基于网格化的变密度 DBSCAN 改进算法

唐志鹏 赵青 吴珺 刘文蔚

（中南大学 信息科学与工程学院，湖南省 长沙市 410012[1]）

指导教师 沈海澜 副教授

中文摘要：聚类分析是数据挖掘领域的重要课题之一。DBSCAN 算法是一种基于密度的聚类分析算法。然而 DBSCAN 算法有两个缺点：可伸缩性不强、无法识别变密度的簇。本文提出了一种基于网格化的变密度 DBSCAN 改进算法 CubeDBSCAN，通过将数据空间网格化从而解决 DBSCAN 可伸缩性不强的问题；通过建立同一个簇内各个点的密度相近的先验假设使算法能够识别变密度的簇。性能分析与仿真实验结果表明所提出算法能在多项式时间复杂度 *O(m)*解决上述两个问题。

英文摘要：Clustering analysis is a sub-subject of data mining. DBSCAN algorithm is a classical clustering algorithm based on density. However, the DBSCAN algorithm has poor scaling capability and cannot be used to recognize multi-density clusters. We proposed a multi-density DBSCAN algorithm based on gridding method to improve the two deficiencies. We improve the scaling capability of DBSCAN with gridding data space. We improve DBSCAN to recognize multi-density clusters with making priori hypothesis which the density of each point in the same cluster is roughly the same. Performance analysis and simulation experiments results shows the proposed algorithm can be solve the two weakness of DBSCAN with time complexity $O(m)$.

关键词：聚类分析；DBSCAN；网格化；可伸缩性；变密度

一、 引言

作为数据挖掘的重要研究课题之一，聚类挖掘的目的是将数据进行分组，使得组内数据（称为簇）的相似性尽可能大，组间数据的相似性尽可能小[1]。

从某种意义来说，簇的定义是难以精确化的，因此出现了基于明显分离原则、基于中心原则、基于图原则和基于密度原则等簇的定义方法。其中基于密度原则的簇被描述成“被低密度区域环绕的对象稠密区域”[2-3]。

不同的簇定义形成了不同的聚类算法，并能解决不同类型的问题。基于密度的簇的聚类算法通常用来在空间数据中发现被噪音包围的任意形状的模式。典型的基于密度的聚类算法包括 DBSCAN[4]、DENCLUE[5]和 OPTICS[6]等。

DBSCAN 算法是第一个基于密度的聚类算法，它通过不断在簇的周围接纳符合要求的

[1]**作者简介：**唐志鹏（1993-），男，籍贯湖南省，计算机科学与技术专业，本科三年级，研究方向：数据挖掘。赵青（1992-），男，籍贯广西省，计算机科学与技术专业，本科四年级，研究方向：数据挖掘。吴珺（1993-），女，籍贯安徽省，计算机科学与技术专业，本科三年级，研究方向：数据挖掘。刘文蔚（1994-），男，籍贯湖南省，计算机科学与技术专业，本科三年级，研究方向：机器学习。

点的方法来形成聚类。然而传统的 DBSCAN 算法有两个不足：（1）运算开销大，可伸缩性不强，在大数据环境下不能很好地运行；（2）无法识别数据集中的变密度的簇。

本文针对上述两个问题，提出了一种基于网格化的变密度 DBSCAN 改进算法。通过将数据空间网格化从而解决可伸缩性的问题；同时利用“同一个簇内各个点的密度相近”的先验假设使算法能够识别变密度的簇。

二、 相关研究

DBSCAN 是第一个被提出的基于密度的聚类算法[4]，其他常用的基于密度的聚类算法还有 DENCLUE[5]、OPTICS[6]等。有许多关于 DBSCAN 的改进算法，文献[7]提供了很好的综述。

文献[8]和[9]试图通过网格化的方式优化 DBSCAN 算法。前者首先将数据尽量切分成密度相似的网格并将密度相同的网格合并，然后通过网格的密度来确定Eps和$MinPts$，用它们调用 DBSCAN 聚类；后者基于分治策略，将数据集网格化后分别聚类，最后又合并得到全局聚类结果。

文献[10][11]和[12]改进了 DBSCAN 算法使它能够对存在变密度的簇的数据集进行聚类。文献[10]提出了 VDBSCAN 算法，通过 K-dist 图和 DK 分析，对数据集中的不同密度层次自动选择一组Eps值分别调用 DBSCAN 算法，进而达到分辨变密度的簇的效果。文献[11]提出的 GMDBSCAN 算法定义了通过建立空间索引及$local\ MinPts$的参数来聚类。文献[12]使用 K-近邻(K^{th} nearest neighbors)的平均值来表示密度，通过先搜索密度最大的点来加快聚类速度并且不需要输入参数。

上述已有方法中，主要从提高运算效率改进 DBSCAN 算法，而如何在优化运算效率的同时解决可变簇问题，需要进一步研究。

三、 基于网格化的变密度 DBSCAN 改进算法

（一）DBSCAN 算法和变密度簇的问题

为阐明 DBSCAN 算法的基本思想思想，我们先给出几个相关定义以及基本 DBSCAN 算法代码：

定义 1（点的邻域）：以该点为圆心、以Eps为半径的圆所包围的区域称为该点的邻域。

定义 2（点密度Di）：指该点的邻域内所包围的点的个数(包括该点本身)。

定义 3（核心点）：若点的密度大于等于 *MinPts*，则此点为核心点。

定义 4（边界点）：非核心点，但是落在某个核心点的邻域内。

定义 5（噪声点）：既非核心点也非边界点的点。

对数据中的某个核心点，DBSCAN 将此点及其邻域内的边界点包含进一个新的簇中，并寻找此点邻域内的其它核心点，递归地将它们加入到簇中。DBSCAN 算法的伪代码如下描述：

Algorithm ***DBSCAN(D, Eps, MinPts, C)***
Begin
  For *point k* in *D*
    If *k is a core point* ***and*** *has't been assigned to any cluster* ***then***
      Creat a new cluster C_j
      AddIntoCluster(C_j,*k*)
      Extend C_j*-Recursively*(C_j , *k*)
      j=j+1
End

Algorithm ***Extend*** C_j***-Recursively***(C_j , *k*)
Begin
  For *point h in neighborhood of k*
    If *h hasn't been assigned to any cluster* ***then***
      AddIntoCluster(C_j,*h*)
      If *h* is a core point ***then***
        Extend C_j*-Recursively*(C_j , *h*)
End

Algorithm *AddIntoCluster*(C,t)
Begin
  C=C+t
  Set the flag of t to indicate it has been assigned to a cluster
End

由于 DBSCAN 算法对自然簇的提取使用了一种绝对的标准，当Eps一定时，$MinPts$只要大于某阈值则点将被视为核心点。这也导致其并不能很好地处理存在变密度的簇的数据集。示例如图 1 所示。

如图 1 所示，此数据集形如三个密度从外到内越来越密的同心圆，显然这个数据集中包含三个自然簇。当Eps一定时，若$MinPts$小的话，则最内层的两个自然簇会被聚类成一个簇；若$MinPts$大的话，则最外层的两个自然簇会被聚类成一个簇。无论如何调整输入参数，均无法得到想要的结果。

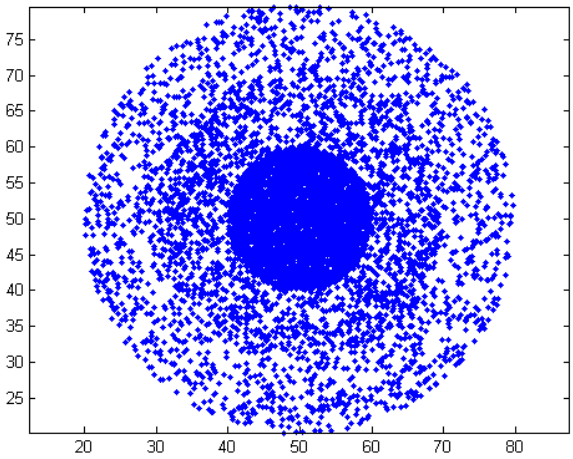

图 1 变密度簇问题

（二）CubeDBSCAN 算法

算法的基本思想是将数据空间切分成 n 维立方体，统计各个立方体内所包含的数据点的数量，记为立方体密度。以立方体为单位，根据立方体密度大小，采用 DBSCAN 算法进行聚类。基于网格化的变密度 DBSCAN 改进算法的自然语言描述如算法 CubeDBSCAN 所示：

Algorithm *CubeDBSCAN (D,gridsize,Eps)*

Input：数据集 $D=\{x_1,x_2,...,x_m\}$，其中 $x_i=(d_1^{(i)},d_2^{(i)},...,d_n^{(i)})$，$d_j^{(i)}\geq 0$

gridsize：用于调整网格大小的参数

Eps：用于调整簇的密度允许变化大小的参数

Output：聚类结果 *C*

Begin

Step1 （特征归一化）

对输入的数据集*D*中的每一个数据点 $x_i(1\leq i\leq m)$ 通过以下公式归一化，使 x_i 的每一个维度 $d_j^{(i)}(1\leq j\leq n)$ 的值在[0,1]范围内，新的 x_i' 组合成数据集 D'，所有的数据点均位于一个n维立方体的空间中。

$$d_j'^{(i)}=d_j^{(i)}/\max(d_j^{(1)},d_j^{(2)},...,d_j^{(m)})(1\leq \mathrm{j}\leq \mathrm{n})$$

Step2 （网格化）

根据参数 *gridsize*，将数据空间切分成n维立方体（用$(k_1,k_2,...,k_n)$表示某一具体立方体）。统计各个立方体内所包含的数据点的数量，记为立方体密度$P_{(k_1,k_2,...,k_n)}$。

Step3 **(消除噪声)**

若立方体的密度过小，则将其并入噪声簇中。

Step 4 **(聚类)**

Step 4.1

从所有尚未被归入某簇的立方体中找到立方体密度最大的T，建立一个新的簇C_j，将T并入簇C_j之中，并将T的立方体密度记为P_T。

Step 4.2

调用 DFS_ $gather(C_{j,}P_T,Eps,T)$ 或 *BFS_* $gather(C_{j,}P_T,Eps,T)$，其中 *DFS_gather* 是基于深度优先搜索，*BFS_gather* 是基于广度优先搜索的。后者更加适应在大数据环境下进行聚类。

End

Algorithm *DFS_* $gather(C_{j,}P_T,Eps,T)$

Begin

对 *T* 周围的每个立方体 *R*，若 *R* 尚未被归入某簇中，则将它并入簇 C_j 中；若 $P_R \geq P_T - Eps$，则进而调用 DFS_ $gather(C_{j,}P_T,Eps,R)$

End

Algorithm *BFS_* $gather(C_{j,}P_T,Eps,T)$

Begin

创建一个队列Q，将立方体T加入队列Q中。

从Q中取出一个立方体R，将 R 的邻域中尚未归入某簇的立方体全部并入簇C_j中，并将使$P_R \geq P_T - Eps$成立的立方体加入队列Q中。

End

四、 性能分析与仿真实验

对 *Grid_DBSCAN* 算法进行时间复杂度分析，步骤 1、2 和 3 可以在$O(m)$的时间求得；步骤 4 可以在$O(n)$时间内求得，其中n是立方体数量。由于n和m之间是常数倍关系，所以算法可以在$O(m)$的时间内得到聚类结果，总的时间复杂度可以用$O(m)$表示，其中m是数据点个数。

以下使用由 MATLAB 实现的程序来对此算法进行实验评估。数据是使用均匀分布随机产生几个不同大小的圆形簇，它们密度相近并相互叠加，从而在叠加处产生密度不同的非规则图形簇，另外我们还在空间中加入了稀疏的噪声点,如图 2 所示。

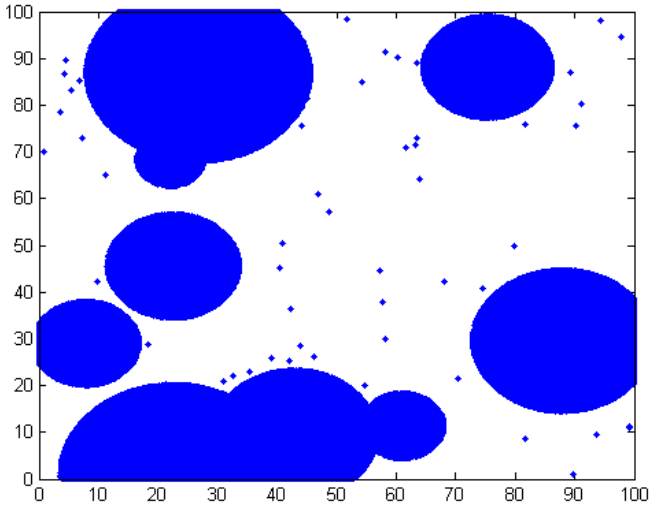

图 2 原始数据集（由于数据量大，点相互叠加导致圆形簇已被填充）

图 3 所示是用某次实验中随机产生的数据集放大后形成的，它表示两个圆形相交的部分。很明显，这个数据集中间密、两边疏，左右两边的密度并不均等，总体应该被分成三个簇。图 4 说明了算法的聚类结果，它通过颜色标明网格所属的簇（网格内的点将归入该簇中），不同的颜色表明不同的簇。可以看出，聚类效果是十分可观的。

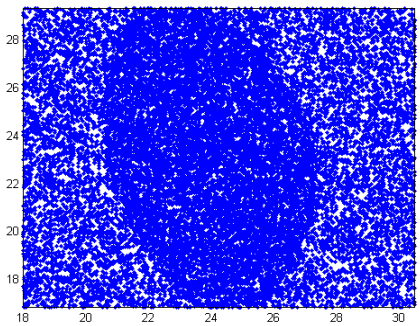

图 3 某次实验中数据集

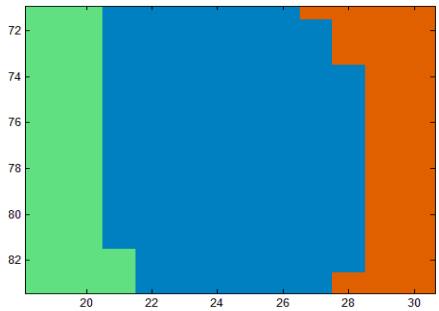

图 4 图 3 所示数据的聚类结果

类似的结果如图 5 所示，这是在两个大圆的簇相交边界处有一个小圆的簇的情况，它的聚类结果如图 6 所示。这一样例同样说明了此算法的有效性。

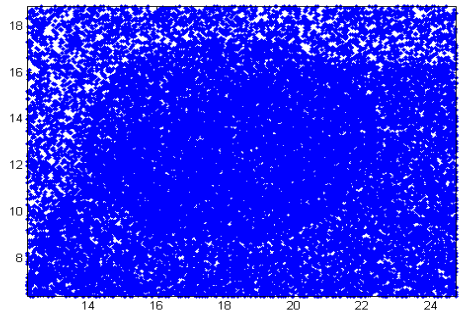

图 5 在两个大圆相交处有一个小圆的情况

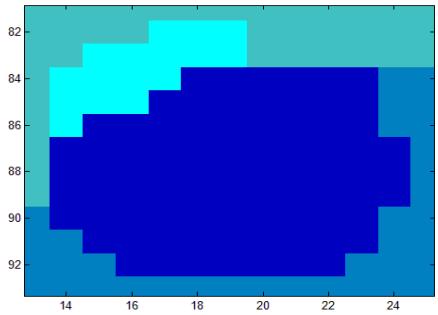

图 6 图 5 的聚类结果

五、 结束语

DBSCAN 算法是聚类分析中的重要算法之一，然而传统的 DBSCAN 算法存在可伸缩性不强、无法识别变密度的簇的缺点。本文提出了一种基于网格化的变密度 DBSCAN 改进算法，它通过网格化的方式减小运算代价，通过设置簇的密度变化阈值来区分不同密度的自然簇，从而改善了解决了这两个缺点。实验证明，此算法能够得到较好的聚类结果。

然而此算法仍然存在以下两个不足之处：由于网格化的影响，簇的边界处聚类结果模糊。需要合理设置$gridsize$和Eps两个参数。如何解决这一问题是未来的研究工作之一。

参考文献

[1] Grabmeier, Johannes, and Andreas Rudolph. "Techniques of cluster algorithms in data mining." Data Mining and Knowledge Discovery 6.4，2002: 303-360.

[2] Tan, Pang Ning, Kumar Steinbach, and Vipin Kumar. "Data Mining Cluster Analysis: Basic Concepts and Algorithms." (2006).

[3] 孙吉贵，刘杰，赵连宇.聚类算法研究[J], 软件学报, 2008,19(1): 48-61.

[4] Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise." In Kdd, vol. 96, no. 34, pp. 226-231. 1996.

[5] Hinneburg, Alexander, and Hans-Henning Gabriel. "Denclue 2.0: Fast clustering based on kernel density estimation." In Advances in Intelligent Data Analysis VII, pp. 70-80. Springer Berlin Heidelberg, 2007.

[6] Ankerst, Mihael, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. "OPTICS: ordering points to identify the clustering structure." In ACM Sigmod Record, vol. 28, no. 2, pp. 49-60. ACM, 1999.

[7] Parimala, M., Daphne Lopez, and N. C. Senthilkumar. "A survey on density based clustering algorithms for mining large spatial databases." International Journal of Advanced Science and Technology 31.1 (2011).

[8] 刘淑芬，孟冬雪，王晓燕. 基于网格单元的 DBSCAN 算法[J]. 吉林大学学报：工学版 2014: 1135-1139.

[9] 李双庆，慕升弟.一种改进的 DBSCAN 算法及其应用[J],计算机工程与应用，2014， 50（8）： 72-77

[10] Liu, Peng, Dong Zhou, and Naijun Wu. "VDBSCAN: varied density based spatial clustering of applications with noise." In Service Systems and Service Management, 2007 International Conference on, pp. 1-4. IEEE, 2007.

[11] Xiaoyun, Chen, Min Yufang, Zhao Yan, and Wang Ping. "GMDBSCAN: multi-density DBSCAN cluster based

on grid." In e-Business Engineering, 2008. ICEBE'08. IEEE International Conference on, pp. 780-783. IEEE, 2008.

[12] Liu, Qing-Bao, Su Deng, Chang-Hui Lu, Bo Wang, and Yong-Feng Zhou. "Relative density based k-nearest neighbors clustering algorithm." In Machine Learning and Cybernetics, 2003 International Conference on, vol. 1, pp. 133-17. IEEE, 2003.