基于密度峰值和网格的自动选定聚类中心算法

夏庆亚
(浙江工业大学信息工程学院 杭州 310023)

摘 要 针对快速搜索和发现密度峰值的聚类算法(DPC)中数据点之间计算复杂,最终聚类的中心个数需要通过决策图手动选取等问题,提出基于密度峰值和网格的自动选定聚类中心的改进算法GADPC。首先结合Clique网格聚类算法的思想,不再针对点对象进行操作,而是将点映射到网格,并将网格作为聚类对象,从而减少了DPC算法中对数据点之间的距离计算和聚类次数;其次通过改进后的聚类中心个数判定准则更精确地自动选定聚类中心个数;最后对网格边缘点和噪声点,采用网格内点对象和相邻网格间的相似度进行了处理。实验通过采用UEF(University of Eastern Finland)提供的数据挖掘使用的人工合成数据集和UCI自然数据集进行对比,其聚类评价指标(Rand Index)表明,改进的算法在计算大数据集时聚类质量不低于DPC和K-means算法,而且提高了DPC算法的处理效率。

关键词 数据挖掘,聚类分析,密度峰值,网格,相似度
中图法分类号 TP181 **文献标识码** A

Automatically Selecting Clustering Centers Algorithm Based on Density Peak and Grid

XIA Qing-ya
(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract Aiming at the shortcomings of clustering by fast search and find of density peaks algorithm(DPC), which calculates massive distance between point objects, has high computational-complexity about clustering process, and needs to select the final cluster centers manually, an improved algorithm that choose clustering centers automatically based on density peak and grid(GADPC) was proposed. Firstly, with the idea of Clique algorithm, all data points are mapped to grid clustering with grid objects rather than point objects, in order to reduce the distance computation and clustering complexity of DPC algorithm. Secondly, the decision accuracy of the number of cluster centers is improved so that it can automatically select cluster centers more precisely. Finally, the relative similarity between grid internal points and adjacent grid points is dealt, so that the edge points and noise points can be solved well. Comparing with machine learning synthetic data sets of UEF and UCI natural data sets, the rand index of those data sets shows that the clustering quality of the improved algorithm is not lower than DPC and K-means algorithm when calculating large data sets, and it improves the dealing efficiency of DPC algorithm.

Keywords Data mining, Clustering analysis, Density peak, Grid, Similarity

1 引言

聚类分析在数据挖掘技术中已经占有举足轻重的地位[1],目前很多领域都应用了该技术,如机器学习[2]、模式识别[3]、数据分析、图像处理、市场研究[4]等。

聚类(Clustering)就是将数据对象划分为不同的类簇(cluster),使得同一类簇中的对象具有较高的相似性,而不同类簇中的对象相差较大。聚类算法大体上可分为:基于划分的方法、基于层次的方法、基于密度的方法、基于网格的方法、基于模型的方法。如基于划分的聚类算法[5]有K-means[6],K-medoids[7],CLARANS[8]等;基于密度的聚类算法有DBSCAN[9],GDBSCAN[10],OPTICS[11]等;基于层次的聚类算法有CURE[12],CHAMELEON[13],BIRCH[14]等;基于网格的聚类算法有STING[15],WaveCluster[16],Clique[17]等。

经典的聚类算法在不同的情景下各自表现出优势,但在其他数据应用上存在缺陷,例如K-means算法虽然实施简单、复杂度低,但是聚类的数目需要事先指定,对初始值K比较敏感,且只适用于发现球形数据;DBSCAN算法的聚类速度快,处理噪音和发现任意形状的聚类能力强,但是受密度半径Eps和密度阈值$MinPts$影响较大;STING,CLIQUE算法处理时间复杂度与数据对象的个数无关,但是受每维空间划分的单元对象影响,一定程度上降低了聚类的质量和准确性。

Alex Rodriguez和Alessandro Laio在科学杂志上发表了基于密度峰值的新聚类算法DPC(clustering by fastsearch and find of density peaks)[18]。该算法的核心思想是查找高密度集群使得低密度集群归属其中,能快速发现任意形状数据集的密度峰值点(类簇中心),并能很好地把数据点归属到正确的类中,且能删除噪音点。DGCCD算法[19]以网格化数据集来减少聚类过程中的计算复杂度,它提出了一种基于密度和网格的簇心可确定聚类算法,但是最终的聚类中心及其个数也需要通过决策图手动确定,而且经过数据测试,该算法由于边缘处理不够完善,导致许多数据类型的准确率较低,本

夏庆亚(1990—),男,硕士,主要研究方向为数据挖掘,E-mail:xqy0610@foxmail.com。

文是在该算法上做进一步改进。

针对以上问题，本文提出一种基于密度峰值和 Clique 网格划分思想且可自动选定簇类中心的聚类算法 GADPC。该算法首先对数据集网格化，将每一个网格看作处理的元数据；再通过网格的两个属性(网格密度值和距离值)，利用 DPC 的聚类思想，选取被很多网格围绕并且与局部密度比其大、距离也很远的网格作为类簇中心。选定中心后，让周围的网格采取“跟随”策略，归类到密度比自己大的最近邻居所在的簇中。

2 DPC 和 Clique 算法的优缺点

2.1 DPC 算法

DPC 算法假设类簇中心周围是一些具有较低局部密度的点，并且与具有更高密度的点的距离都较大。对于每一个数据点 x 都要计算点的局部密度 ρ_i 以及该点到更高局部密度的点的距离 δ_i，而这两个值的计算都要依赖于数据点之间的距离。数据点 x 的局部密度 ρ_i 的定义如式(1)、式(2)所示：

$$\rho_i=\sum_j \chi(d_{ij}-d_c) \tag{1}$$

$$\chi(x)=\begin{cases}1, & x<0\\ 0, & x\geqslant 0\end{cases} \tag{2}$$

其中，d_c 是截止距离，它对于数据点的密度大小相对比较敏感。数据点 i 到更高局部密度的点的距离 δ_i 如式(3)所示，它是点到任何比其密度大的节点的距离的最小值。

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

然而，对于密度最大的节点，$\delta_i=\max_j(d_{ij})$。因此，将 δ_i 异常大的节点作为最终的类簇中心。

2.2 Clique 算法

Clique(CIustering InOUEst)算法[20]在使用网格的基础上加入了密度概念。首先对每个属性进行 n 等分，整个数据空间就被等距离划分，然后统计单元格中的数据个数作为密度。当密度超过设定的密度阈值则被看作稠密单元，将这些稠密单元进行聚类。Clique 拥有较高的效率，对数据输入顺序不敏感，对于大型数据库中的高维数据的聚类非常有效。但它需要用户输入网格划分的间隔距离和密度阈值参数，且不能自动去除孤立点，聚类结果的精确性因此降低。

3 GADPC 聚类算法

3.1 网格划分

定义 1(网格边长) 在基于网格划分的聚类分析中，将每个维度做等距离划分。给定一个 d 维的数据集 $D=\{x_1,x_2,\cdots,x_n\}$，将 D 分成 d 维网格单元。设第 i 维上的属性值分布在区间 $[l_i,h_i)$ 中，其中 $i=1,2,\cdots,d$，则 $D=[l_1,h_1)\times[l_2,h_2)\times\cdots\times[l_d,h_d)$。将数据空间的每一维划分成长度相等的 M 段，这样整个数据形成 Md 个网格单元。网格单元 C 表示为 $(c_{i1},c_{i2},\cdots,c_{id})$，$c_{ij}=[l_{ij},h_{ij})$ 是一个左闭右开的区间。将数据空间分割成 Md 个等长的超矩形网格单元，为使下一步聚类更加精确，经过文献对比和实验结果，当每一维上的数据的数目取 $\lceil n/2\sqrt{M}\rceil$($\lceil\ \rceil$ 向上取整)时聚类效果最佳。

定义 2(网格密度值) 以落在网格对象 gi 中数据对象的数量作为网格对象密度值，记为 ρ_{gi}。

定义 3(网格距离值) 以网格对象 gi 到更高密度网格对象 gj 的最近距离，作为网格对象 gi 的距离值，记为 δ_{gi}。设 $\{gi\}_{i=1}^{M}$ 表示 $\{\rho_i\}_{i=1}^{M}$ 的一个降序排列的下标序号，将网格密度排序 $\rho_{g1}\geqslant\rho_{g2}\geqslant\cdots\geqslant\rho_{gM}$。

$$\delta_{gi}=\min(d_{gi,gj}),i\geqslant 2,i>j \tag{4}$$

$$\delta_{gi}=\max(\delta_{gj}),i=1,j\geqslant 2 \tag{5}$$

其中，$d_{gi,gj}$ 为网格对象 gi 的中心位置到网格对象 gj 的中心位置之间的欧氏距离。gj 表示除 $i_{g\max}$ 以外的所有网格对象。

3.2 自动确定聚类中心

在 DPC 算法中，簇类中心是以决策图手动选择预期的，选取高密度和高距离值。然而在一个类簇中有多个密度峰值，当这些值的分布差距不大时便很难选择确切的聚类中心的数目，这种现象很容易观察到，如图 1 所示。

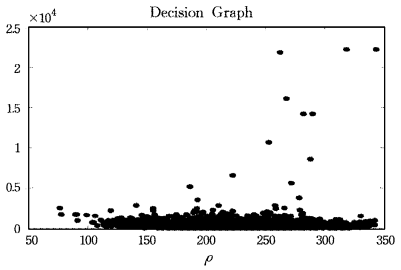

图 1 DPC 算法的密度与距离分布图($n=7500,dc=2$)

为解决上述问题，文献[21]提出了解决方案，但经过测试发现该文提出的方法对数据集的测试的准确度较低，因为 Fuzzy-CF SFDP 算法与 DPC 算法都对 dc 的取值敏感，设置不同，得到的聚类个数也不同。本文对 GADPC 方法进行改进。

GADPC 是一个自适应的方法，无需人工干预选择确切聚类中心的数量。其判定函数为：

$$\rho_{Cgi}-\mu(\rho_{gi})\geqslant 0 \tag{6}$$

$$(\delta_{Cgi}-E(\delta_{gi}))/2\geqslant\sigma(\delta_{gi}) \tag{7}$$

其中，ρ_{Cgi} 是聚类中心网格的密度，$\mu(\rho_{gi})$ 是所有网格密度的均值，δ_{Cgi} 是同一个类簇中距聚类中心最小的距离，$E(\delta_{gi})$ 是所有 δ_{gi} 的期望。当网格对象符合以上判定函数时，将该网格作为最终的聚类中心。

3.3 聚类过程

根据已经计算出的每个网格的密度 ρ_{gi}，每一维网格之间的距离 $d_{gi,gj}$，网格对象 gi 与密度比其大、距离其最小的对象 gj 之间的距离 δ_{gi}，密度最大的网格对象 $gi_{\max}$ 的距离值 δ_{gmax}，将网格对象赋给密度比它大且距离最近的对象，也即该网格和比该网格密度大且距离最近的网格中的点对象进行聚合。在自动选定聚类中心点后，将网格归属到离它最近的中心的类。

3.4 边缘与噪声的处理

网格内点对象的相似度如果是 x_i,x_j 且在同一个网格内，则这两个数据对象之间的相似度为式(9)：

$$inSim(i)=\max\ inDiff-inDifff(i,j) \tag{8}$$

$$\max\ inDifff=\max(inDiff(i,j)) \tag{9}$$

$$inDiff(i,j)=\sum_{l=1}^{d}|x_i-x_j| \tag{10}$$

相邻网格间点的对象相似度如果是指，x_i,x_j 分别位于相邻的网格，即 $|grid(i)-grid(j)|=\pm 1$，且这两个网格属于不同类，则两个数据对象之间的相似度为式(12)，其中，$grid(i)$ 代表第 i 个点所在的网格号：

$$adjSim(i)=\max\ adjDiff-adjDiff(i,j) \tag{11}$$

$$outDiff(i,j)=\sum_{l=1}^{d}|x_i-x_j| \tag{12}$$

$$\max\ adjDiff=\max(adjDiff(i,j)) \tag{13}$$

如果 $\rho_i\leqslant\bar{\rho},\rho_i\leqslant\rho_j$，且 $inSim(i)<adjSim(j)$，则将该点 i 归属到 j。其中 $adjSim$ 是相邻网格间的点对象相似度，ρ_i 是数据点 i 所在网格的网格密度，ρ_j 是数据点 j 所在网格的网格密度，$\bar{\rho}$是一维中所有网格密度的期望值，且 $|grid(i)-grid(j)|=\pm1$。

由于只是计算单个网格和相邻网格之间的相似性，而且都只是采取两点之差的方式，因此计算复杂度相对较低。图 2 是经过边缘与噪声处理后的图，黑色点代表噪声点，从图中可以看出该算法可以很好地处理边缘和噪声。

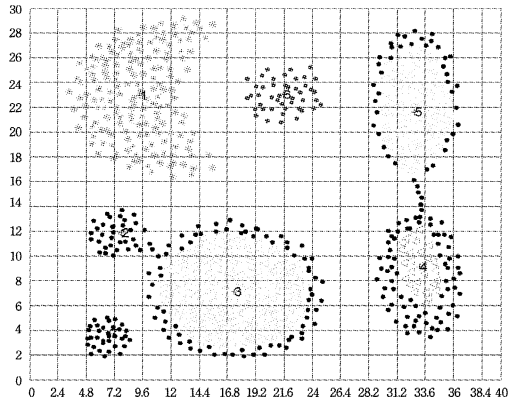

图 2　GADPC 算法边缘与噪声处理结果

3.5　聚类算法描述及步骤

GADPC 聚类算法的基本思想是首先将数据空间划分为网格单元，并将数据对象集 D 映射到网格单元中；计算网格单元的密度，并计算每一维中网格对象之间的距离；将网格聚类到比其密度高且距离其最近的网格中，经自动筛选，得到密度和相对距离较大的几个聚类中心；最后根据网格边缘处理规则完成聚类。

GADPC 算法步骤的详细描述如下。

输入：数据集 Data

输出：聚类中心 clusters

步骤 1：首先将数据标准化处理，然后将数据的每一维均匀划分为均匀且相等的网格；

步骤 2：记非空网格数为 M，且使得 M 等于 $n/2\sqrt{M}$取整，整个数据空间 D 划分成 Md 个网格单元；

步骤 3：将数据对象集 D 映射到网格单元中；

步骤 4：依次扫描每个网格单元，其中网格单元中的数据对象个数为网格密度，排序后记为 ρ_{gi}，$1\leqslant gi\leqslant M$；

步骤 5：计算每一维的网格之间的距离 $d_{gi,gj}$；

步骤 6：根据式(5)和式(6)计算网格的距离值；

步骤 7：找到聚类中心，计算所有密度期望值、距离的期望以及标准方差，根据式(7)和式(8)选出 Ci；

步骤 8：根据处理噪声和边缘，计算每个网格内所有数据对象的相似度，以及相邻网格不属于同一类簇的对象相似度，并进行比较。若 $inSim(i)<adjSim(j)$，则将点 i 聚类到它相邻的网格 j。

4　算法复杂度分析

假设聚类数据集有 n 个 d 维数据集，DPC 算法中时间复杂度主要用于计算样本间的距离，每个样本的密度 ρ 以及每个样本的 δ 距离。各部分的时间复杂度都为 $O(n^2)$，因此总的时间度杂度为 $O(n^2)$。

本文提出的 GADPC 算法初始化网格单元并将数据点映射到网格单元中的复杂度为 $O(n)$。统计网格信息并排序的复杂度为 $O(M\log M)$，其中 M 为排除空白网格之后的非空网格数目，通常情况下 $M\log M<n$。网格化并获取网格对象的密度，该过程的计算代价为 $O(nd)$，算法进行一次划分完成聚类，其计算代价为 $O(M^2)$，最后边缘处理过程的计算代价为 $O(bn)$，其中 b 表示边缘点的个数。因此 GADPC 算法的算法时间复杂度为 $O(nd+M^2+bn)$。

5　实验结果及分析

实验使用的环境为 Intel(R) Core(TM) i3 CPU，6GB 内存，操作系统为 Windows 7 Ultimate，仿真软件为 MATLAB 8.3.0.532 (R2014a) (64-bit)。

5.1　结果分析

实验采用带有标签的人工合成数据集 Data1，Data2，Data3，Data4，Data5，Data6，UCI 自然数据集 iris 以及 CHAMELEON 数据集。其中人工数据集数据个数从 240 到 8000 分布，以便比较不同数据量对算法的影响。

在聚类算法中，可以使用兰德指数(Rand Index)来描述不同类簇间的相似度和同一类簇中的相似度。一个数据集中有 N 个数据，其中包括同一类的数据被分到同一个簇(TP)，不同类的数据被分到不同簇(TN)，不同类的数据被分到同一个簇(FP)，同一类的数据被分到不同簇(FN)，RI 度量指数 $RI=(TP+TN)/(TP+FP+FN+TN)$，其算法精度指标为 $Precision=TP/(TP+FP)$，调整后的兰德指数 $AdjRI=\frac{RI-E[RI]}{\max(RI)-E[RI]}$。本文采用 Rand Index、精度和调整后的 AdjRI 评价聚类质量，具体比较结果如表 1 所列。

GADPC 与 K-means 的精度对比如表 2 所列。

表 1　GADPC 与 DPC 的精度对比

数据	DPC				GADPC	
	数据大小	RandIndex	Precision	AdjRI	RandIndex	Precision	AdjRI
Data1	373	0.7565	0.9965	0.9903	0.8562	0.9265	0.7055
Data2	600	0.9680	0.8150	0.7554	0.9974	0.9795	0.9786
Data3	3100	0.8476	0.8394	0.4655	0.9951	0.9220	0.9207
Data4	788	0.8174	0.8504	0.5817	0.9971	0.9985	0.9913
Data5	240	0.8341	0.9650	0.0133	0.9671	0.9652	0.9338
Data6	399	0.9092	0.9981	0.7826	0.9016	0.9188	0.7651
Iris	150	0.8763	0.8951	0.8681	0.7406	0.7748	0.7217

由表 1 可以看出，GADPC 与 DPC 的聚类质量基本持平，数据集 Data2，Data3，Data4 由于采用结合网格的技术，表现得更为出色；对自然数据集 Iris 进行聚类也取得了相对不错的效果。

表 2 GADPC 与 K-means 的精度对比

数据	K-means				GADPC	
	数据大小	RandIndex	Precision	AdjRI	RandIndex	Precision	AdjRI
Data4	788	0.9971	0.9985	0.9913	0.9971	0.9985	0.9913
Data6	399	0.8517	0.7948	0.5533	0.9016	0.9188	0.7651
Iris	150	0.8169	0.8864	0.5166	0.7406	0.7748	0.7217

由表 2 可以看出，在算法精度上 GADPC 算法比 K-means 算法略低，但相差不大；而 K-means 算法有一定限制性，因为它需要提前设定 K 值（聚类个数），本文改进的算法能够自动确定聚类个数，减少了人工参与的过程。

由图 3 中时间复杂度的比较可以看出，相比 DPC 算法，本文提出的算法在计算大数据集时比 DPC 更具优势。

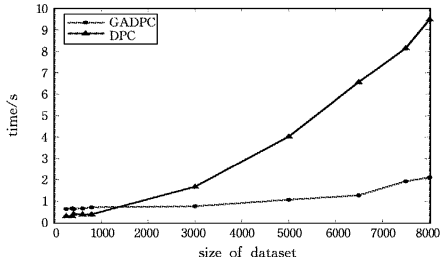

图 3 GADPC 与 DPC 算法的时间对比

由图 4 可以看出，本文改进后的算法对 CHAMELEON 数据集处理得很好。

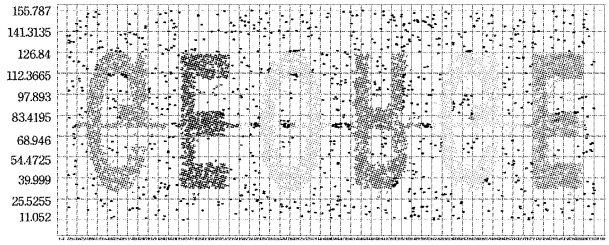

图 4 GADPC CHAMELEON 数据集显示效果

结束语 本文提出了一种基于密度峰值和网格思想，可自动选定聚类中心个数的聚类算法。基于密度的聚类算法对于任意形状分布的数据集都有较好的聚类结果，且应用了网格划分来减少距离和聚类次数的计算复杂度，针对大型数据，其运行速度明显提高。本文中网格划分在一定程度上影响着聚类结果，对非平衡数据上的密度值和距离值的选取也有特殊要求，因此这两个问题将是下一步研究的重点。

参 考 文 献

[1] ARABIE P, HUBERT LJ. An Overview of Combinatorial Data Analysis[M]// Clustering and Classification. 2003:5-63.

[2] MICHALSKI R S, STEPP R E. Learning from Observation: Conceptual Clustering[M]. Machine Learning: An Articial Intelligence Approach, 1983:331-363.

[3] FUKUNAGE K. Introduction to Statistic Pattern Recognition [M]. Academic Press, 1990.

[4] QIAN W, ZHOU A. Analyzing popular clustering algorithms from different viewpoints[J]. Journal of Software, 2002, 13(8): 1382-1394.

[5] YANG W, WANG T, LI J D. Clustering parameter selection algorithm based on density for divisional clustering process[J]. Control & Decision, 2016, 31(1): 21-29.

[6] LLOYD S. Least squares quantization in PCM[J]. IEEE Transactions on Information Theory, 1982, 28(2): 129-137.

[7] 夏宁霞，苏一丹，覃希. 一种高效的 K-medoids 聚类算法[J]. 计算机应用研究，2010，27(12)：4517-4519.

[8] NG R T, HAN J. CLARANS: A Method for Clustering Objects for Spatial Data Mining[J]. IEEE Transactions on Knowledge & Data Engineering, 2002, 14(5): 1003-1016.

[9] ESTER B M, KRIEGEL H P, SANDER J. et al. A Density Based algorithm for discovering clusters in large spatial databases[C]// Proceedings of International Conference on knowledge Discovery and Data Mining. AAAI, 1996: 226-231.

[10] SANDER J, ESTER M, KRIEGEL H P, et al. Density Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications[J]. Data Mining &Knowledge Discovery, 1998, 2(2): 169-194.

[11] ANKERST M, BREUNIG M M, KRIEGEL H P, et al. OPTICS: Ordering Points to Identify the Clustering Structure[J]. Stanford Research Inst Memo Stanford University, 1999, 28(2): 49-60.

[12] GUHA S, RASTOGI R, SHIM K. CURE: An Efficient Clustering Algorithm for Large Databases[C]// Proc. of the ACM SIGMOD International Conference on Management of Data. 1998: 73-84.

[13] KARYPIS G, HAN E H, KUMAR V. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling[J]. IEEE Computer, 1999, 32(8): 68-75.

[14] 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.

[15] WANG W, YANG J, MUNTZ R R. STING: A Statistical Information Grid Approach to Spatial Data Mining[C]// Proceedings of the 23rd International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc. 1997: 186-195.

[16] SHEIKHOLESLAMI G, CHATTERJEE S, ZHANG A. WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases[C]// International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc. 1998: 428-439.

[17] AGRAWAL R, GEHRKE J E, GUNOPULOS D, et al. Automatic subspace clustering of high dimensional data for data-mining applications[M]// ACM SIGMOD Record. ACM, 1998: 94-105.

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

[19] 何熊熊，管俊轶，叶宣佐，等. 一种基于密度和网格的簇心可确定聚类算法[J]. 控制与决策，2016，7(5)：913-919.

[20] GOIL S, NAGESH H, CHOUDHARY A. MAFIA: Ecient and Scalable Subspace Clustering for Very Large Data Sets[R]. Technical Report, 1999.

[21] MEHMOOD R, BIE R, DAWOOD H, et al. Fuzzy clustering by fast search and find of density peaks[C]// International Conference on Identification, Information, and Knowledge in the Internet of Things. IEEE, 2016: 258-261.