文章编号: 1001-0920(2011)01-0879-04

节点优先级导向的聚类算法

宗 瑜[1,2], 徐贯东[2], 张彦春[2], 李明楚[1]

(1. 大连理工大学 软件学院，辽宁 大连 116621；2. 澳大利亚维多利亚大学 信息应用中心，墨尔本 VIC3011)

摘 要: 基于密度的聚类算法具有挖掘任意形状聚类结果和处理“噪声”数据等优势, 同时也存在无法处理高维和密度分布不均匀数据的缺陷; 鉴于此, 给出了节点优先级导向的聚类算法. 首先建立数据集的有向 K 邻居图; 然后用 K-最近邻核密度估计方法获得数据对象的局部信息, 并在图中迭代地传播, 以产生数据对象的优先级; 最后以该优先级为导向从图中搜索聚类结果. 实验结果表明, 该算法适合处理高维、密度分布不均匀的数据.

关键词: 密度聚类； K-最近邻核密度； 节点优先级

中图分类号: TP181 **文献标识码:** A

Node priority guided clustering algorithm

ZONG Yu[1,2], *XU Guan-dong*[2], *ZHANG Yan-chun*[2], *LI Ming-chu*[1]

(1. School of Software，Dalian University of Technology，Dalian 116621，China；2. Center of Applied Information, Victoria University，Melbourne VIC3011，Australia．Correspondent：ZONG Yu，E-mail：nick.zongy@gmail.com)

Abstract: Density-based clustering algorithms have the advantages of clustering with arbitrary shapes and handling noise data, but cannot deal with unsymmetrical density distribution and high dimensionality dataset. Therefore, a node priority guided clustering algorithm(NPGC) is proposed. A direct K neighbor graph of dataset is set up based on KNN neighbor method. Then the local information of each node in graph is captured by using KNN kernel density estimate method, and the node priority is calculated by passing the local information through graph. Finally, a depth-first search on graph is applied to find out the clustering results based on the local kernel degree. Experiment results show that NPGC has the ability to deal with unsymmetrical density distribution and high dimensionality dataset.

Key words: density clustering； KNN kernel density； node priority

1 引 言

聚类是数据挖掘、模式识别等研究方向的重要研究内容之一, 在识别数据的内在结构方面具有重要的作用. 近年来, 人们提出了很多聚类算法[1-2], 密度聚类方法是其中重要的一类方法, 它在以空间信息处理为代表的众多领域有着广泛应用. 特别是伴随着处理大规模数据集、可伸缩聚类方法的开发, 其在空间数据挖掘研究领域日趋活跃. DBSCAN[3]是经典的密度聚类算法, 该算法用密度参数发现数据集中数据对象的局部分布, 然后合并密度可达的密集区域形成任意形状的聚类结果. 文献 [4] 用密度等值线图描述数据样本的分布, 给出了 GDILC 算法. [5] 提出一种新的基于移位网格概率、密度和网格的聚类算法 SGC. [6] 将密度-网格聚类算法和并行轴划分策略相结合, 给出了一种能够处理大型、高维空间数据库的聚类算法 GCHL. [7] 将对象间的空间距离概念扩展到轨迹间的时空距离概念, 从而将密度聚类算法扩展到轨迹上. ST-DBSCAN[8]是在扩展了 DBSCAN 算法的核对象、噪声对象和邻近类簇等概念的基础上提出的. FDBSCAN 算法[9]以核心对象-邻域中的某些代表对象为种子对象进行聚类探测, 以此提高聚类的探测效率. DCHT 算法[10]用层次树模型来描述聚类过程中的所有子聚类信息及整个数据集信息.

目前, 已有的密度聚类算法均以提高时间效率、强化发现任意形状聚类结果和处理“噪声”对象为基本目的, 忽视了密度分布不均匀和高维数据的聚类问题. 为了捕获数据集的真实密度分布, 本文首先在 KNN 邻居及 KNN 核密度估计技术的基础上, 定义了局部覆盖度和局部核心度, 前者反映数据对象所在区域是否为密集区域的信息, 后者反映该数据对象是否

收稿日期: 2009-10-25； **修回日期:** 2010-01-10.

基金项目: 国家自然科学基金项目(60503003)； 国家973计划项目(2007CB714205)； 安徽省教育厅重点项目(KJ2009A54).

作者简介: 宗瑜(1976−), 男, 副教授, 博士, 从事智能算法、数据挖掘等研究； 李明楚(1963−), 男, 教授, 博士生导师, 从事图论、网格计算等研究.

能够代表该密度区域; 然后, 设计了一种局部信息传递方法, 该方法将每个数据对象的局部覆盖度和局部核心度等局部信息进行传播, 以反映数据集的全局分布; 最后, 用收敛后的局部核心度来定义数据对象的优先级, 并按照优先级从大到小的顺序排序数据. 节点优先级导向的聚类算法(NPGC)完成了局部信息的产生、传播及聚类的全过程. 该方法首先建立数据集的有向K邻居图G, 利用KNN核密度估计方法产生每个数据对象的局部信息; 然后在图G中传播这些局部信息, 获得数据对象的优先级; 最后根据数据对象的优先级在有向邻居图G中搜索聚类结果. 实验结果表明, 该方法适合处理高维、密度分布不均匀的数据.

2 相关定义

定义1 给定数据集$D=\{x_1,x_2,\cdots,x_n\}, x_i\in \boldsymbol{R}^d$及所有数据对象的$K$个最近邻居, 则有向$K$邻居图定义为$G=(V,E)$. 其中: V是D中所有数据对象的集合; E是有向边$e\langle p,q\rangle$集合, $e\langle p,q\rangle$表示如果$q\in N(p)$, 则$p\to q$, $N(p)$表示节点p的K个邻居集合.

定义2 给定K邻居图G, 其邻居矩阵定义为A. 其中: $A(p,q)=1$表示存在有向边$e\langle p,q\rangle$; $A(p,q)=0$表示不存在有向边$e\langle p,q\rangle$.

图G中不被任何节点指向的节点看作“噪声”点, 在实际运算过程中将被删除.

定义3 给定d维空间中的N个数据对象, $K_d(x)$为d维概率密度函数, $H_x=[h_1^x,h_2^x,\cdots,h_d^x]$是$V_x$在$d$维空间的窗宽向量, 则称

$$\hat{f}_n(x)=\frac{1}{NV_x}\sum_{i=1}^{n}K_d((x-x_i)/H_x) \tag{1}$$

为$f(x)$的一个多维KNN核密度估计, 其中

$$V_x=\prod_{i=1}^{d}h_i^x.$$

KNN核密度估计能够捕捉数据集的局部特征和密度分布情况, 很好地反映数据集的真实数据分布, 同时KNN核密度估计具有估计结果光滑和自适应核宽度的特点. 本文首先在每个节点的K邻域内, 利用KNN核密度估计方法计算其KNN核密度值; 然后再用KNN核密度值来计算每个节点的局部覆盖度和局部核心度.

定义4 给定节点p及其K邻域$N(p)$, 节点p的局部覆盖度lc(local cover)由其K个邻居的核密度定义, 即$\mathrm{lc}(p)=\sum_{q\in N(p)}f(q)$.

$\mathrm{lc}(p)$是被其指向的邻居的核密度之和, 其值越高, 说明p的邻居中高核密度的数据对象越多, 其在高密度区域的概率越大.

定义5 给定节点p及K邻域中包含p的节点集合Q, 节点p的局部核心度lk(local kernel)由Q中所有节点的核密度定义, 即$\mathrm{lk}(p)=\sum_{q\in Q}f(q)$.

$\mathrm{lk}(p)$是所有指向p的节点的核密度之和, 其值越高, 说明同时指向p点的高密度数据对象个数越多, 其作为局部核心的可靠性越大.

利用每个数据对象的KNN核密度捕捉每个数据对象周围的数据分布情况, 再用这些局部信息刻画每个数据对象, 作为密度中心的程度及其可能的覆盖. 局部覆盖度和局部核心度刻画的是数据局部分布, 而聚类结果反映的是数据集的全局密度分布. 因此, 需要将这些局部分布传播开, 以捕获数据集的全局分布, 从而产生最终的聚类结果.

定义6 假设LC_i表示i次迭代运算后的所有节点的lc值向量, LK_i表示i次迭代运算后的所有节点的lk值向量. 操作I和O定义为

$$I:\mathrm{LC}_i=A^{\mathrm{T}}\times\mathrm{LK}_{i-1},\ O:\mathrm{LK}_i=A\times\mathrm{LC}_i.$$

其中: 操作I传播指向节点p的所有节点向p的lk值传递, 操作O则表示节点p向所有被其指向的节点的lc值传递.

定义6给出了一种迭代的局部信息传播方法. 该方法通过邻接矩阵与局部覆盖度、局部核心度的迭代相乘传播局部信息. 当信息传递收敛时, 任意节点的优先级由LK_i决定, 对LK_i按降序排列可得到数据集D中所有数据对象的节点优先级序列.

3 NPGC算法

3.1 NPGC算法框架

NPGC框架如下:

Algorithm: NPGC.

Input: D, K;

Output: C.

Step 1: 产生$x_i\in D$的K个最近邻居, 并计算其对应的KNN核密度;

Step 2: 根据K最近邻居创建K最近邻居有向图$G=(V,E)$;

Step 3: 计算图G中每个顶点的初始lc和lk;

Step 4: 迭代计算LC_i和LK_i直至收敛;

Step 5: 按降序排列LK_i;

Step 6: 若G中还有顶点没被处理, 则

$$v=\max(\mathrm{LK}_i),\ C(v)=\mathrm{DFS}(G,v);$$

Step 7: Return C.

在上述算法中, Step 1发现数据集中每个数据对象的K个最近邻居, 并计算在该邻域中数据对象的KNN核密度值. 因为核密度能够反映数据集的真实分布情况, 所以本文用核密度表示数据对象的局部信

息, 并借助该局部信息的传播达到发现数据集 D 中最有可能成为中心点的数据对象. 为了实现局部信息的传播, 创建该数据集的 KNN 邻居图, 并在该邻居图中实现传播. Step 2 完成了创建 K 最近邻居图的功能. Step 3 用定义 4 和定义 5 完成局部信息初始化, 局部信息必须沿着有向图 G 中的有向边来传播. 定义 6 给出了信息传播的两个步骤 I 和 O, 当局部信息传递收敛时, 对其中的 LK_i 值进行降序排列, 然后选择 LK_i 中 lk 值最大的数据对象作为当前深度优先搜索的起始点, 在图 G 中搜索其所有可到达的且 lk 值小于当前搜索点的数据对象. 如此继续, 直至图 G 中所有顶点均被处理为止.

3.2　节点优先级与 K 的关系

图 G 中每个节点的优先级与数据对象的 KNN 核密度估计有关, 而核密度估计是在该节点的 K 最近邻域内产生的, 因此节点的邻居数 K 与节点优先级之间存在着一定的联系. 为了考察这层关系, 构造 1 个包含 30 个 2 维数据对象的不均匀数据集, 该数据集包含 2 个聚类结果. 在数据对象的 KNN 核密度及其 K 邻域的基础上, 按照定义 4 和定义 5 产生每个数据对象的初始局部信息 LC_0 和 LK_0, 然后根据定义 6 在有向图 G 中传播该局部信息. 当传播收敛后, 以 LK_i 表示数据对象的优先级. 图 1 给出了在不同 K 值的情况下, 每个数据对象的优先级排列情况.

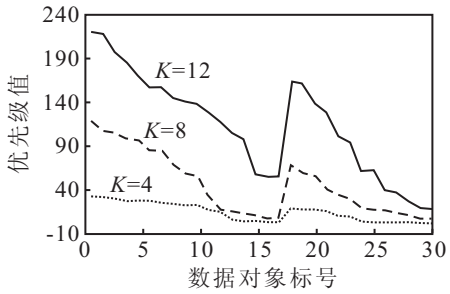


图 1　不同 K 值情况下的节点优先级

为了在图 1 中更清晰地表示优先级, 用原优先级值×100 作为纵坐标标度. 从图 1 中可以观察到, 节点优先级不会因为 K 值的变化而改变. K 值的增大只是更明确了不同聚类之间的优先级差别. 因此, 可以认为数据对象的节点优先级对于参数 K 不敏感.

3.3　时间复杂度分析

一般而言, 计算数据集 D 中所有数据对象的 K 最近邻居, 其时间消耗为 $O(N^2)$, 但是采用 K-d 树的最近邻算法仅需要 $O(N\log N)$ 的时间消耗. 计算每个数据对象的 KNN 核密度需要考虑数据对象邻域中的数据对象, 其时间消耗为 $O(K)$, 因此计算 N 个数据对象共需要 $O(NK)$. 创建 K 最近邻居图与计算 K 最近邻居可以同时进行, 其时间消耗为 $O(KN\log N)$. 第 3.1 节 Step 4 迭代计算 LC_i 和 LK_i 值直至收敛, 所需的时间与迭代次数有关, 本文记其时间消耗为 $O(\cdot)$. Step 5 降序排列 LK_i 时间消耗为 $O(N\log N)$, 而 Step 6 以图 G 的剩余节点中 LK_i 值最大的节点为起点, 执行 DFS 搜索所需时间最多, 为 $O(N^2)$. 因此, 算法 NPGC 的总时间消耗为 $O(N^2)+(K+2)O(N\log N)+O(NK)+O(\cdot)$, 其中 $O(\cdot)$ 为 $O(N^2)$ 量级.

4　实验结果及其分析

本文首先用文献 [3] 提供的实验数据集考察算法 NPGC 发现密度不均匀聚类的能力, 然后考察算法 NPGC 对于高维数据及时间的可扩展性. 实验算法均由 Matlab 7.0 编程语言实现, 并在 Intel 2.0/1 GM/80 G 兼容机和 windows XP 环境下执行.

4.1　NPGC 发现密度不均匀聚类的能力

为了考察 NPGC 算法发现密度不均匀聚类结果和处理噪声的能力, 本文用 DBSCAN 算法采用的公共数据集对其进行实验. DBSCAN 数据集包含 4 个不同形状、不同大小、不同密度的聚类, 且布满了噪声数据. 图 2 给出了 NPGC 算法在 DBSCAN 数据集的实验结果. 从图 2 中可以看出, NPGC 算法能够准确发现正确的 DBSCAN 数据集中不同密度分布的聚类结果, 且去除了大量噪声数据对象.

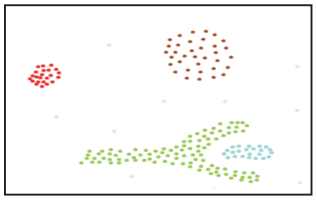


图 2　DBSCAN 数据集上的实验结果

4.2　NPGC 对于高维数据的可扩展性

NPGC 算法的本质是首先找到最有可能成为聚类中心的数据对象, 然后以此为起点对有向图 G 进行搜索. 该算法通过 KNN 核密度估计和有向 K 邻居图来产生数据对象的优先级, 因此, 算法 NPGC 既有密度算法的思想, 又有 CHAMELEON 算法[11]最近 K 邻居图的概念. 基于此, 本文在一组实际数据集上对比了 NPGC, DBSCAN 和 CHAMELEON 算法的聚类质量, 验证了算法对于高维数据的可扩展性. 聚类质量的评价指标采用文献 [12] 中的 micro$-p$ $=\sum a_t/N$, 其中 a_t 是第 t 个聚类中包含了第 t 个分类中数据对象的个数. micro $-p$ 的值越大, 说明聚类质量越高. 实验采用的数据集的基本描述见表 1.

表 1 所示的 5 个数据集的属性取值均是实数或整数, 其中最后 1 个属性为类标号, 在实验中该属性不被考虑, 仅在衡量算法质量时被使用. 在实验之前, 实验数据进行如下的正则化处理:

表 1 实际数据集的基本描述

data set	numbers	attributes	class
pima indians diabetes database(PIDD)	768	9	2
blocks classification(BC)	5 473	11	5
pen-based recognition of handwritten digits(PRHD)	7 494	17	10
connectionist bench dataset(CBD)	208	61	6
quadruped mammalsQM	5 000	73	4

$$x_{ij} = (x_{ij} - \min\{x_{lj}\})/(\max\{x_{lj}\} - \min\{x_{lj}\}),$$
$$l = 1, 2, \cdots, d. \tag{2}$$

图 3 给出了 3 种算法在 5 个数据集上的实验结果. 由图 3 可见, 在维数不太高的数据集 (PIDD, BC 和 PRHD) 上, 3 种算法的 micro$-p$ 值相差不大. 但是在维数较大的数据集中, DBSCAN 和 CHAMELEON算法的 micro$-p$ 值急剧下降, 该现象反映了维数对于算法的影响. 但是, NPGC 算法随着维数的增加, micro$-p$ 的值并没有受到太大的影响. 这是因为本文用数据对象的 KNN 核密度捕捉了数据的真实分布, 并将这种分布信息通过有向图 G 进行传播, 从而发现最有可能成为簇中心的数据对象, 再搜索聚类结果. NPGC 算法的基础是数据对象 KNN 核密度信息, 而 KNN 核密度对维数是可扩展的[13]. 因此, 算法 NPGC 产生的聚类结果质量较高.

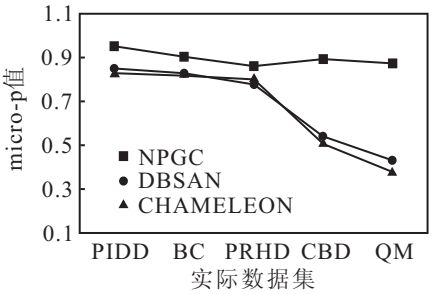
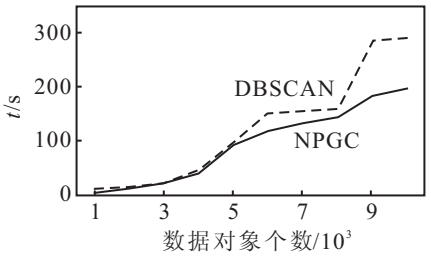


图 3 3种算法在数据集上的实验结果

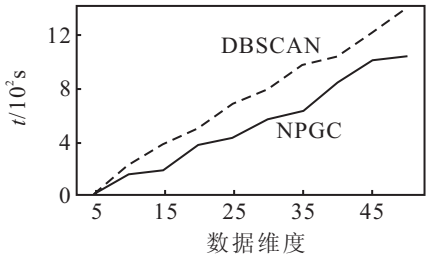
4.3 NPGC 与 DBSCAN 的时间对比

剧增的数据规模要求聚类算法具备处理大规模、高维数据的能力, 本文使用多组仿真数据集进行实验. 这些仿真数据集均由文献 [14] 给出的随机数据生成器产生. 为了测试算法消耗时间与数据集规模之间的关系, 生成维数为 19, 数据对象个数从 1 000 递增到 10 000 的 10 个数据集 (步长为 1 000). 同样, 为了测试算法时间消耗与数据集维度之间的关系, 生成数据个数为 10 000, 维度从 5 递增到 50 的 10 个数据集 (步长为 5). 实验结果如图 4 所示.

从图 4 可以看出, NPGC 算法的时间消耗趋势与 DBSCAN 相似, 造成这种相似的原因是 NPGC 算法也是 $O(N^2)$ 量级的算法. 采用 K-d 树结构构造数



(a) 算法对于数据规模的可扩展性



(b) 算法对于维度的可扩展性

图 4 NPGC 与 DBSCAN 在仿真数据集上的时间消耗对比

据集 D 的 KNN 邻居图后, NPGC 算法的主要工作都是在该邻居图上完成的, 因此, 算法在一定程度上依赖于邻居图的存储结构. 本文的 KNN 邻居图是用邻接矩阵存储的, 所以在该存储结构上发现最大连通子图 (即聚类结果) 的时间复杂度为 $O(N^2)$. 由于每个节点都被分配了一个优先级, 深度有限算法的搜索空间被减小, 从而降低了 NPGC 算法的总时间.

5 结 论

密度聚类算法能够挖掘任意形状聚类的聚类结果, 但是无法处理高维和密度分布不均匀的数据集. 本文给出一种节点优先级导向的聚类算法, 以数据对象的 KNN 核密度捕获数据分布的局部信息; 并通过在有向图中传播局部信息来获得数据对象的优先级; 最后利用节点优先级从有向图中搜索聚类结果. 实验结果表明, NPGC 算法具备发现密度不均匀聚类结果的能力, 且对于高维数据具有较好的可扩展性.

参考文献(References)

[1] Rui X, Wunsch D. Survey of clustering algorithms[J]. IEEE Trans on Neural Network, 2005, 16(3): 645-678.

[2] 孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J]. 软件学报, 2008, 19(1): 48-61.
(Sui J G, Liu J, Zhao L Y. Research on clustering[J]. J of Software, 2008, 19(1): 48-61.)

[3] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]. Proc of the 2nd Int Conf on Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996: 291-316.

(下转第887页)