改进 *k* 值自动获取 VDBSCAN 聚类算法①

赵文冲, 蔡江辉, 张继福

(太原科技大学 计算机科学与技术学院, 太原 030024)

摘 要: 针对 DBSCAN 聚类算法不能对变密度分布数据集进行有效聚类, VDBSCAN 算法借助 *k*-dist 图来自动获取各个密度层次的数据对象的邻域半径, 解决了具有不同密度层次分布数据集的聚类问题. *k*-VDBSCAN 算法通过对 *k* 值的自动获取, 减小了 VDBSCAN 中参数 *k* 对最终聚类结果的影响. 针对 *k* 值的自动获取, 在原有的 *k*-VDBSCAN 聚类算法基础上, 依据数据集本身, 利用数据对象间距离的特征, 提出了一种 *k* 值改进自动获取聚类算法. 理论分析与实验结果表明, 新的改进算法能够有效的自动获得参数 *k* 的值, 并且在聚类结果、时间效率方面都有明显的提高.

关键词: 变密度; VDBSCAN; *k* 值获取; *k*-VDBSCAN; 点间距离特征

Based on Improved Parameter *k* Chosen Automatically in VDBSCAN Clustering Algorithm

ZHAO Wen-Chong, CAI Jiang-Hui, ZHANG Ji-Fu

(School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China)

Abstract: For DBSCAN algorithm can't cluster some variable density data sets effectively, VDBSCAN solved this question by a *k*-dist figure to automatically obtain the neighborhood radiuses of various density levels of data objects. *k*-VDBSCAN algorithm obtains the parameter k automatically to reduce the parameter *k* 'sinfluence in the final clustering results . Based on the data set itself, using the characteristics of the distance between the data objects , on the basic of the *k*-VDBSCAN algorithm, an clustering algorithm based on improved parameter *k* obtained automatically is proposed. Theoretical analysis and experimental results show that the improved algorithm can effectively automatically obtain the value of the parameter *k* and the clustering results and time efficiency has improved significantly.

Key words: variable density; VDBSCAN; parameter *k* obtained; *k*-VDBSCAN; distance feature

聚类[1]是数据挖掘和数据分析领域的一项重要任务, 它是一系列对象的分配组合. 聚类分析[2]就是根据数据对象间的相似度, 将数据对象划分成簇, 并使得相同簇中的数据对象之间距离尽可能近、信息相似度尽可能高, 不同簇中的数据对象之间距离尽可能远、信息差异性尽可能大的过程. 目前, 存在着大量的聚类算法, 大致可以分为层次化聚类算法、划分式聚类算法、基于密度和网格聚类算法以及其他聚类算法等几大类[3].

DBSCAN[4](Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise)是一种经典的、使用最广泛的基于密度聚类算法. 同时由于算法中的参数具有全局性, 对于分布不均匀、局部数据对象子集密度层次存在差异(即变密度)的数据集, 适应性不好, 甚至难以有效聚类. 针对该问题, Jahirabadkar S 等人[5]利用子空间采用两阶段对对象邻域半径 *Eps* 进行选择, 改进了全局参数对变密度数据集聚类的困难; Cassisi C 等人[6]采用了基于影响函数 INFLO[7]的空间分层概念对 DBSCAN 算法中的输入参数进行了限制, 能够很好的处理变密度数据集聚类以及去除数据集中的离群点问题; Zhou 等人[8]提出了一种利用 *k*-plot 数据结构自动获取数据集中各个密度层次的邻域半径 *Eps* 的 VDBSCAN(Varied Density Based Spatial Clustering of Applications with Noise)算法.

① 收稿时间:2015-12-25;收到修改稿时间:2016-02-25 [doi:10.15888/j.cnki.csa.005325]

VDBSCAN 是一种较好的变密度聚类算法, 但参数 k 的选择会对最终聚类结果产生很大影响. 本文在已有的 k 值自动获取 VDBSCAN 聚类算法——k-VDBSCAN[9]基础上, 提出了一种改进 k 值自动获取的 VDBSCAN 聚类算法——k-LVDBSCAN(Local Distance-Based for Varied Density Based Spatial Clustering of Application with Noise). 该算法不需要用户参与情况下, 能够自动获得适合当前数据集的参数 k, 最终得到有效、高质量的聚类结果.

1 相关理论算法

1.1 DBSCAN

DBSCAN 是一种基于密度的聚类算法. 该算法将具有足够高密度的区域划分为簇, 并且在具有噪声的空间数据库中发现任意形状的簇[10].

算法 DBSCAN 的目的是为了发现带有噪声的空间数据簇[11], 算法的主要步骤为: 1)任意选择数据集中的一个对象 p, 并从该对象开始检索从 p 关于 Eps 密度可达的所有对象; 2)如果 p 是一个核心(core)对象, 则建立一个关于 Eps 和 $Minpts$ 的新簇; 3)如果 p 仅仅是一个边缘(border)对象, 则没有对象是从 p 密度可达的, DBSCAN 就会访问数据集中的下一个对象点; 4)重复以上过程, 直到所有的数据对象都被访问过为止.

尽管 DBSCAN 算法能够发现任意形状的簇, 以及对噪声数据具有很好的适应性等优势. 但仍然存在很多缺陷, 其中最主要的缺陷[12]: 首先, 它需要用户输入制定的参数值, 而参数的设定非常困难; 其次, 它在分布不均、有不同密度层次的数据集中进行有效聚类是非常困难的, 单纯的设置全局参数 Eps, 极大可能造成结果的严重偏差. 算法 VDBSCAN 正是针对上述问题对传统 DBSCAN 算法的一种改进算法.

1.2 VDBSCAN

VDBSCAN 算法针对 DBSCAN 算法中参数 Eps 由用户输入, 且为全局性的, 对分布不均、多个密度层次的数据集, 聚类效果较差的问题, 通过求解数据对象的 k 近邻邻域半径 k-dist, 并构造 k-dist 图, 能够根据不同的密度层次自动得到相对应的 Eps 值, 而后对每一个 Eps 迭代施行 DBSCAN 算法, 发现不同密度的簇, 达到有效聚类、产生更好的聚类结果.

图 1 为文献[13]中一个简单实例的 k-dist 图. 分析可得: a、c、e 为三条平缓曲线, 分别代表了一个密度层次; b、d 为密度转折线, 它们分别连接两条平缓曲线; f 为异常点曲线, 它只连接了一条平缓曲线. 故曲线 B 所代表的数据集中有三个密度层次, b、d、f 分别对应一个相应的 Eps, 而曲线 A 所代表的数据集比较均匀, 它只有一个密度层次.

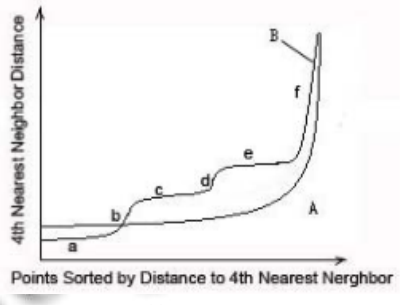

图 1　一个简单实例的 k-dist 图

VDBSCAN 算法主要有两步: 第一步, 选择合适的参数 Eps_i; 第二步, 根据得到的各个层次的 Eps_i 进行聚类. 具体步骤为: (1)任选一个正整数 k, 计算数据集中各个数据点的第 k 个最近邻数据点到该数据点的距离 k-dists; (2)对 k-dists 进行升序排序, 并作 k-dist 图, 得到平缓曲线和非平缓曲线; (3)选取各平缓曲线所对应的 Eps, 并将这些 Eps_i 升序排序; (4)将 k 值赋值给 $Minpts$, 通过得到的排好序的 Eps_i 迭代进行 DBSCAN 聚类.

VDBSCAN 算法适用范围更加广泛, 尤其对于分布不均、变密度的数据集, 适用性更强. 但同时 k 值的引入以及选择, 对最终聚类结果的影响同样很大. k 值的合理选择, 对于能够得到好的聚类结果非常重要, 但 k 值的选择, 往往不令人满意.

1.3 k-VDBSCAN

为了寻找更适合数据集的 k 值, A.K.M Rasheduzzaman Chowdhury 等人[9]提出了用数据集本身的特性自动的产生 k, 而后再使用 VDBSCAN 的一种改进算法——k-VDBSCAN.

算法具体实现描述:

1) 假设二维空间中有 n 个数据点, 将每个数据点 $p_i(i=1,2,\ldots,n)$ 都视为目标点, 并计算目标点到其他所有点之间距离的平均值.

$$d(p_i)=\frac{\sum_{i=1}^{n} distance(p_i,x_i)}{n-1} \tag{1}$$

2) 计算全局平均值.

$$avg(d)=\frac{\sum_{i=1}^{n} d(p_i)}{n} \tag{2}$$

3) 以 avg(d)为半径, 各 p_i 为圆心画圆, 可以得到仅仅位置不同的 n 个完全相同的圆.

4) 以下列公式寻找距离这些圆周最近的点的位置

$$\min |\text{distance}(r\text{-}X_i)| \quad (3)$$

将 X_i 视为所要寻求的关于 p_i 相对应的目标点, 并标记为 T_i.

5) 得到所有数据点 p_i 的 T_i 的位置坐标 $T_i(pos)$.

6) 通过所有数据点 p_i 的 $T_i(pos)$, 我们可以很清楚的得到 $T_i(pos)$的众数, 即 $T_i(pos)$的最大重复次数, 而这个众数就是我们所要得到的 k 值

7) 进行 VDBSCAN 算法.

通过上述方法, 自动生成 k 值, 执行 VDBSCAN 算法, 可以对数据集进行有效聚类, 聚类质量得到很大改善, 但是此算法是依靠的是所有数据点之间的距离, 得到的全局性的 avg. 在分布不均匀的数据集中, 不仅计算量大, 而且未充分利用数据集本身的特性, 极大可能将全局性的avg放大, 致使得到的k值在最终聚类结果上仍然存在聚类质量较低的问题.

2 k-LVDBSCAN聚类算法

2.1 算法思想

假设在某个数据集中, 数据点的分布是不均匀的, 就会形成数据点之间的距离变化差异非常大. 图 2 显示了一个基本例子, 设Δd_k=(k+1)-dist-k-dist, 对于点 p, 可以明显发现: 当 k≤4 时,Δd_k 变化很小; 但Δd_5(即 6-dist-5-dist)变化幅度很大, 其中 p 的 k-dist 以及各Δd_k表现在图 3 中. 则我们可以得到, 点 p 以及其最邻近的 5 个点极大可能处于同一个簇中, 而与点 o 和点 q 极大可能处于不同的簇中. 这样, 在 k-LVDBSCAN 算法求全局 avg 过程中, 我们不需要计算目标点到其他所有点之间的聚类的平均值, 而仅仅对于可能处于相同簇中的点, 求取它们距离的平均值, 所求得的 avg, 对于数据集所想得到的 k 值来说, 会更加精确.

图 2 数据点的分布情况

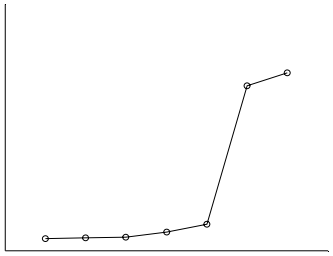

图 3 点 p 的 k-dist 图

2.2 算法描述

局部距离求参数 k 的方法进行 VDBSCAN 聚类算法 k-LVDBSCAN 具体描述:

输入: 有 n 个数据点的数据集 DB.

输出: 参数 k 的值.

具体步骤:

1) for each data object $p_i \in DB$ do

2) for each data object $p \in DB$

3) calculating the distance D_i do;

/* D_i 是 p_i 到 DB 中各个数据点的距离矩阵 */

4) end

5) end

6) sorting the $D_i \rightarrow DD_i$;

7) for every vector D_i do

8) checking out and record the S_i;

/* S_i 是各个距离向量中相邻的两个元素之间变化激烈的前一个元素序号 */

9) for 1 to S_i do

10) calculating the summary Q_i;

/* Q_i 是各个向量 D_i 的前 S_i 个元素的值和 */

11) end

12) end

13) calculating the average of Q_i

14) $$avg=\frac{\sum_{i=1}^{n} Q_i}{n}$$

15) searching the position X_i,for

$$\min |\text{distance}(r\text{-}X_i)|;$$

/* 寻找能够满足公式的每一个数据对象 p_i 相应的点 X_i,其中 r=avg*/

16) getting the $T_i(pos)$ of every X_i;

/* $T_i(pos)$是 X_i 的位置坐标 */

17) abtaining the mode of $T_i(pos)$;

18) determining the k;

/*k 即为 the mode of $T_i(pos)$*/

19) running the VDBSCAN algorithm

需要注意的是,由于在求得 k 值过程中, 已经对求得的各个数据点的距离排好序, 所以在 VDBSCAN 过程中, 我们可以直接从画 k-dist 图开始.

2.3 算法性能理论分析

2.3.1 聚类质量分析

聚类质量的高低是衡量一种聚类算法好坏的重要指标. 对 DBSCAN 算法、VDBSCAN 算法、k-VDBSCAN 算法以及本文提出的算法 k-LVDBSCAN 进行分析比较, 我们可以得到:

性质 1. VDBSCAN 聚类质量不低于 DBSCAN, k-VDBSCAN 算法的聚类质量不低于 VDBSCAN, 而本文所提出的改进算法的聚类质量要不低于于 k-VDBSCAN.

理论分析: 由 1.1 节可知, 当数据集中密度水平恒定, 即单个 *Eps* 适用者整个数据集时, 只要合适选取 *Minpts*, 算法 DBSCAN 是非常有效的. 如果数据集中的某些簇处于不同的密度层次, 此时, 若只选用一个全局的 *Eps*, 聚类效果可能不佳, 即处于同一簇中的数据点,由于该簇的邻域半径大于 *Eps*, DBSCAN 算法就将它们视为噪声, 而不将它们进行聚类. 算法 VDBSCAN 通过得到一个或几个 *Eps*, 按升序对数据集进行 DBSCAN 聚类, 这样可以更接近得到变密度下的全部簇, 达到更有效更准确的聚类. 算法 k-VDBSCAN 从数据集本身出发,通过数据集本身特性, 自动获取得到一个比较合适的 k 值. k-VDBSCAN 主要考虑的是所有数据点之间的距离,从而得到一个全局的平均值 *avg*, 利用该 *avg* 进行 k 值的选取, 显然, 对分布不均的数据集可能不适合. 我们提出的利用局部数据点, 即有可能最终聚集为同一簇的数据点间的距离,得到局部 *avg*, 再求 k 值, 无疑可以得到更加合适的 k 值, 而对分布比较均匀的数据集, 它将退化为 k-VDBSCAN.

2.3.2 聚类效率分析

分析比较算法 k-VDBSCAN 以及算法 k-LVDBSCN 时间效率, 可以得到:

性质 2. 改进算法 k-LVDBSCAN 算法自动求 k 值过程, 时间效率要优于原 k-VDBSCAN 算法.

理论分析: 算法中第 1-5 行为第一部分, 求解的是数据集中所有数据对象间的距离, 故时间复杂度为 $O(n^2)$; 第 6 行为对距离矩阵中每一行的数据进行排序, 采用快速排序策略, 则计算复杂度为 $O(n\log n)$; 第 7-12 行为获取距离矩阵 D_i 中相邻数据变化极大的数据标号 S_i, 并存储下来, 需要花费 n 个空间记录这些标号, 随后求取每个数据对象 p_i 所对应的距离矩阵 D_i 中的前 S_i 个数据之和, 计算复杂度为 $O(n \cdot S_i)$, 其中 $S_i \le n$; 13-14 行求取平均值, 即圆半径, 时间复杂度 $O(n)$; 15-18 行为寻找求得最终的 k 值, 时间主要消耗在 15 行中的最小位置查找上, 时间复杂度为 $O(n^2)$. 通过上述理论分析, 改进的 k 值选择算法自动求取 k 值步骤的时间复杂度为 $O(n^2)+O(n\log n)+O(n \cdot S_i)+O(n)+O(n^2)$, 其中 $S_i \le n$. 而 k-VDBSCAN 算法求取 k 值步骤主要分为 4 部分, 没有对距离矩阵 D 的排序, 因此时间复杂度为 $O(n^2)+O(n\log n)+O(n)+O(n^2)$, 但在 VDBSCAN 中同样要进行排序, 因此时间复杂度 $O(n\log n)$不可缺少. 因为 $S_i \le n$, 所以我们的改进算法在时间效率上要优于原 k-VDBSCAN 算法.

3 实验结果及分析

为了进一步说明 k-LVDBSCAN 算法, 本节利用实验形式对算法性能进行测试. 实验平台配置为 2.3GHz Intel core i3 2310M CPU, NVIDIA GeForce GT 520M 显卡, 2GB DDR3 内存, 320G 硬盘的 acer PC 机, 算法程序采用 MATLAB 2010b 编写. 实验中, 分别使用人工数据集以及 UCI、USPS 等数据集作为测试数据集.

在聚类质量方面, 文献[14]提供了一种评价聚类结果好坏的标准. 度量标准为

$$Micro\text{-}precision=\frac{1}{n}\sum_{i=1}^{k}\alpha_i \tag{4}$$

其中, n 为数据集数据对象个数, k 为聚类的簇数, α_i 为聚类的簇 i 与已知数据集类别对应后, 簇 i 中被正确归为相应类别的样本个数. *Micro-precision* 的值越大, 表示算法在该数据集上聚类效果越好.

3.1 人工合成数据集

图 4 为利用文献[15]所述方法构造的二维数据集, 它由 3 个簇构成, 各簇中的数据点均满足中心点分别为(4,6)、(9,3)、(10,11), 方差分别为 0.4、0.8、1.2 的高斯分布.

图 5 为利用 VDBSCAN 算法进行聚类, 参数 k 的变化对最终聚类结果产生的影响, 实验中, 我们分别取 k 值为 10,20,30,40,50. 由图可以清晰看到, 不同的 k 值, 产生不同的最终聚类结果, 而且, 在某一小区域内的变化, 会对聚类结果产生很大的影响.

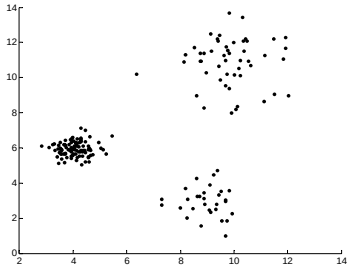

图 4 人工合成数据集

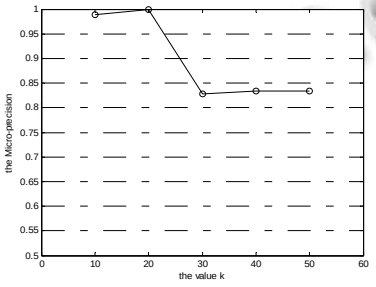

图 5 人工数据集中 k 值变化的影响

表 1 为采用 k-VDBSCAN 算法以及 k-LVDBSCAN 算法对人工数据集求 k 并聚类得到的最终结果.

表 1 两个算法在人工数据集上的对比

算法	k 值	簇数	*Micro-precision*(%)
k-VDBSCAN	36	2	83.33
k-LVDBSCAN	22	3	99.44

3.2 真实数据集

表 2 为实验中所使用的 UCI、USPS 等真实数据集.

表 2 测试数据集的特征

数据集	维数	数据个数
Perfume	2	560
4k2_far	2	400
Haberman	3	306
iris	4	150

表 3 k-VDBSCAN 在各数据集上的聚类结果 *precision*

算法	数据集	k 值	簇数	*precision*(%)
	Perfume	84	4	20
k-VDBSC	4k2_far	100	2	74
AN	Haberman	7	2	72.22
	iris	17	2	66.67

表 4 k-LVDBSCAN 在各数据集上的聚类结果 *precision*

算法	数据集	k 值	簇数	*precision*(%)
	Perfume	28	9	45
k-LVDBSCAN	4k2_far	28	4	100
	Haberman	10	2	72.88
	iris	8	2	66.67

分析表 3、表 4 和图 6 可得, 利用算法 k-LVDBSCAN 得到的最终聚类结果要明显优于算法 k-VDBSCAN, 这就表明利用 k-LVDBSCAN 得到的 k 值要比利用 k-VDBSCAN 得到的 k 值, 在 VDBSCAN 进行聚类时要更加适合, 从而证明我们的改进算法是有效且可行的.

通过实验对算法的效率分析, 通过图 7, 可以明显得到, 数据集 4k2_far、Haberman、iris 下, 两种算法迭代次数相同, 即得到的 *Eps* 个数相同的情况下, 算法 k-LVDBSCAN 的执行时间都要比 k-VDBSCAN 算法的执行时间低, 说明算法 k-LVDBSCAN 拥有比 k-VDBSCAN 更高的时间效率.

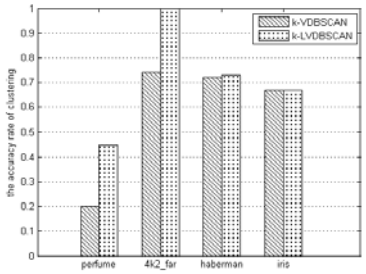

图 6 各数据集的聚类精度

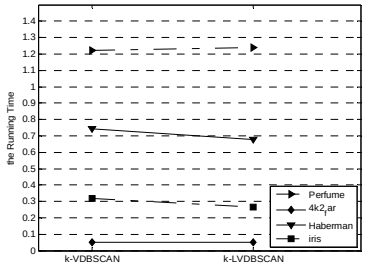

图 7 不同算法在各个数据集的执行时间

4 结语

本文针对基于密度的聚类方法 DBSCAN、VDBSCAN 中普遍存在要求用户输入参数, 且参数的选取对聚类结果产生重大影响, 甚至导致比较差的聚

类结果的问题. 利用数据集本身数据点间的距离的特性, 将局部距离代替已存在的 *k*-VDBSCAN 算法中全局距离, 能够自动的生成 *k* 值. 理论分析和实验结果表明, 改进以后的算法 *k*-LVDBSCAN 能够得到合适的 *k* 值, 而且在聚类结果、时间效率上都要优于 *k*-VDBSCAN, 算法是有效可行的.

参考文献

1 Kaur MNK. Review paper on clustering techniques. Global Journal of Computer Science and Technology, 2013, 13(5).

2 Jain AK, Murty MN, Flynn PJ. Data clustering: A review. ACM Computing Surveys (CSUR), 1999, 31(3): 264–323.

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

4 Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. Proc. of 2nd International Conference on Knowledge Discovery and Data Mining(KDD). AAAI Press. 1996. 226–231.

5 Jahirabadkar S, Kulkarni P. Algorithm to determine ε-distance parameter in density based clustering. Expert Systems with Applications, 2014, 41(6): 2939–2946.

6 Cassisi C, Ferro A, Giugno R, et al. Enhancing density-based clustering: Parameter reduction and outlier detection. Information Systems, 2013, 38(3): 317–330.

7 Jin W, Tung AKH, Han J, et al. Ranking outliers using symmetric neighborhood relationship. Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg, 2006: 577–593.

8 Liu P, Zhou D, Wu N. VDBSCAN: Varied density based spatial clustering of applications with noise. 2007 International Conference on Service Systems and Service Management. IEEE. 2007. 1 – 4.

9 Chowdhury AKMR, Mollah ME, Rahman MA. An efficient method for subjectively choosing parameter ‘k’ automatically in VDBSCAN (Varied Density Based Spatial Clustering of Applications with Noise) algorithm. 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE). IEEE. 2010. 38 – 41.

10 Jiawei H, Kamber M. Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann, 2006.

11 Parimala M, Lopez D, Senthilkumar NC. A survey on density based clustering algorithms for mining large spatial databases. International Journal of Advanced Science and Technology, 2011, 31(1).

12 Khan K, Rehman SU, Aziz K, et al. DBSCAN: Past, present and future. 2014 Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT). IEEE. 2014.232–238.

13 周董,刘鹏.VDBSCAN:变密度聚类算法.计算机工程与应用,2009, 45(11).

14 Modha DS, Spangler WS. Feature weighting in k-means clustering. Machine Learning, 2003, 52(3): 217–237.

15 He J, Lan M, Tan CL, et al. Initialization of cluster refinement algorithms: A review and comparative study. Proc. 2004 IEEE International Joint Conference on Neural Networks. IEEE. 2004. 1.