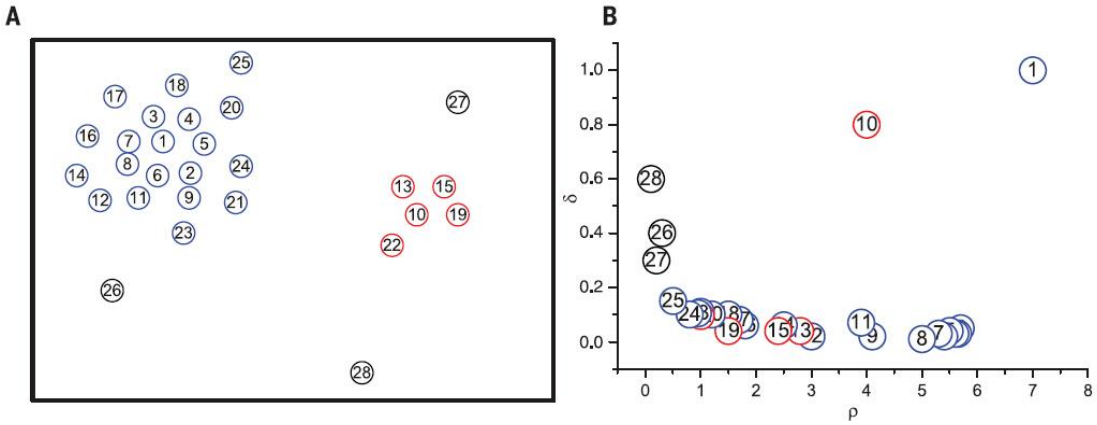


图 1 决策图（decision graph）的实例及示意图

通过生成的 Decision-Graph(delta-rho)图，根据前面的结论，δ_i和P_i值均比其它大的点就是我们要找的聚类中心，然后根据此决策图给定δ_{min}和p_{min}，筛选出同时满足$(\rho_i > \rho_{min})$和$(\delta_i > \delta_{min})$条件的点作为聚类中心点。

确定聚类中心$\{m_j\}_{j=1}^{n_c}$，并初始化数据点归类属性标记$\{C_i\}\}_{i=1}^{N}$，具体为：

$$C_i \begin{cases} k & \text{若}X_i\text{为聚类中心，且归属于第 k 个 cluster} \\ -1 & \text{otherwise} \end{cases} \quad (3)$$

对于那些在决策图中无法用肉眼判断出聚类中心的情形，作者在文中给出了一种确定聚类中心个数的提示：计算一个将ρ值和δ值综合考虑的量

$$\gamma_i = \rho_i \delta_i, i \in I_s \quad (4)$$

显然，γ值越大，越有可能是聚类中心。因此，只需对$\{\gamma_i\}_{i=1}^{N}$进行降序排列，然后从前往后截取若干个数据点作为聚类中心。

将排序后的γ在坐标平面画出来，如图 2 所示。由此可见 L 非聚类中心的γ值比较平滑，而从非聚类中心过渡到聚类中心时，γ值有一个明显的跳跃、这个跳跃用肉眼或数值检测都可以判断出来。作者还提到，对于人工随机生成的数据集，γ的分布还满足幂次定律，即log γ近似呈直线，且斜率依赖于数据维度。

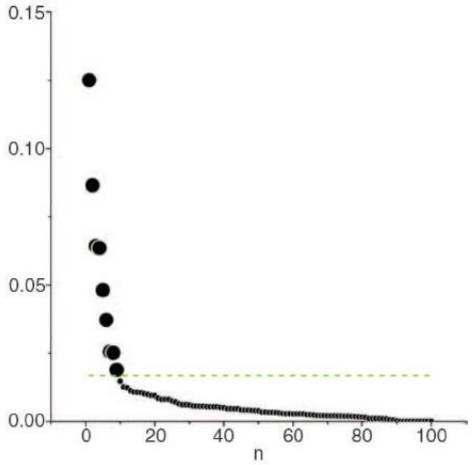


图 2 降序排列的γ值示意图

2.2 剩余点的类别指派：

当聚类中心确定之后，剩下的非聚类中心点的类别标签指定按照以下原则：

当前点的类别标签等于高于当前点密度的最近的点的标签一致。从而对所有点的类别进行了指定。如图 3[11]所示，编号表示密度高低，“1”表示密度最高，以此类推。“1”和“2”均为聚类中心，"3"号点的类别标签应该为与距离其最近的密度高于其的点一致，因此“3”号点属于聚类中心 1，由于“4”号点最近的密度比其高的点为“3”号点，因此其类别标签与”3“号相同，也为聚类中心 1。

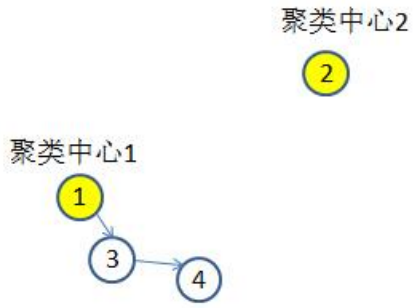


图 3 非聚类中心类别指派示意图

3. 去除噪声点

2.3 类别间边界确定

一个簇中的数据点可以分为簇核心部分和簇的光环（作为噪声点）。在对每一个点指派所属类别之后，这里文章没有人为直接用噪音信号截断的方法去除噪音点，而是先算出类别之间的边界，边界区域由这样的数据点构成：它们本身属于该簇，但在与其距离不超过d_c的范围内，存在属于其它簇的数据点。

2.4 划分出簇的光环

利用边界区域，这个簇就可以计算出一个平均局部密度ρ_b，密度值小于该

值的点则被分为光环部分，即噪声点。如图 4 所示，橙色圈内的为簇核心部分，外的数据点为光环。

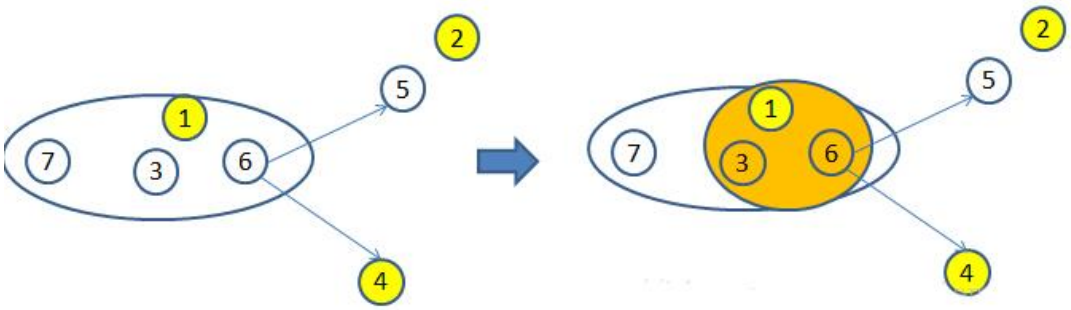


图 4 噪声点的去除

4. 算法改进和优化

4.1 局部密度定义

在原文中局部密度P_i使用的是 Cur-off kernel 的计算方式，在实际代码实现中我们采用的是 Gaussian kernel 的计算方式

- Gaussian kernel

$$P_i = \sum_{j \epsilon I_S \backslash \{i\}} e^{-(\frac{d_{ij}}{d_c})^2} \quad (1.2)$$

对比定义（1.1）和（1.2）易知，Cut-off kernel 为离散值，Gaussian kernel 为连续值，因此，相对来说。后者产生冲突（即不同的数据点具有相同的局部密度值）的概率更小。此外，对于（1.2）仍满足原定义。

4.2 聚类中心个数的确定

利用ρ和δ定义更合适的γ函数，鉴于这两个值可能处于不同的数量级。因此，首先对做一次归一化，都归一到[0,1]区间，然后定义γ=$\rho \times \delta$，画出曲线图，非聚类中心和聚类中心的点会产生一个突然的跳跃，这样就可自动判断出聚类中心，测试结果如图 5 所示：

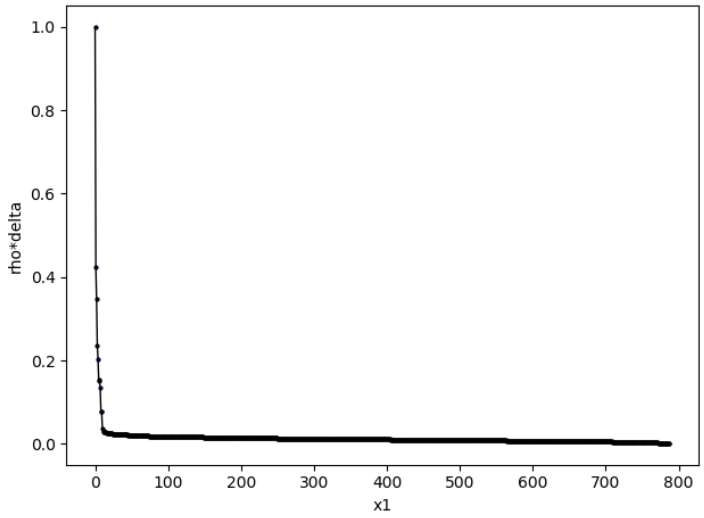


图 5 γ曲线图

4.3 噪声点的去除

光晕（噪声点）即指的是ρ较大而δ较小的点，通过γ曲线图即可以判断出噪声点，效果如图 6 所示：

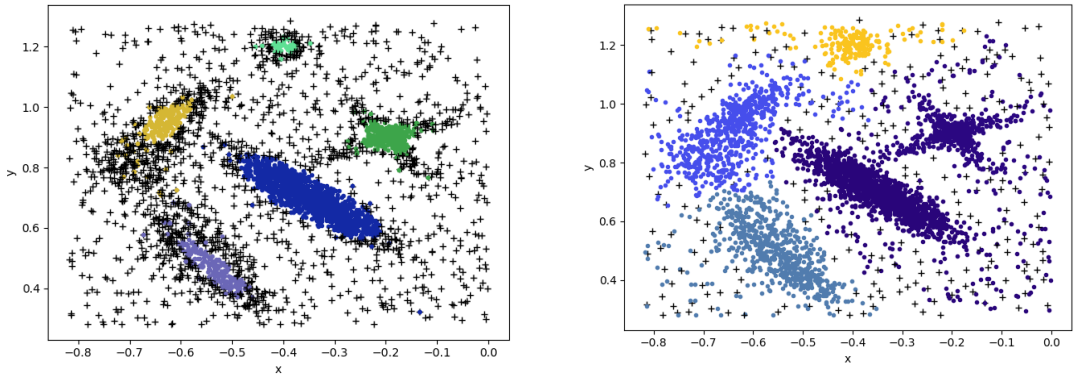


图 6 噪声点的去除

4.4 距离计算方式

在作者提供的 matlab 源代码实现过程中计算距离采用的是欧氏距离的计算方式，由于所采用的的训练测试数据集为二维、三维或者通过降维处理之后的数据，因此采用欧氏距离没有很大的问题。但是如果将该算法应用到实际中，例如图像、音视频识别等具有更高维的数据时欧氏距离将不再使用。

三．实验评估与分析

本节对新型密度聚类算法（Clustering by fast search and find of density peaks）的聚类效果及其在相应数据库的性能表现，并与常用的聚类算法 K-Means[3]、DBSCAN[2]和谱聚类算法[6]进行了对比分析，所有算法均由 Python 实现。

1. 实验数据和测试方法

1.1 实验数据

对四种算法模型进行对比分析所采用的数据集共有 10 个，其中有人工数据集也有真实数据，包含 7 个带标签可用于参数评估的数据集和 3 个不带标签的数据集。所选的 10 个数据集十分具有代表性，涵盖了相连、内嵌、旋转、不同密度重叠和不同形状的情况，有一些是公认的分析聚类效果的常用标准数据集。

其中，7 个带标签的 Aggregation[13]包括了 7 个飞告诉分布得聚类簇；Compound[14]包含了 6 个不同形状的复杂簇结构；D31[15]由 31 个高位密度簇组成； Flame[16]包含了两个相连的密度簇结构； Pathbased[17]包括了一个环形簇和两个内嵌的高斯密度簇；R15[18]由 15 个大小相似且相互重叠的的高位密度簇组成； Spiral[19]由三个相互缠绕旋转的条状密度簇构成。

不带标签的 3 个数据集： t4.8k[20]由 7 个不同形状、互相嵌套且惨掺杂了大量噪声点的密度簇构；panelB[21]由 5 个不同形状且带大量噪声点的密度簇组成； MopsiLocations2012-Joensuu[22]是由 Mopsi 提供的用户位置分布数据。

表 1 实验数据描述

Datasets	# of points	Dimension	Classes
Aggreagtion	788	2	7
Compound	399	2	6
D31	3100	2	31
Flame	240	2	2
Pathbased	300	2	4
R15	600	2	15
Spiral	312	2	3
t4.8k	8000	2	Unknown
panelB	4000	2	Unknown
MopsiLocations2012-Joensuu	6014	2	Unknown

1.2 测试方法

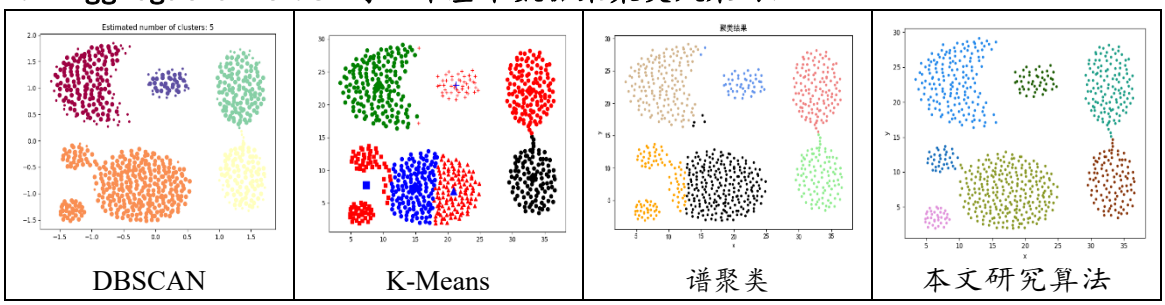
为验证算法的有效性，处于相同实验环境下，在 Aggregation-t4.8k 等 7 个基准数据集上测试了 DBSCAN、谱聚类、k-means 和 Clustering by fast search and find of density peakS 方法的聚类效果，评估方法采用与本文主要实现研究的新型密度聚类算法聚类结果相比较的方式。

最终参与对比评估的聚类效果采用的是在多次实验中效果在最佳的一次，即和数据集标签类别最相近的。为验证算法的高效性和实际应用中的性能，还在 MopsiLocations2012-Joensuu 等数据集上测量了所提方法，评估了模型的效率和对输入参数的敏感性。

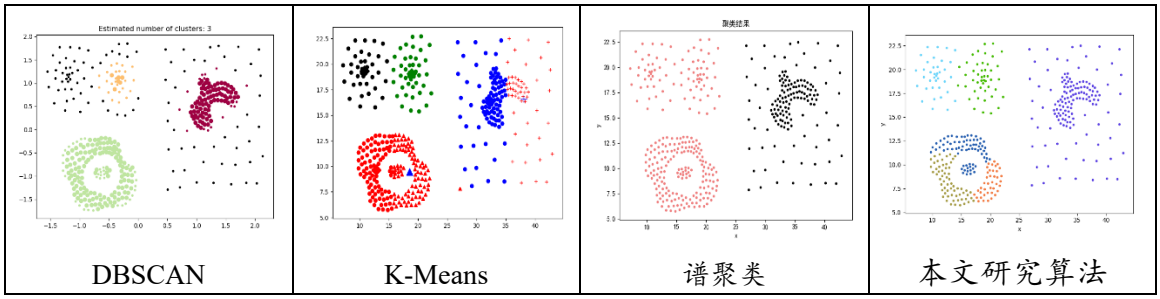
2. 各算法效果评估

在数据集上分别运行各个算法对比各个算法聚类的效果。各类算法参数阈值的选择遵循以下优先原则：(1) 遵循原来算法默认最优值；(2) 尽量采用相同参数；(3) 多次试验聚类效果最优的参数值。

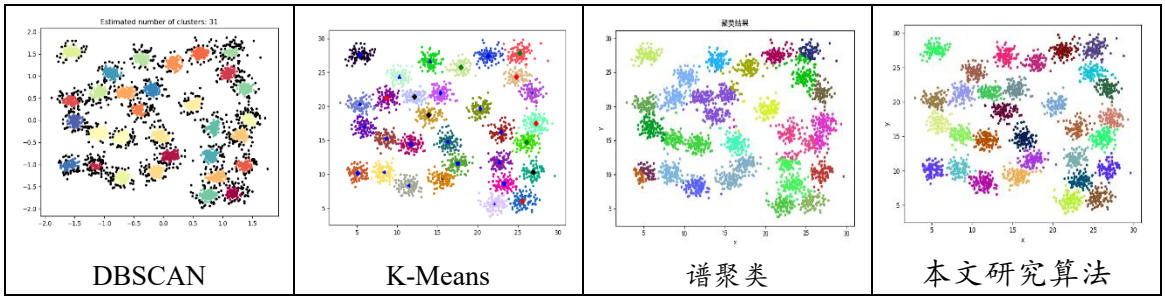
2.1 Aggregation-t4.8k 等 7 个基准数据集聚类效果对比



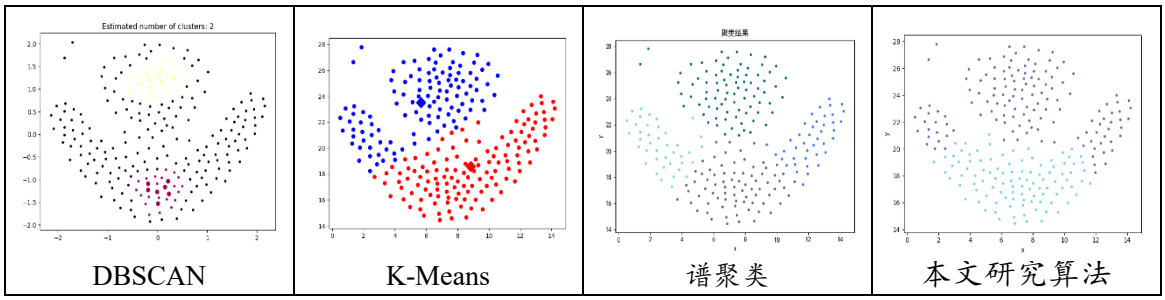
（a）Aggregation



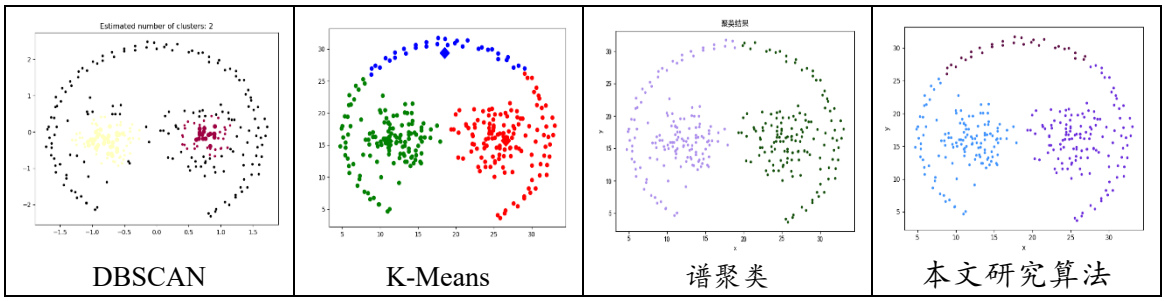
（b）Compound



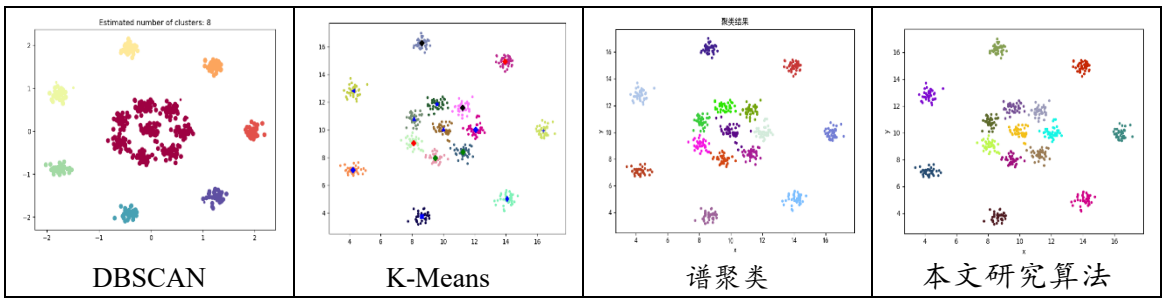
（c）D31



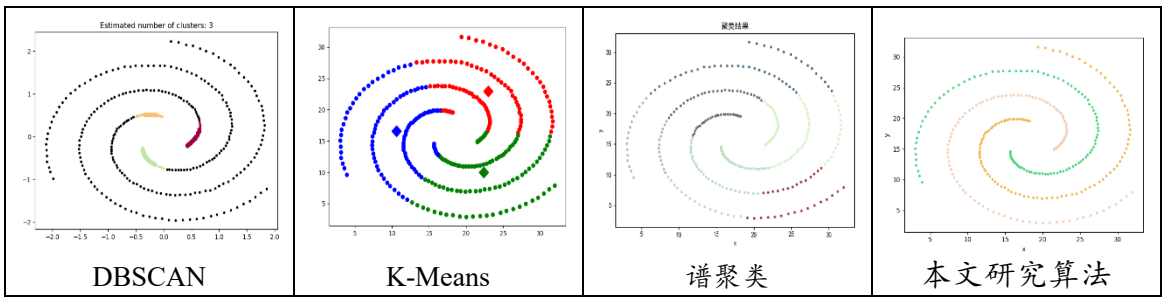
（d）Flame



（e）Pathbased

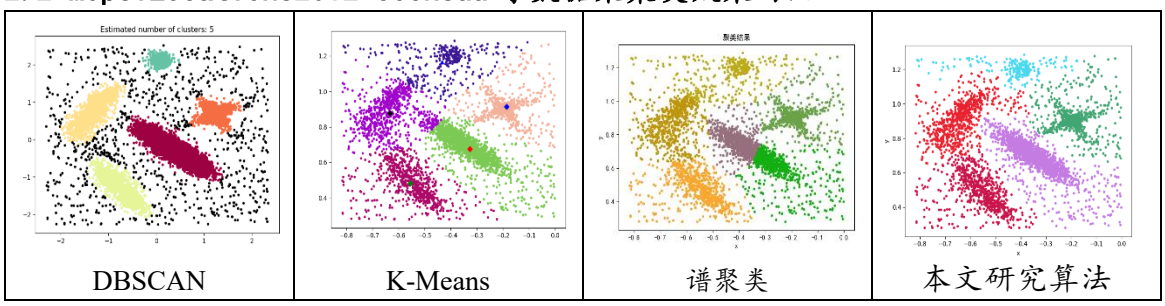


（f）R15

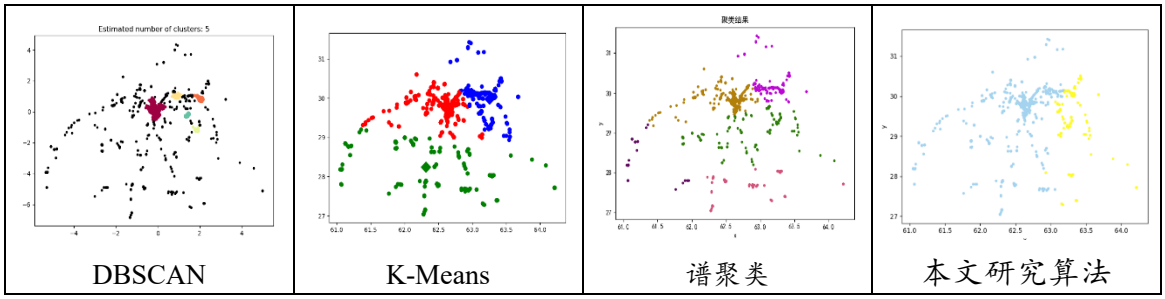


（g）Spiral

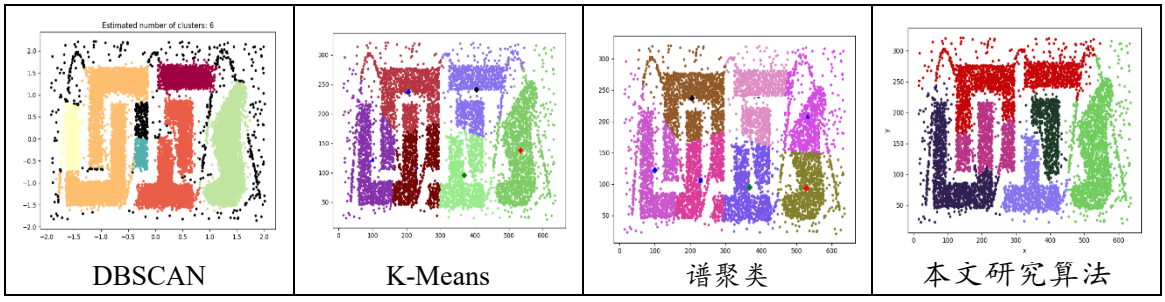
2.2 MopsiLocations2012-Joensuu 等数据集聚类效果对比



（h）panelB



（i）MopsiLocations2012-Joensuu



（j）t4.8k

2.3 模型评估

2.3.1 分群质量评估指标

对聚类效果的评估没有固定的标准，本文采用的是先使用 sklearn 库中提供

的一些常用指标评价标准，然后在不同数据集下交叉验证不同群的分群指标。

（1）同质性：每个群集只包含单个类的成员

（2）完整性：给定类的所有成员都分配给同一个群集。

（3）V-measure：调和平均。

（4）调整兰德指数：整兰德系数假设模型的超分布为随机模型，即 UU 和 VV 的划分为随机的，那么各类别和各簇的数据点数目是固定的。调整的兰德系数为：

$$\mathrm{ARI} = \frac{RI - E(RI)}{\max(\mathrm{RI}) - \mathrm{E}(\mathrm{RI})}$$

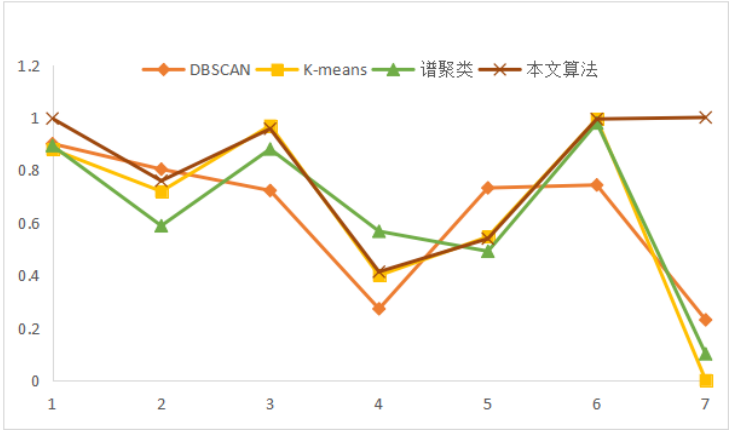
（5）轮廓系数：轮廓系数（Silhouette coefficient）适用于实际类别信息未知的情况。对于单个样本，设 a 是与它同类别中其他样本的平均距离，b 是与它距离最近不同类别中样本的平均距离，轮廓系数为：

$$s = \frac{b - a}{max(a, b)}$$

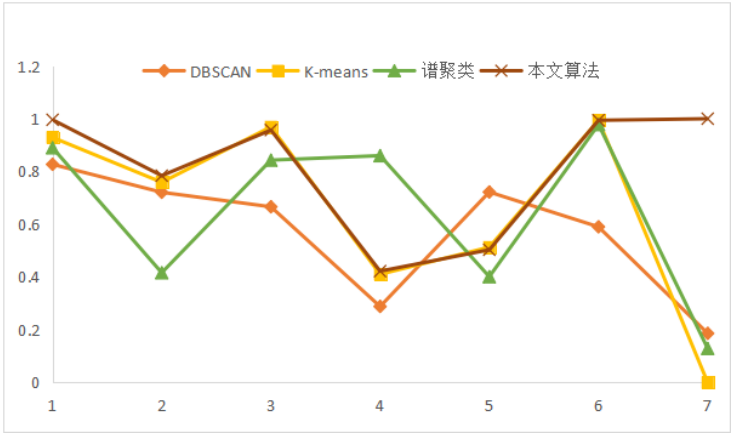
对于一个样本集合，它的轮廓系数是所有样本轮廓系数的平均值。轮廓系数取值范围是$[-1, 1]$，同类别样本越距离相近且不同类别样本距离越远，分数越高。

2.3.2 分析对比

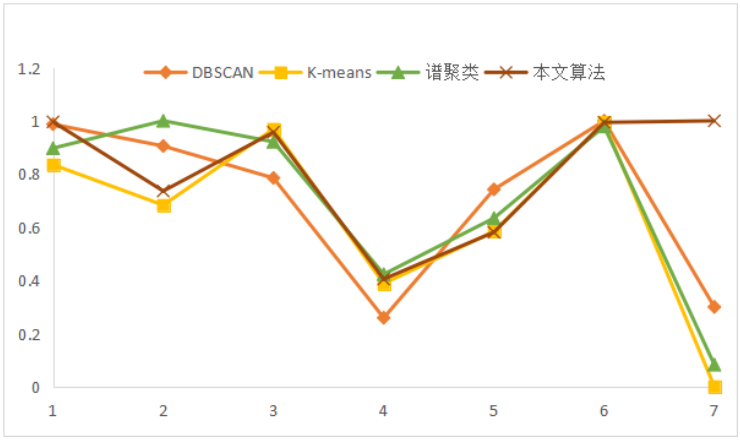
如下图所示，为 DBSCAN, K-means，谱聚类和本文研究算法在 Aggregation 等 7 个有标签的数据集下的各个评估指标对应值。



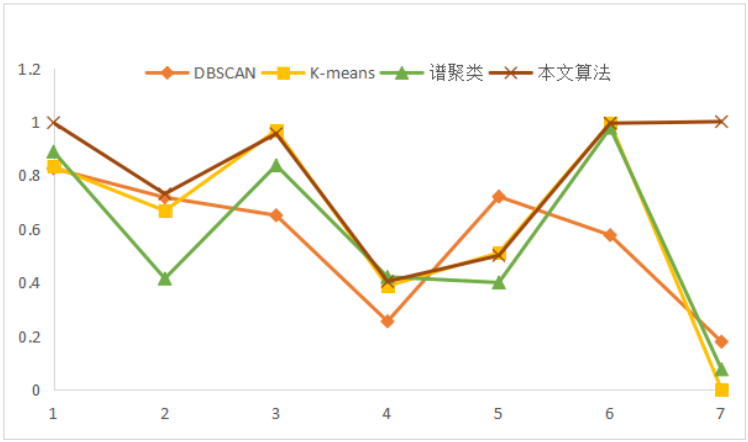
(a) 同质性



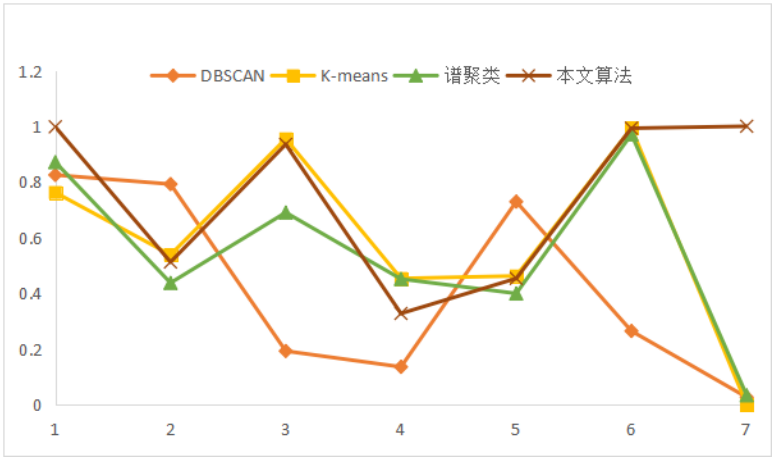
(b) 完整性



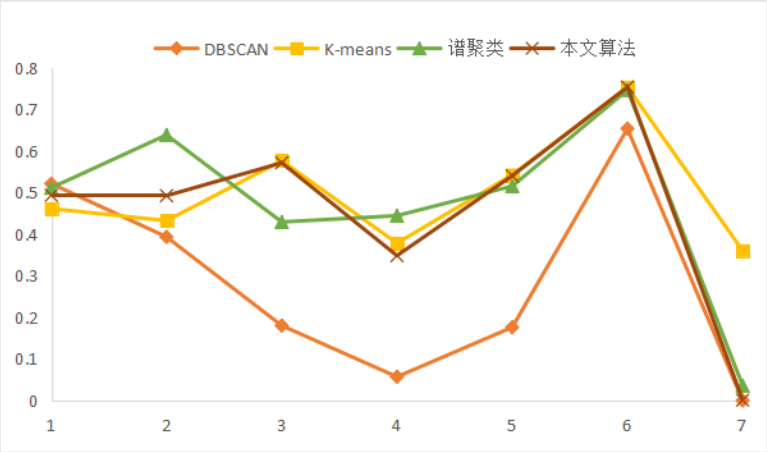
（c）V-measure



(d) 调整兰德系数



（e）调整互信息



（f）轮廓系数

3. 评价结果

从聚类效果的对比图和各个量化评估标准的对比图可以看出，与传统的几类聚类算法相比，本文研究实现的新型密度山峰聚类算法具有的改进的优点：

（1） 在 K-means 算法中，目标函数通常会是所有元素与一套假设出的聚类中心的距离之和，然后对其优化 3-6 次以求出那些最合适的聚类中心。然而，由于一个数据点总是分配到离它最近的中心，因此这种方法不能检测非球形的群，且聚类效果的好坏很大程度生取决于初值的选择和代表数据集的测试概率的可靠性。而 Clustering by fast search and find of density peaks 算法可以识别出非球形的群。

（2） Clustering by fast search and find of density peaks 算法只依赖于数据点之间的距离，计算简单，群的区分不依赖于它们的形状和所处的空间维度。DBSCAN 算法将簇定义为汇集了相同的局部最大密度分布函数的点集。该方法可以分出非球形的集群但是只适用于坐标化的数据而且计算花费时间很长。

（3） 群的数目直观可见、噪声点自动标出并排除在分析之外。在其它算法中需要人为地选取一个密度阈值作为“门槛”，对于密度值低于该“门槛”的数据点则视为噪声点，选择一个合适的“门槛值”是不容易的。

存在的问题和不足：

（1） 原文作者表明该方法对数据的量度变化有鲁棒性而不怎么影响到 dc 的取值，即公式（1）中密度估计值不变。显然，公式（2）中距离值会被这样的量值变化影响，但是很明显决策图的结构（特别是那些 δ 很大的数据点）是一个排名的结果密度值，而不是实际很远的点间的距离值。但在实际操作过程中，由于基于决策图的判断，dc 会对聚类中心的选择产生很大影响从而影响聚类结构。

（2） 该方法对于那些密度比较均匀，即没有密度山峰的数据集测试效果不佳。对于用来模糊聚类的测试集，由于本算法是基于密度这一原理，导致对分布太过均匀的数据点不能很好地识别出聚类中心。

四．具体应用

1．基于密度聚类的警力辖区及权重划分

1.1 待解决的问题

为了解决像中国这样的人口大国接到报警后出警速度慢，对地形不熟悉造成的到场迟缓等问题，我们根据密度，以居民区的经纬度为参数，输入聚类算法划分类簇：以一个类簇的所在区域为一个辖区，以类簇中所包含的样本点个数占总数百分比作为权重，来为每个辖区按权重分配主管该区域的警力资源和确定机构所在位置，以确保每次接到报警时，能够以最快的速度到达案发现场，减少犯罪行为为社会带来的损失，保障人民的人身财产安全。

中国是一个地广人多的国家，人口密度大，2017 年在职的公安干警，交警，协警，武警及刑警人数在 220 万人，平均 63 位公民里有一位是警察，而其中真正能参与日常治安管理的警察其实不到警力资源总人数的二分之一。当今社会在人口稠密区域发生的犯罪行为，如入室抢劫，盗窃，或是出租车司机的道路犯罪，都与人口聚集地区有一定的关系。如果我们能够通过统计宾馆，小区等人群聚居地的地理位置，并按照其密度划分辖区以此代替以行政区为单位分配警力资源，这样可以避免类似于“两区交界责任不清”，“警力不足”等现存问题，使公民的人身财产安全得到更好的保障。

综上，我们选择了基于密度的聚类算法来实现这个目标。

1.2 分析与实现

在收集的四百条小区经纬度数据中，包含了在重庆市二十个区抽取的各二十个小区，经过聚类，我们发现这个算法并不会单纯地根据行政区来划分类簇，而是模糊了行政区划分的概念，在不同的区之间，它能根据密度对区域边缘的数据点，即两区交界处的数据点划分进与其经纬度更相似的类簇中。

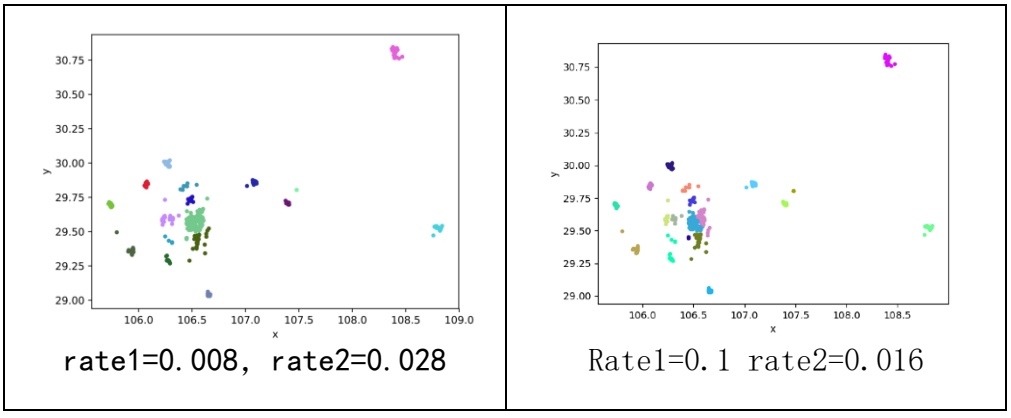


图 7 密度聚类后不同类簇数据点分布的缩略图

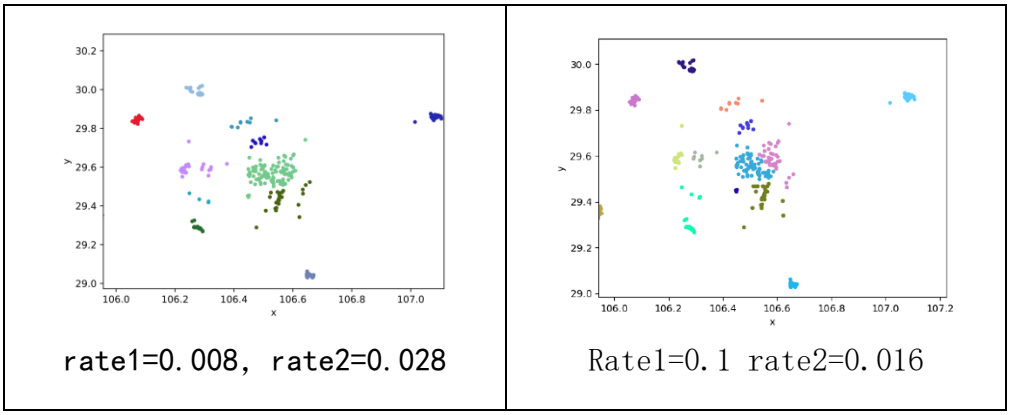


图 8 典型模糊行政区区域示意图

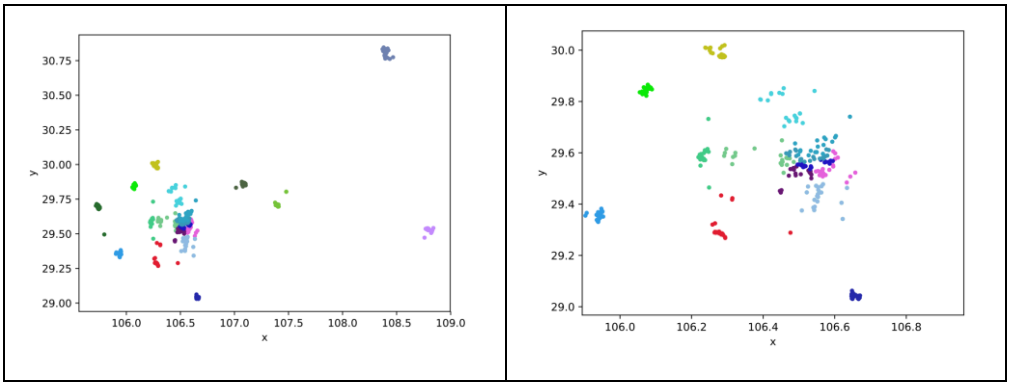


图 9 典型模糊行政区区域示意图

1.3 结果与评估

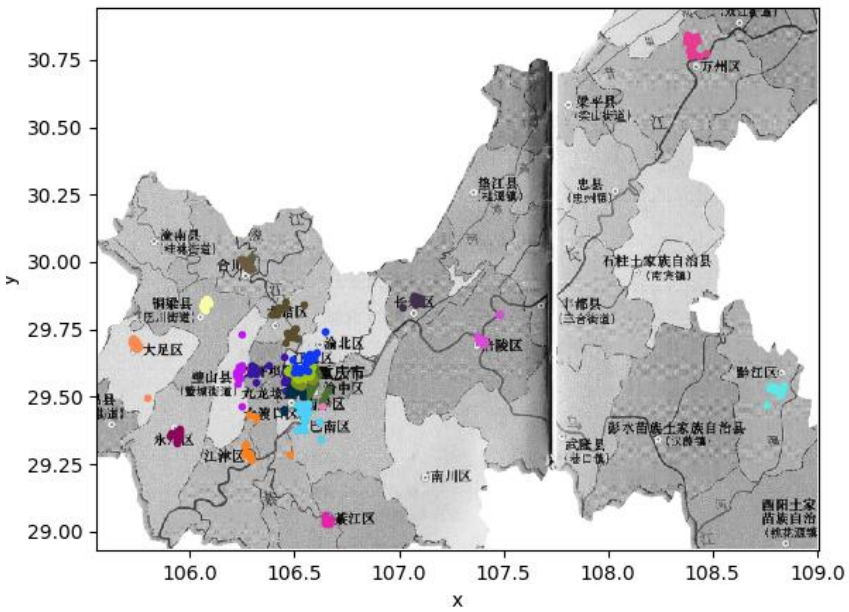


图 11 模糊行政区区域示意图（1）

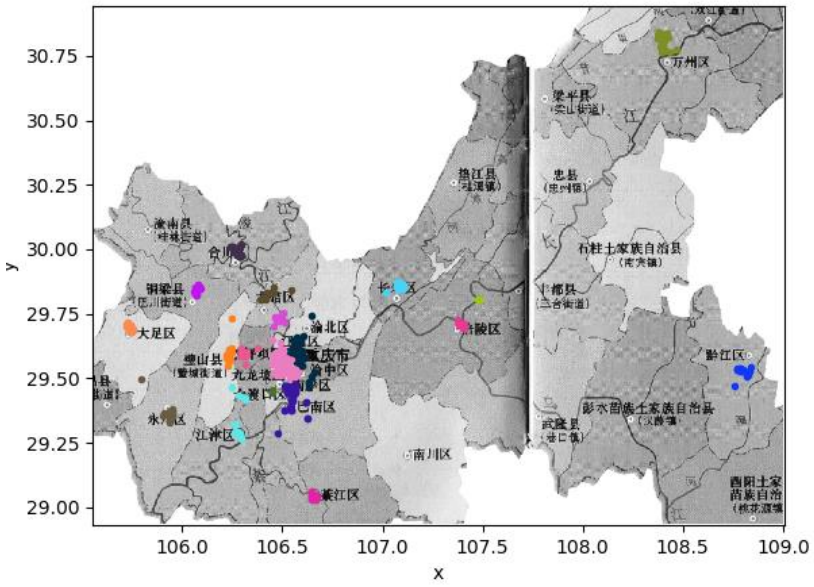


图 11 模糊行政区区域示意图（2）

密度聚类算法对处于狭长地形的行政区的数据点作用非常明显。在这种情况下，若按行政区划分辖区，则警力中心极有可能对辖区某一端，甚至是两端都有较长的路程。但在使用密度聚类时，算法将会把狭长地区分为几块而划分入该区域旁边的拥有更高密度的类簇。

密度越高，范围越大的类簇则代表了该地域人口密集程度，直接表明了对警力资源分配的权重要求。

此时可能面对的一个问题是，过于密集的区域范围可能会过大，如 rate1=0.008，rate2=0.028 时的情况。但适当调高 rate1，调低 rate2 即可解决此问题，如对比图中使用的 rate1=0.1，rate2=0.0016。使用时可以根据需求来调整区域划分，适当将范围过大的簇划分开，减轻辖区压力，也能够缩短簇中心到簇中最远数据点的距离和。

2. 基于 GTD[23]数据库的中国近 20 年恐怖主义事件分析与预测

2.1 待解决的问题

全球恐怖主义数据库（GTD）是一个开源数据库，包含了世界各地恐怖主义事件的信息。GTD 包括了国内和国际恐怖主义事件的系统数据，它现在包括了 180000 多起案件。通过分析中国近 20 年来的恐怖主义事件，用聚类的方法提取特征值，将这些事件分为若干组,；从数据中找到一些地域联系并预测明年最危险的地区。

2.2 分析与实现

2.2.1 事件分类

输入的特征值是量化过后的 GTD 数据库中的[死亡人数，受伤人数]，经过多轮迭代，我选择分为 5 类，分类结果就是不同的人员伤害程度，结果如图 12 所

示：

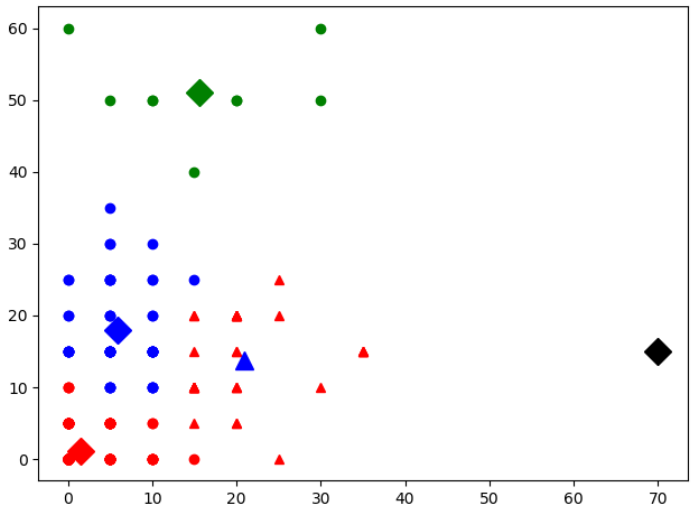


图 12 伤亡程度聚类示意图

估计的聚类个数为：5
各类的统计个数为：
{0：162，1：34，2：41，3：9，4：6}

输入的特征值（target type ,attack type，weapon type 为标准来分类，我暂时将分类结果定为恐怖事件性质，结果如图 13 所示：

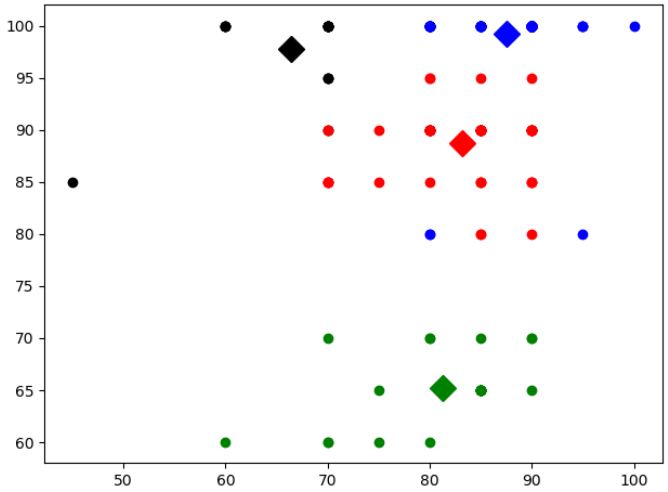


图 13 恐怖袭击性质聚类示意图

估计的聚类个数为：4
各类的统计个数为：
{0：57，1：153，2：24，3：18}

2.2.2 地理特征分析与预测

将事件法发生的地理定位数据，采用本文所研究的新型密度聚类算法实现对其的聚类。并将分类结果数据可视化显示到相应的地图上，从而有利于对中国近 20 年来所发生的恐怖主义事件之间的时空关联，再采用神经网络对提取出的特征与联系进行预测，从而对未来可能发生的恐怖主义事件进行提前预防，

对国家安全和发展具有重要意义：

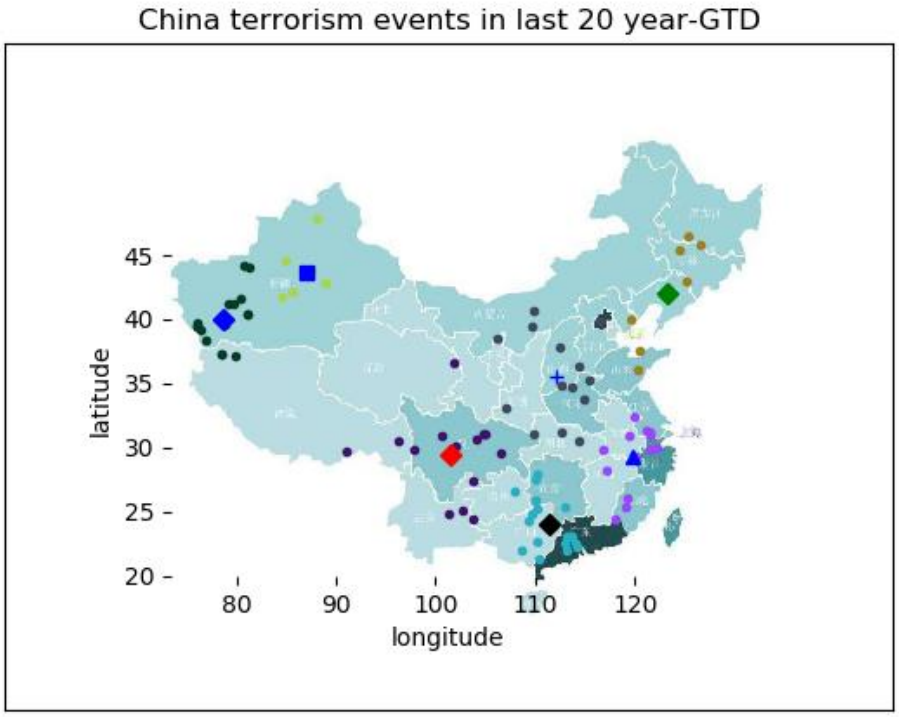


图 14 近 20 年中国恐怖事件分析图

2.3 结果与评估

通过 PCA 降维对数据处理，聚类和神经网络分析后，所得的预测结果如下图所示，可见本文所研究和实现的方法对实际问题的解决具有重要意义：

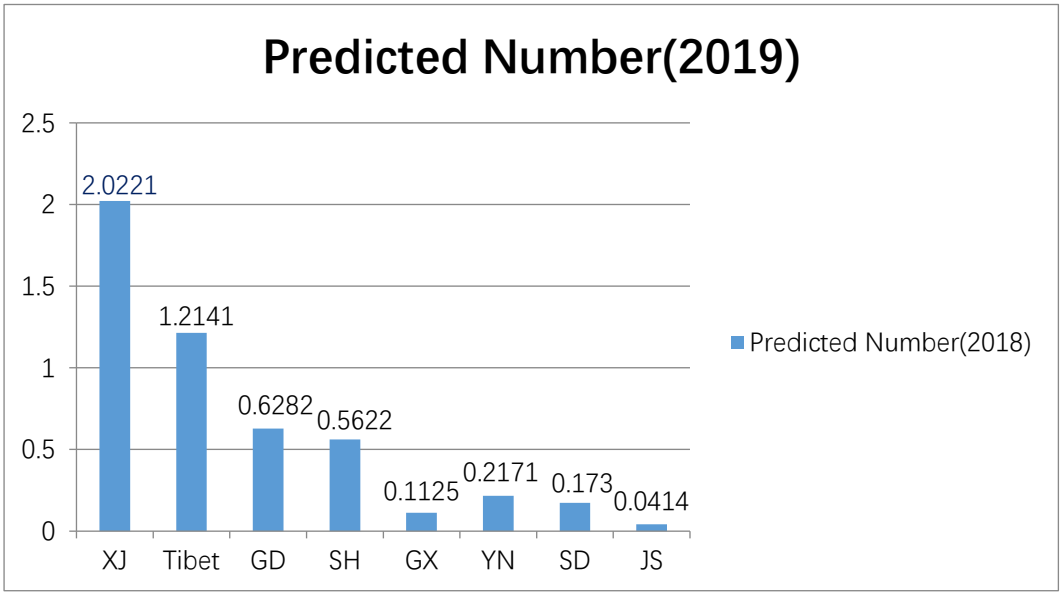


图 15 预测结果示意图

参考文献

[1] Clustering by fast search and find of density peaks MACHINE LEARNING science 2014:1492-1496

[2] 面向位置大数据的快速密度聚类算法 软件学报 2017-07: 961-963

[7] Zheng Y. Trajectory Data Mining: An Overview. Acm Transactions on Intelligent Systems & Technology, 2015, 6(3):1-41.

[8] Zheng Y, Zhang L, Xie X, et al. Mining interesting locations and travel sequences from GPS trajectories[C]// International
Conference on World Wide Web, WWW 2009, Madrid, Spain, April. 2009:791-800

[9] Zheng Y, Zhang L, Ma Z, et al. Recommending friends and locations based on individual location history. Acm Transactions on the Web, 2010, 5(1):99-111.

[10] Guo C, Liu JN, Fang Y, Luo M, Cui JS. Value extraction and collaborative mining methods for location big data.Ruan Jian Xue Bao/Journal of Software, 2014, 25(4):713−730 (in Chinese)

[11] https://blog.csdn.net/lvxiong1990/article/details/40540065

[12] YU Yan-Wei1+, JIA Zhao-Fei1, CAO Lei2, ZHAO Jin-Dong1, LIU Zhao-Wei1, LIU Jing-Lei An Efficient Density-based Clustering Algorithm for Location Big Data

[13] A. Gionis, H. Mannila, and P. Tsaparas, Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2007. 1(1): p. 1-30.

[14] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 1971. 100(1): p. 68-86.

[15] C.J. Veenman, M.J.T. Reinders, and E. Backer, A maximum variance cluster algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence* 2002. 24(9): p. 1273-1280.

[16] L. Fu and E. Medico, FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data.*BMC bioinformatics*, 2007. 8(1): p. 3.

[17] H. Chang and D.Y. Yeung, Robust path-based spectral clustering. *Pattern Recognition*, 2008. 41(1): p. 191-203.

[18] C.J. Veenman, M.J.T. Reinders, and E. Backer, A maximum variance cluster algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2002. 24(9): p. 1273-1280.

[19] H. Chang and D.Y. Yeung, Robust path-based spectral clustering. *Pattern Recognition*, 2008. 41(1): p. 191-203.

[20] G. Karypis, E.H. Han, V. Kumar, CHAMELEON: A hierarchical 765 clustering algorithm using dynamic modeling, *IEEE Trans. on Computers*, 32 (8), 68-75, 1999.

[21] Home Page of Alessandro LaioResearch - Docking

[22] User locations (Joensuu) Mopsi dadasets

[23] 位 珍 珍 后 911 时代恐怖主义的 GTD 数据分析 计算机应用研究，2017，7(36)：