一种多约束的密度聚类算法的研究

江 敏[1] 皮德常[1] 孙 兰[2]

(南京航空航天大学计算机科学与技术学院 南京 210016)[1] (南京航空航天大学理学院 南京 210016)[2]

摘 要 针对传统的密度聚类算法不能处理带有多约束条件的问题,在现有的密度聚类算法的基础上,提出了一个带有多约束条件限制的密度聚类算法。该算法将多约束条件引入到密度聚类分析中,并分析了多约束条件对聚类结果的影响。实验表明该算法在多约束条件下,可有效完成对数据点的聚类并且效果较好,为现实情况中处理多约束聚类提供了良好的理论支持。

关键词 多约束条件,密度,聚类

中图法分类号 TP301.6 **文献标识码** A

Research on Density Clustering Algorithm with a Multiple Constraints

JIANG Min[1] PI De-chang[1] SUN Lan[2]

(College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)[1]

(Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)[2]

Abstract Traditional clustering algorithms based on density can not overcome the shortage with a variety of constraints in the existing clustering algorithm based on the density proposed. This paper proposed a clustering algorithm based on density with a variety of constraints. The algorithm introduces a variety of constraints into clustering algorithm to analyze the clustering results affected by a variety of constraints. Experimental results show that the algorithm in a multi-constrained condition can complete a cluster analysis of the data points, and can obtain a better clustering results, provides a good theoretical support for really dealing with multiple constraints clustering.

Keywords Multiple constraints limit, Density, Clustering

1 引言

聚类是对物理的或者抽象的对象集合分组的过程,聚类分析在统计学、数据挖掘、模式识别、信息检索等方面的研究较为活跃,出现了很多有效的、可伸缩的算法,包括:层次聚类算法[1-3]、分割聚类算法[4,5]、基于约束的聚类算法[6,7]、机器学习中的聚类算法[8-10]以及用于高维数据的聚类算法[11,12]等。Han[5]提出了一种带障碍的聚类问题:如果需要在一个地区放置一些ATM,就需要考虑该区域河流、高速公路等约束条件。Han在这篇文章中提出了一个带障碍的聚类算法COD-CLARANS[5],它能有效地处理这种带障碍的聚类问题,但是存在占用内存过多,还有密集区域需放置多台ATM的用户限定的约束条件等问题。另一个考虑障碍空间中比较著名的聚类算法是DBCLuC[13]。DBCLuC是在基于密度的聚类算法DBSCAN的基础上改进的,从而适应了障碍空间中的聚类算法。在DBCLuC中,如果两个对象之间的连线与障碍物相交,即两个对象互相不可见,则规定它们之间的距离为无穷大。与COD-CLARANS相比,DBCLuC有着许多优点,如对噪声不敏感,聚类的效率高,能够发现任意形状的簇类。

研究带有障碍的密度聚类算法已经成为当今数据挖掘研究的热点之一。本文提出了一种多约束的密度聚类算法,它在原有的密度聚类算法DBSCAN基础上,增加了对聚类条件的限制,在数据点间放置了障碍物和数据点间增加关联度,更加符合实际应用中的现实情况。如对某些差异较大的数据点,可以减少彼此间关联度的值;若低于给定的阈值,则不应聚类到同一簇中,从而人为地改变聚类的结果,大大改善了聚类的质量。

2 DBSCAN算法原理

DBSCAN的算法思想是:从数据集D中的任意一个点P开始,查找D中所有关于Eps和Minpts的从P密度可达的点。若P是核心点则其领域内的所有点和P同属于一个类,这些点将作为下一轮的考察对象(即种子点),并通过不断查找从种子点密度可达的点来扩展它们所在的类,直至找到一个完整的类;若P不是核心点,即没有对象从P密度可达,则P被暂时地标注为噪声。然后,算法对D中的下一个对象重复上述过程……当所有种子点都被考察过,一个类的扩展就完成了。此时,若D中还有未处理的点,算法则进行另一个类的扩展;否则,D中不属于任何类的点即为噪声。算法步骤如下:

(1)从数据集中任意选取一个点P,对其进行区域查询;

(2)如果P是核心点,则寻找所有从P密度可达的点,最终形成一个包含P的簇;

(3)否则,P被暂时标注为噪声点;

(4)访问数据集中的下一个点,重复上述步骤,直到数据集中所有的点都被处理。

3 多约束空间中的密度聚类

3.1 多约束模型

在现实应用中，存在着多种多样的约束，这里将其简化为障碍模型和用户约束的情形。一般可将障碍模型分为线形（高速公路和河流等）和面形（山和湖泊等）[14]。

本文将线形障碍定义为一个二元组 $O(u_{start}, u_{end})$，其中 u_{start} 为线段的起始端点，而 u_{end} 为线段的终止端点。空间部分中的数据点之间存在可见性，假设线段 L 是连接数据点 p,q 的线段且 $p,q \in D$，D 为数据库；如果线段 L 与任一障碍物 (u_{start}, u_{end}) 都不相交，则说明 p,q 是互相可见的。然而，对于面形的障碍，可以把面行的障碍的每一条边当成一个线形的障碍来看，将面形障碍转化为多个线形障碍处理。

本文将用户约束定义为一个关联矩阵 R，对 $p,q \in D$，D 为数据库，R_{ij} 表示为 p 和 q 的相关程度。由用户指定相关性的阈值 T，若低于指定的阈值，将视为不相关，即 p 和 q 不能聚类在一个簇中；否则，p 和 q 可以聚类在一个簇中。

3.2 模型中的相关定义

这里给出模型中的相关定义，以便更加清楚地描述后面提出的算法。

定义 1（数据点间的连接距离） 设点 $p,q \in D$ 且互不可见，线段 pq 与 $O(u_1,u_2) \in O(u_{start}, u_{end})$ 相交，则 p,q 的连接距离为：

$$dist_1(p,q) = distance(p,u_1) + distance(u_1,q)$$

$$dist_2(p,q) = distance(p,u_2) + distance(u_2,q)$$

定义 2（数据点间的障碍距离） 设点 $p,q \in D$ 且互不可见，线段 pq 与 $O(u_1,u_2) \in O(u_{start}, u_{end})$ 相交，则 p,q 的障碍距离为：

$$ObstacleDist(p,q) = \min(dist_1, dist_2)$$

在带有障碍的空间中，数据点之间的距离已经不再是欧几里得距离，而是通过绕行障碍物得出的绕行距离。

定义 3（数据点间的距离） 设点 $p,q \in D$，则 p,q 之间的距离 $DataDistance$ 为：

（1）若 p,q 可见，则距离为：

$$DataDistance(p,q) = distance(p,q)$$

（2）若 p,q 不可见，则距离为：

$$DataDistance(p,q) = \min(dist_1, dist_2)$$

定义 4（障碍下边界点） 障碍空间中任意一点，它本身不是核心点，但在某个核心点的领域内，由于位于类的边界，因此被称为障碍下边界点。

从上面的定义可以看出，核心点位于数据对象的密集区域，而障碍下边界点相对于核心点则处于相对稀薄的边缘区域。

定义 5（障碍下直接密度可达） 给定 $Eps, Minpts$，若点 p 和点 q 满足：$p \in NEps(q)$，而且 $|NEps(q)| >= Minpts$，则点 p 和 q 障碍下直接密度可达。

很明显，对于两个核心点来说，障碍下直接密度可达是对称的关系。对于一个核心点和一个障碍下边界点来说，障碍下直接密度可达是不对称的关系。

定义 6（障碍下密度可达） 给定 $Eps, Minpts$，若点 p 和点 q 之间存在一个链 $p_1, p_2, \cdots, p_n$，其中 $p_1 = p, p_n = q$，且有 p_i, p_{i+1} 障碍下直接密度可达，则点 p 到点 q 障碍下密度可达。

障碍下密度可达是障碍下直接密度可达的一个扩展。障碍下密度可达是可传递的，但是障碍下密度可达也不是对称的关系。

定义 7（障碍下密度相连） 给定 $Eps, Minpts$，若存在点 o，使得点 p 和点 q 都从 o 障碍下密度可达，则点 p 和点 q 是障碍下密度相连的。

障碍下密度相连是一个对称关系。如果 p 与 q 密度相连，那么 q 与 p 也是密度相连的关系。

定义 8（障碍下的聚类） 对于数据集 D 的非空子集 C 是一个类，当且仅当 C 满足下面的条件：

（1）（极大性） 对于任意的 p,q，若 $p \in C$，且 q 从 p 障碍下密度可达，则 $q \in C$；

（2）（连通性） 对于任意的 p,q，若 p 和 $q \in C$，则 p 和 q 是障碍下密度相连的。

定义 9（最小关联性阈值） 在给定的用户约束条件下，指定两个数据点具有关联性的最小值，称之为最小关联性阈值。

3.3 多约束密度聚类模型

本文提出了在多约束条件下，进行密度聚类，是一个更加符合实际应用的情况。同时考虑到了障碍约束和用户约束，用户约束由用户指定约束的条件，指定了某些具有明显差异的数据，应归属于不同的簇中；而对于障碍约束，通过绕行障碍，得到数据间的距离，满足约束条件后实现聚类。

本文所提出的多约束密度聚类算法思想如图 1 所示。

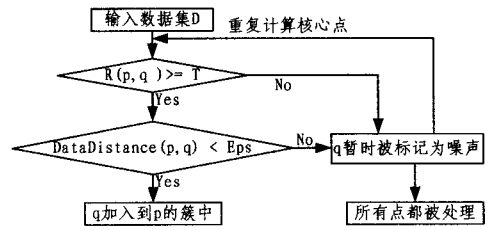

图 1 多约束密度聚类算法

多约束密度聚类模型分为以下 3 个部分。

（1）核心点与数据集中某点之间的关联度。本模型中，两点之间的关联度是通过用户约束的关联矩阵中对应的值来确定的。对应的关联度的值大于用户给定的最小关联性阈值，则认为有可能聚类在一个簇中；对应的关联度的值小于该阈值，则认为不可能聚类在一个簇中。

（2）核心点与数据集中某点之间的距离。若核心点和某一点之间的线形与障碍不存在线形相交，则认为其距离就是普通的欧几里得距离；若核心点和某一点之间的线段与障碍存在线形相交，则认为其距离为障碍距离，即绕行线形障碍的距离。

（3）核心点与数据集中某点的障碍下密度可达。若满足两点之间的关联度大于给定的阈值和两点之间的距离小于或等于 Eps，则认为两点之间障碍下密度可达，可以聚类在一个簇中。

4 多约束密度聚类算法实现

4.1 多约束的密度聚类算法

Input：a data set D，Eps，Minpts，a relative matrix R，Obstacle O(u_1，u_2)

Output：a set of cluster C={C_1，C_2，…，C_n} and noise data

1. for each dataPoint o∈D do
2. IsCore=RegionQuery()；

```
3. if(IsCore)
4.      a data set S
5.           =FindDensityReach(o);
6. for each Si do
7.    if(R(Si,o) >= T)
8.      //判断是否相交
9.      if(IsCross(Sio,u1u2))
10.        //相交则为障碍距离
11.        DataDistance(Si,o) =
12.          ObstacleDist(Si,o);
13.        if(DataDistance(Si,o)<=Eps)
14.              Si→o;
15.        else
16.              DataDistance(Si,o)=∞;
17. else
18.      o is marked temp noise;
19. repeat the above steps;
20. output C and the noise data;
```

上述的算法是针对用户约束和障碍约束而设计的算法过程;在多约束环境下,先找到核心点密度可达的点,再考虑用户约束的限制,主要是查询关联矩阵;其次,再考虑两点间是否存在障碍;若存在,则需要考虑到障碍距离。上述过程适应多约束条件下的数据点聚类分析。

5 算法的实验分析与比较

本算法实现的环境是 Visual Studio 2005,使用的语言是 C#;所有测试都是在一台 PC 机(CPU Pentium4,Memory 1G,Hard discs 80G)上进行的。实验的数据集来源于 WILEY 网站,选择的数据集为 2 Dimensional Cluster Text;本实验选取 2000 个数据点作为测试数据集。

和 DBSCAN 算法一样,本文的算法时间复杂度也是 $O(n * \log n)$(n为数据库中点的个数,对于每一个点至多进行一次区域查询,每次区域查询的时间至多为 $O(n * \log n)$)。先用 DBSCAN 算法,对数据点进行聚类分析,如图 2 所示。

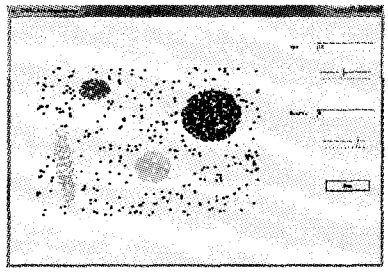

图 2 DBSCAN 算法的聚类结果

针对上面的聚类结果,加入了障碍进行约束,从而改变了聚类的结果,增加了聚类簇的数量,如图 3 所示。

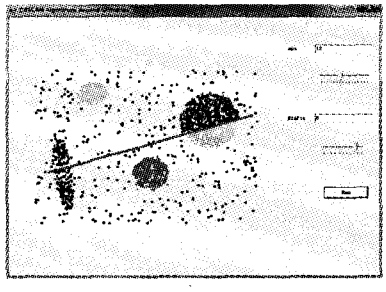

图 3 障碍约束下的聚类结果

针对 DBSCAN 算法所示的聚类结果,增加了用户约束的限制条件,从而改变了聚类的结果。人为地改变聚类的结果,使得距离相近的点聚类在不同的簇中,如图 4 所示。

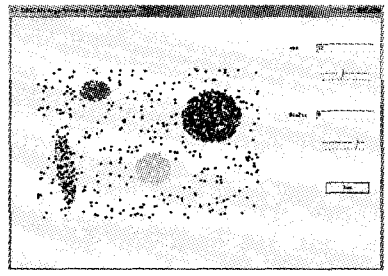

图 4 用户约束下的聚类结果

对于以上基于障碍约束和用户约束的情况,下面的聚类同时考虑到两种约束的情况,产生了不同于上述聚类的情况,聚类的结果是上述两种情况的共同作用的结果,如图 5 所示。

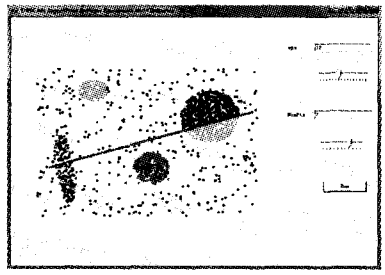

图 5 多约束下的聚类结果

上述实验中,图 2 是对数据点进行基于密度聚类的结果,图 3 和图 4 分别是基于障碍约束和用户约束聚类所得到的结果,而图 5 则是基于前两种约束条件下所得到的结果。实验表明文中的算法在多约束条件下,可有效完成对数据点的聚类并且效果较好,为现实情况中处理多约束聚类,提供了良好的理论支持。

结束语 本文提出了一种多约束条件下对数据集进行密度聚类的模型,在现实世界中对数据集限制的要求较多,文中的多约束模型更加符合现实情况。实验结果说明增加的约束条件对于聚类的结果产生了一定的影响。文中算法的时间复杂度和 DBSCAN 一样,进一步的研究需要减少时间复杂度,采用优化的区域查询算法来降低时间复杂度。同时,多约束也是以后研究的一个热点,它符合现实世界复杂多变的情况。

参 考 文 献

[1] Guha S, Rastogi R, Shim K CURE: An Efficient Clustering Algorithm for Large Database[C]// Proceeding of the ACM SIGMOD Conference. Seattle, 1998: 73-84

[2] Guha S, Rastogi R, Shim K. ROCK: A Robust Clustering Algorithm for Categorical Attributes[C]// Proceedings of the 15th ICDE. Sydney, 1999: 512-521

[3] Karypis G, Han E-H, Kumar V. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling[J]. IEEE Computer, 1999, 32(8): 68-75

[4] Ester M, Kriegel H-P, Sander J, et al. A Density-based Algorithm for Discovering Clusters in Large Spatial Database with Noise[C]// Proceedings of the 2nd ACM SIGKDD. Portland, 1996: 226-231

(下转第 164 页)

流。信息模型由相互关联的实体组成，能够为用户提供充足的信息，用来构建某一异常事件导出历史的跨越时间的结构图。

实体的世系信息分为如下几个部分：在数据流和查询注册器件收集到信息为静态世系，例如元信息是静态世系的一部分；在查询处理期间收集到的信息为动态世系，例如数据流流动速率的变化就是一种动态世系。

资源管理负责维护系统的当前状态，与查询计算进行交互。世系追踪服务对系统中当前资源的变化保持跟踪，同时提供 GUI 监控当前的查询、数据流和计算性资源。用户通过调用世系服务注册输入数据流和冰山查询。当提交新的查询时，系统产生导出流的注册。流/冰山查询注册之后，用户可以向世系数据集添加注释和元信息等附加信息。静态世系数据存储于世系知识库。

在执行期间，查询处理引擎将查询的更新发送到世系服务，当用户感兴趣的事件发生时，查询计划与查询执行引擎更新世系服务。用户感兴趣的事件包括：新查询的开始、由于流集减少或处理节点失败时查询执行计划的改变、某个已注册的查询过期、流速率变化、流/查询近似度与精度的改变等。

结束语 总之，目前缺少科学合理的流数据世系追踪模型，实现逆向跟踪 RFID 数据流的冰山查询存在较大难度。本文以 RFID 数据流的冰山查询中热门元素的数据世系追踪为研究对象，提出了面向 RFID 冰山查询的世系追踪模型。世系可以一直跟踪数据的读取到数据显示的整个过程。通过世系可以标注每一条原始数据流的来源和时间来准确定位异常发生的时间和位置，下一步将针对 RFID 数据流中的异常事件进行原子事件的世系追踪的检测与查询研究。

参 考 文 献

[1] Wang F, Liu P. Temporal management of RFID data[C]// Proceeding of the 31th International Conferance on Very Large Data Base(VLDB05). 2005:1128-1139

[2] Derakhshan R, Orlowska M E, Xue Li. RFID Data Management: Challenges and Opportunities[C]// Proceeding of IEEE International Conference on RFID 2007. March 2007:175-182

[3] Benjelloun O, Sarma A, Halevy A, et al. Uldbs: Databases with uncertainty and lineage[C]// Proceeding of the 32th International Conferance on Very Large Data Base(VLDB06). 2006

[4] Fang M, Shivakumar N, Molina H, et al. Computing Iceberg Queries Efficiently[C]// VLDB'1998. 1998:299-310

[4] Woodruff A, Stonebraker M. Supporting Fine-grained Data Lineage in a Database Visualization Environment[C]// Proc. of the Int' l Conf on Data Engineering(IEEE ICDE). 1997:91-102

[5] Charikar M, Chen K, Farach-Colton M. Finding Frequent Items in Data Streams[C]// Widmayer P, ed. Proceedings of the 29th International Colloquium on Automata, Languages and Programming. Malaga, Spain: Springer-Verlag, 2002:693-703

[6] Gibbons P B, Matias Y. New Sampling-based Summary Statistics for Improving Approximate Query Answers[J]. SIGMOD Record(USA), 1998, 27(2):331-342

[7] 金澈清，钱卫宁，周傲英. 流数据分析与管理综述[J]. 软件学报，2004，15(8)：1172-1181

[8] 聂国梁. 流数据统计算法研究[D]. 武汉：华中科技大学，2006：46-48

[9] 骆吉洲，李建中，赵锴. 大型压缩数据仓库上的 Iceberg Cube 算法[J]. 软件学报，2006，17(8)：1743-1752

[10] 潘定. 持续时态数据挖掘及其实现机制[M]. 北京：经济科学出版社，2008：156

[11] 高明，金澈清，王晓玲，等. 数据世系管理技术研究综述[J]. 计算机学报，2010，33(3)：373-389

[12] Blount M, Davis J, Misra A, et al. A time-and-value centric provenance model and architecture for medical event streams[C]// Proceeding of the 1st International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments. San Juan, Puerto Rico, June 11, 2007:95-100

[13] Wang Yong-li, Qian Jiang-bo, Ma Ran. RFIDSLT: A Data Lineage Tracing Method for Complex Query over RFID Streams[C]// ICEBE'09. Oct. 2009, 22:233-240

(上接第 145 页)

[5] Hinneburg A, Keim D. An Efficient Approach to Clustering Large Multimedia Databases with Noise[C]// Proceedings of the 4th ACM SIGKDD. New York, 1998:58-65

[6] Tung A K H, Hou J, Han J. Spatial Clustering in the Presence of Obstacles[C]// Proceedings of the 17th ICDE. Heidelberg, 2001:359-367

[7] Han J, Kamber M, Tung A K H. Spatial Clustering Methods in Data Mining: A Survey[C]// Geographic Data Mining and Knowledge Discovery. 2001

[8] Kohonen T. Self-Organizing Maps[M]. Springer Series in Information Sciences, 2001, 30

[9] Cao Yong-qiang, Wu Jian-hong. Dynamics of Projective Adaptive Resonance Theory Model: The Foundation of PART Algorithm [J]. IEEE Transactions on Neural Network, 2004, 15(2): 245-260

[10] Brown D, Huntley C. A Practical Application of Simulated Annealing to Clustering[R]. University of Virginia, 1991

[11] Tung P C, Zaki Y. CARPENTER: finding closed patterns in long biological data sets[C]// Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2003

[12] Ferhatosmanoglu H, Tuncel E, Agrawal D, et al. High dimensional nearest neighbor searching[J]. Information Systems, 2006, 31:512-540

[13] Zaiane O R, Lee C H. Clustering Spatial Data When Facing Physical Constraints[C]// Proc. of the IEEE International Conference On Data Mining. Maebashi City, Japan, 2002:737-740

[14] Li Yi-fan, Han Jia-wei. Clustering Moving Objects[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 200:492-110