软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(12):3115−3128 [doi: 10.13328/j.cnki.jos.005237]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于距离不等式的 *K*-medoids 聚类算法*

余冬华[1], 郭茂祖[1,2], 刘 扬[1], 任世军[1], 刘晓燕[1], 刘国军[1]

[1](哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

[2](北京建筑大学 电气与信息工程学院,北京 100044)

通讯作者: 刘扬, E-mail: yliu76@hit.edu.cn

摘 要: 研究加速 *K*-medoids 聚类算法,首先以 PAM(partitioning around medoids)、TPAM(triangular inequality elimination criteria PAM)算法为基础给出两个加速引理,并基于中心点之间距离不等式提出两个新加速定理.同时,以 $O(n+K^2)$额外内存空间开销辅助引理、定理的结合而提出加速 SPAM(speed up PAM)聚类算法,使得 *K*-medoids 聚类算法复杂度由 $O(K(n-K)^2)$降低至 $O((n-K)^2)$.在实际及人工模拟数据集上的实验结果表明:相对于 PAM,TPAM, FKMEDOIDS(fast *K*-medoids)等参考算法均有改进,运行时间比 PAM 至少提升 0.828 倍.

关键词: 数据挖掘;聚类算法;*K*-medoids;距离不等式

中图法分类号: TP181

中文引用格式: 余冬华,郭茂祖,刘扬,任世军,刘晓燕,刘国军.基于距离不等式的 *K*-medoids 聚类算法.软件学报,2017,28(12): 3115−3128. http://www.jos.org.cn/1000-9825/5237.htm

英文引用格式: Yu DH, Guo MZ, Liu Y, Ren SJ, Liu XY, Liu GJ. *K*-Medoids clustering algorithm based on distance inequality. Ruan Jian Xue Bao/Journal of Software, 2017,28(12):3115−3128 (in Chinese). http://www.jos.org.cn/1000-9825/5237. htm

K-Medoids Clustering Algorithm Based on Distance Inequality

YU Dong-Hua[1], GUO Mao-Zu[1,2], LIU Yang[1], REN Shi-Jun[1], LIU Xiao-Yan[1], LIU Guo-Jun[1]

[1](School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[2](School of Electrical Engineering and Information Technique, Beijing University of Civil Engineering and Architecture, Beijing 100044, China)

Abstract: This paper presents a research on speeding up *K*-medoids clustering algorithm. Firstly, two acceleration lemmas are given based on partitioning around medoids (PAM) and triangular inequality elimination criteria PAM (TPAM) algorithms. Then two new acceleration theorems are proposed based on distance inequality between center points. Combining the lemmas with the theorems with the aid of additional memory space $O(n+K^2)$, the speed up partitioning around medoids (SPAM) algorithm is constructed, reducing the time complexity from $O(K(n-K)^2)$ to $O((n-K)^2)$. Experimental results on both real-world and artificial datasets show that the SPAM algorithm outperforms PAM, TPAM and FKEMDOIDS approaches by at least 0.828 times over PAM in terms of running time.

Key words: data mining; clustering algorithm; *K*-medoids; distance inequality

聚类分析是数据挖掘、模式识别等研究方向的重要研究内容之一,主要是将数据集中相似的样本尽可能划分为相同的簇,而把相异的样本尽可能划归为不同的簇.经过几十年的发展,已经形成众多经典聚类方法[1,2].在广受欢迎的新聚类算法方面,Frey 和 Dueck 在 2007 年基于因子图的信念传播和最大化算法提出近邻传播聚类(affinity propagation,简称 AP)[3,4],Rodriguez 和 Laio 在 2014 年基于聚类中心的刻画提出快速搜索密度峰值聚类

* 基金项目: 国家自然科学基金(61571164, 61571163, 61671188, 61671189, QC2014C071)
Foundation item: National Natural Science Foundation of China (61571164, 61571163, 61671188, 61671189, QC2014C071)
收稿时间: 2016-06-18; 修改时间: 2016-09-07, 2016-10-26; 采用时间: 2016-12-08; jos 在线出版时间: 2017-03-24
CNKI 网络优先出版: 2017-03-24 15:31:41, http://kns.cnki.net/kcms/detail/11.2560.TP.20170324.1531.006.html

算法(clustering by fast search and find of density peaks)[5],为聚类算法的设计提供了一种全新思路.

K-means 与 *K*-medoids 是基于划分的经典聚类算法.*K*-means 以其低计算复杂度受到广泛欢迎,*K*-medoids 因强鲁棒性、对噪声数据及异常点处理能力强等优势,同样被广泛应用[6,7].一直以来,很多学者都致力于改进 *K*-medoids 聚类算法,以期在效率上追赶 *K*-means 算法,同时又维持其优点.近年来,对 *K*-medoids 聚类效率的研究可以概括为如下 4 个方面.

1) 以抽样为基础,减少聚类样本数及(或)中心点交换次数.

Kaufman 提出的 CLARA(clustering large applications)[8]算法针对整个数据集进行多重抽样,以抽样子集的最优中心点充当整个数据集的中心点进行 PAM(partitioning around medoids)聚类.相对于聚类子集抽样,Ng 等人提出的 CLARANS(CLARA based on randomized search)[9]算法是对中心点进行抽样,随机地选择一个代表对象(中心点)、一个非代表对象(非中心点)计算交换代价,当随机选择次数达到最大阈值后停止搜索,返回当前最优结果.Barioni 提出的 PAM-SLIM(slim-tree applied to PAM)[10]算法首先把数据集建成 Slim-tree 结构,然后以 Slim-tree 中的一层作为抽样集进行 PAM 聚类,得到的最优中心点作为整个数据集的最优中心点.

2) 以优化中心点为基础,减少聚类过程中的迭代次数或者中心点交换次数.

Park 等人[11]提出的快速 *K*-medoids 算法(fast *K*-medoids,简称 FKMEDOIDS)以密度排序选择初始中心点,并事先计算出所有样本的距离矩阵,在聚类过程中,凡涉及样本点距离计算时,只需要调用该距离矩阵的值,然而,保存距离矩阵需要占据大量空间内存,是一种典型的以空间换取时间的算法.Zadegan 等人[12]提出的排序 *K*-medoids 算法与 FKMEDOIDS 都事先计算相异度矩阵,不同点在于,基于排序更新中心点可以减少中心点交换次数,获得聚类效率的提升.预先计算距离矩阵的思想也被二分 *K*-medoids(bisecting *K*-medoids,简称 BPAM)[13]聚类所采用,并且在基因共表达数据上验证了该方法的实用性.Lai 等人[14]提出了方差增强 *K*-medoids 聚类,该方法的中心点是逐个增加,这与 PAM 直接给定类的个数不同;其次,该方法还引入了 Polygon 算子,沿类中心点分割线计算方差,以此选择合适的中心点.

3) 以并行计算为基础,实现并行 *K*-medoids 聚类.

Jiang 等人[15]实现了基于 Hapdoop 分布式计算平台上的 *K*-medoids 聚类.Yue 等人[16]基于迭代 MapReduce 过程提出并行化 *K*-medoids 算法.Han 提出了基于具备分布式、极大并行性和非确定等特点的 *P* 系统优化 *K*-medoids 算法[17,18].

4) 以避免重复计算为基础,减少诸如点之间距离等的重复计算.

CLATIN[19]算法利用中心点之间的三角形不规则网络(triangular irregular network,简称 TIN)确定交换中心点后的受影响子集,并只计算该子集的交换代价,以此决定是否交换中心点,有效提高了聚类效率.2002 年,Chu 等人[20,21]提出了部分距离(partial distance)、前一次中心点指标(previous medoid index)、三角不等式消除条件(triangular inequality elimination criteria)优化 PAM 的方法,这些方法可以有效减少重复的距离计算.2007 年,Chu 等人[22]在原有基础上推导出一些新的不等式,包括涉及 3 个中心点之间的距离关系、新增样本点与坐标原点的距离关系并保存在内存中以供直接调用.2008 年,Chu 等人[23]再次对上述优化方法进行总结,舍弃一些优化效果不佳的不等式,比如含 3 个中心点情形等,并探讨了这些优化方法的组合情形及相应的实验结果.

纵观上述基于抽样的优化算法,基于中心点优化及并行化加速 *K*-medoids 聚类的算法均未针对 PAM 算法本身进行改进,因此,针对 PAM 优化的算法可以直接应用到上述 3 种优化方法中.而避免重复计算的方法直接优化 PAM 算法,此优化最大的优势在于:可以保持 PAM 算法所有的优秀性质,并且可以直接应用在基于抽样、中心点优化、并行化等 *K*-medoids 聚类中,使得聚类效率可以叠加.基于以上综述及文献[8,20–23]中的加速思想,本文将以距离不等式为基础提出 SPAM(speed up PAM)优化算法,主要包括以下 3 个方面.

1. 将文献[8]与文献[20–23]中能够加速 *K*-medoids 聚类且具有普适性的距离不等式分别整理成引理 3.1、引理 3.2,应用到 SPAM 算法中;
2. 提出新的距离不等式定理 3.1 及定理 3.2 加速 *K*-medoids 聚类过程,并证明其正确性;
3. 提出以 $O(n+K^2)$额外内存空间开销辅助引理 3.1、引理 3.2、定理 3.1 的结合,并应用到 SPAM 算法中,

使得 K-medoids 聚类复杂度由 $O(K(n-K)^2)$降低至 $O((n-K)^2)$.

本文所涉及的 4 种方法——PAM,TPAM,FKEMEOIDS,SPAM 及引理、定理之间具有紧密联系,如:SPAM 算法是基于 TPAM 算法、定理 3.1 及空间加速,而 TPAM 算法又是基于 PAM 算法及引理 3.2,FKMEDOIDS 与 TPAM 均是基于 PAM 算法的改进.图 1 可以清晰简洁地反映它们之间这种层次关系,同时也能直观反映方法之间的差异,如 FKMEDOIDS 与 TPAM 方法都是基于 PAM 方法.然而 FKMEDOIDS 方法利用了距离矩阵,同时对中心点进行了优化,这些并没有应用在 TPAM 方法中.

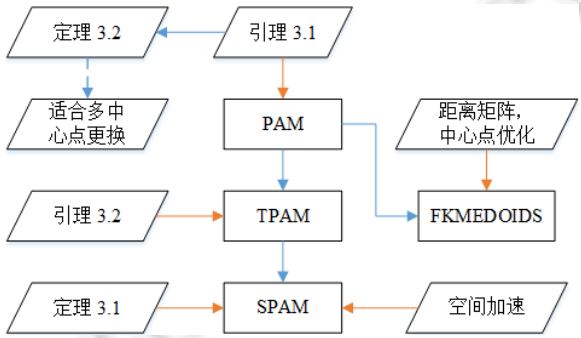


Fig.1 Hierarchical relationship of four methods and lemma, theorem

图 1 4 种方法及引理、定理的层次关系

1 K-medoids 算法简介

设数据集 D 包含 n 个 d 维欧式空间中的对象$\{x_1,x_2,\ldots,x_n\}$,$x_i\in R^d$,K-medoids 聚类就是把 D 中的对象分配到 K 个中心点 $O=\{O_1,O_2,\ldots,O_K\}$所代表的簇 $C_1,C_2,\ldots,C_K$ 中,使得对于 $1\leqslant i,j\leqslant K$,$C_i\subset D$ 且 $C_i\cap C_j=\Phi$,$\cup_{i=1}^{K}=D$,并且使得簇内对象相互相似,而与其他簇中的对象相异.围绕中心点划分 PAM(partitioning around medoids)算法是 K-medoids 聚类的一种主流实现,PAM 算法流程如下[24]:

输入:数据集 D,簇个数 K;

输出:K 个簇的集合.

1) 从 D 中随机选择 K 个对象作为初始的代表对象;

2) Repeat

3) 将每个剩余的对象分配到最近的代表对象所代表的簇;

4) 随机地选择一个非代表对象 O_{random};

5) 计算用 O_{random} 代替代表对象 O_j 的总代价 S;

6) If S<0,then O_{random} 替换 O_j,形成新的 K 个代表对象的集合;

7) Until S 不发生变化

本文以上述 PAM 算法为基础进行改进研究.

2 K-medoids 加速理论

2.1 K-medoids距离加速理论

在 PAM 算法步骤 3)中,需要反复计算非代表对象与每一个中心点之间的距离,并通过比较,将非代表对象重新分配到合适的簇中,每更换一个中心点,都需要重复上述步骤并计算交换总代价.文献[8]将初始聚类(聚类过程中,初次对非代表对象进行划分)与后续聚类(聚类过程中,除初次划分之外的所有划分)分开处理,在后续聚类过程中,一般分为 4 种情形对所有非代表对象进行重新分派,并计算交换总代价,我们将上述 4 种情形整理成如下引理.

引理 3.1(加速引理 I). 设 $O_i,i=1,2,\ldots,K$ 代表第 i 个簇 C_i 的中心点,$dist(P,E)$代表样本点 $P,E\in D$ 之间的距离.在 K-medoids 聚类中,中心点组 $O=\{O_1,O_2,\ldots,O_K\},j=1,2,\ldots,K$ 由非代表对象 O'_j 交换代表对象 O_j 成为簇 C_j 的中心点,即,交换后新中心点组为 $O'=\{O_1,\ldots,O'_j,\ldots,O_K\}$,则对于任意非代表对象 $P\in D$,有:

(1) 当 $P\in C_i,i\neq j$,若 $dist(P,O_i)<dist(P,O'_j)$,则 $P\in C_i$;否则,$P\in C_j$;

(2) 当 $P\in C_j$,若 $dist(P,O_j)>dist(P,O'_j)$,则 $P\in C_j$;否则,$P\in C_i$.其中,i 满足 $dist(P,Q_i)=\min_l\{dist(P,Q_l)|Q_l\in Q'\}$.

引理 3.1 说明:非代表对象新簇标号的更新并不需要完全重新计算该非代表对象与新中心点组中所有中心点之间的距离,这就可以节省无效的距离计算.这种基于距离不等式的方法被应用于 Chu 等人[20−23]提出的 TPAM 算法中(注:原文中将不等式推广在随机抽样的 CLARANS 算法基础上,为了在本文中参照,将不等式推广在 PAM 算法上,将其称为 TPAM 算法),并提出一个新的距离不等式.尽管原文并未将其以严格的定理形式给出,但是对其进行了严谨的数学阐述,现将其整理为如下加速引理:

引理 3.2(加速引理 II). 设 $O_i,i=1,2,\ldots,K$ 代表第 i 个簇 C_i 的中心点,$dist(P,E)$代表样本点 P,E 之间的距离.在 K-medoids 聚类中,中心点组 $O=\{O_1,\ldots,O_j,\ldots,O_K\},j=1,2,\ldots,K$ 由非代表对象 O'_j 交换代表对象 O_j 成为簇 C_j 的中心点,即,交换后新中心点组为 $O'=\{O_1,\ldots,O'_j,\ldots,O_K\}$.则对于任意非代表对象 $P\in C_i$,若 $dist(O_i,O'_j)\geqslant 2dist(P,O_i)$,则 $P\in C_i$.

引理 3.2 考虑了前一次聚类时的中心点 O_i 及新交换后的中心点 O'_j 之间的关系,O'_j 与簇 C_j 有着紧密联系,我们发现,簇 C_j 的前一次聚类中心点 O_j 与当前新中心点 O'_j 也蕴含着与引理 3.2 类似的关系,我们将其称为定理 3.1.

定理 3.1(加速定理 I). 设 $O_i,i=1,2,\ldots,K$ 代表第 i 个簇 C_i 的中心点,$dist(P,E)$代表样本点 P,E 之间的距离.在 K-medoids 聚类中,中心点组 $O=\{O_1,\ldots,O_j,\ldots,O_K\},j=1,2,\ldots,K$ 由非代表对象 $O'_j,O'_j\notin C_i$ 交换代表对象 O_j 成为簇 C_j 的新中心点,即,交换后新中心点组为 $O'=\{O_1,\ldots,O'_j,\ldots,O_K\}$.则对于任意非代表对象 $P\in C_i,i\neq j$,若 $dist(P,Q_i)\leqslant dist(O_i,O_j)/2+|\cos\theta|\,dist(O_j,O'_j)$,则 $P\in C_i$.其中,$\cos\theta=\dfrac{dist^2(O_i,O_j)+dist^2(O_j,O'_j)-dist^2(O_i,O'_j)}{2dist(O_i,O_j)dist(O_j,O'_j)}$.

证明:分两种情况进行证明.

- 情形 1,P 恰好在平面内 $O_iO_jO'_j$,此时令 $P_0=P$.

令 $O'_j\notin C_j$,由于 C_i,C_j 不是同一个簇,则 C_i,C_j 必定可以由 O_iO_j 的中垂线分开(在 R^d 空间中,任意不在同一直线上的 3 个点必可以在同一平面内,以下证明均在此平面 $O_iO_jO'_j$ 内),如图 2 所示.

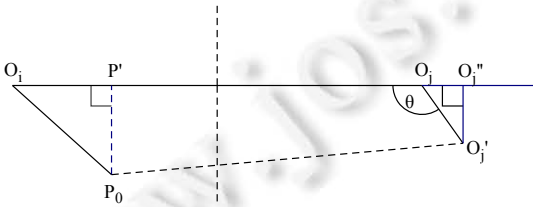


Fig.2 Sketch map of speeding up theorem I

图 2 加速定理 I 证明图示

令 P'是 P_0 在直线 O_iO_j 上的投影,O''_j 是 O'_j 在直线 O_iO_j 上的投影,易知:

$$dist(P',O''_j)\leqslant dist(P_0,O'_j).$$

由于 $P_0\in C_i$,因此 P'点不可能超过 O_iO_j 的中垂线(图 2 垂直虚线),故:

$$dist(O_i,O_j)/2-\cos\theta\cdot dist(O_j,O'_j)\leqslant dist(P',O''_j).$$

根据余弦定理可知:

$$\cos\theta=\frac{dist^2(O_i,O_j)+dist^2(O_j,O'_j)-dist^2(O_i,O'_j)}{2dist(O_i,O_j)dist(O_j,O'_j)}.$$

根据已知:

$$dist(P_0,O_i)=dist(P,O_i)\leqslant dist(O_i,O_j)/2+|\cos\theta|\,dist(O_j,O'_j)$$

得到:

$$dist(P,O_i)\leqslant dist(P',O''_j)\leqslant dist(P_0,O'_j)=dist(P,O'_j).$$

因此,$P\in C_i$.注意:由图中所示,去掉绝对值符号需要添加一个负号.

- 情形 2,此时 P 不在平面 $O_iO_jO'_j$ 内,令 P_0 为 P 在该平面内的投影.则:

$$dist(P_0,Q_i)\leqslant dist(P,Q_i).$$

由已知条件:

$$dist(P,O_i)\leqslant dist(O_i,O_j)/2+|\cos\theta|\,dist(O_j,O'_j)$$

可知:

$$dist(P_0,O_i)\leqslant dist(O_i,O_j)/2+|\cos\theta|\,dist(O_j,O'_j).$$

此时与情形 1 一致,有:

$$dist(P_0,O_i)\leqslant dist(P_0,O'_j).$$

又因为 P_0 是 P 在平面 $O_iO_jO'_j$ 内的投影,故:

$$dist(P,O_i)\leqslant dist(P,O'_j).$$

因此,$P\in C_i$.

综上两种情形,定理得证. □

定理 3.1 与引理 3.2 都是以中心点之间的距离为基础,但是选择了不同的距离.

$dist(O_i,O'_j)/2$ 与 $dist(O_i,O_j)/2+|\cos\theta|\,dist(O_j,O'_j)$ 的值有些相近,特殊情形时也可以相等.

- 当 $\theta<\pi/2$ 时,绝对值前需要改成负号,这样,定理 3.1 要劣于引理 3.2;
- 但是 $\theta>\pi/2$ 时,虽然 $dist(O_i,O'_j)/2$ 是三角形 $O_iO_jO'_j$ 的最长边的一半,$dist(O_i,Q_j)/2$ 是较短边的一半,然而定理 3.1 中增加了 $|\cos\theta|\,dist(O_j,O'_j)$,使得 $dist(O_i,O_j)/2+|\cos\theta|\,dist(O_j,O'_j)$ 可能优于 $dist(O_i,O'_j)/2$.

为了更严格估计 θ 的合理范围,假设定理 3.1 优于引理 3.2,此时,不等式(1)必须成立:

$$dist(O_i,O_j)/2+|\cos\theta|\,dist(O_j,O'_j)\geqslant dist(O_i,O'_j)/2 \quad (1)$$

注意:限定 $\theta>\pi/2$,由于式(1)两边均大于 0,故可以先乘以 2,然后直接平方,且不改变不等式符号,可得:

$$dist^2(O_i,O_j)+4|\cos\theta|\,dist(O_j,O'_j)dist(O_i,O_j)+4|\cos\theta|^2\,dist^2(O_j,O'_j)\geqslant dist^2(O_i,O'_j) \quad (2)$$

将 $\cos\theta$ 用余弦定理代入式(2),需要注意:由于 $\theta>\pi/2$,加上绝对值符号后应添加一个负号,并进行整理得:

$$2|\cos\theta|\,dist(O_i,O_j)\geqslant(1-4|\cos\theta|^2)dist(O_j,O'_j).$$

上式同除以 $2|\cos\theta|$, $dist(O_j,O'_j)>0$ 的数,不等式符号不变,则:

$$dist(O_i,O_j)/dist(O_j,O'_j)\geqslant(1-4|\cos\theta|^2)/(2|\cos\theta|) \quad (3)$$

即:只要 θ 满足式(3),就能保证式(1)成立.此时,定理 3.1 优于引理 3.2.图 3 给出了 θ 与 $(1-4|\cos\theta|^2)/(2|\cos\theta|)$ 在区间 $[2\pi/3,\pi]$ 上的图像.

当 $\theta=2\pi/3$ 时,$(1-4|\cos\theta|^2)/(2|\cos\theta|)=8.8818e-16\sim0$,结合图 3 可知:$\theta>2\pi/3$ 时,式(3)总是成立的,亦即定理 3.1 要优于引理 3.2.因此,建议 $\theta>2\pi/3$ 时,应用定理 3.1;反之,则用引理 3.2.

引理 3.1、引理 3.2 与定理 3.1 都是针对交换单个中心点的算法,如 PAM,TPAM 等,但是诸如 FKMEDOIDS 聚类算法,可能同时交换多个中心点,此时,上述引理与定理将不再适用.为此,我们将引理 3.1 推广至同时交换多中心点情形,有如下定理.

定理 3.2(加速定理 II). 设 $O_i,i=1,2,\ldots,K$ 代表第 i 个簇 C_i 的中心点,$dist(P,E)$ 代表样本点 P,E 之间的距离.在 K-medoids 聚类中,中心点组 $O=\{O_1,O_2,\ldots,O_K\}$ 中任意 $m(1\leqslant m\leqslant K)$ 个代表对象被非代表对象交换后成为新中心

点组 $O'=\{O'_1,O'_2,...,O'_K\}$,则对于任意非代表对象 P,有:

(1) 当 $P\in C_i,O_i\in O\cap O'$,若 $dist(P,O_i)<\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$,则 $P\in C_i$,否则重新分配 $P\in C_l$.其中,l 满足 $dist(P,O'_l)=\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$;

(2) 当 $P\in C_i,O_i\notin O'$,若 $dist(P,O_i)>\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$,则重新分配 $P\in C_l$,其中,l 满足 $dist(P,O'_l)=\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$;否则,重新分配 $P\in C_l$,其中,l 满足 $dist(P,O'_l)=\min_j\{dist(P,O'_j)\mid O'_j\in O'\}$.

定理 3.2 是引理 3.1 的推广,唯一不同点在于引理 3.1 考虑交换一个中心点,而在定理 3.2 中考虑同时交换多个中心点,此时只需要考虑中心点组的差集 $O'-O$ 就可以完成相关证明,简略的证明过程见附录.这个定理也表明:基于距离不等式思想的加速方法可以推广到多中心点交换情形,并不局限于 PAM,TPAM,SPAM 等算法.

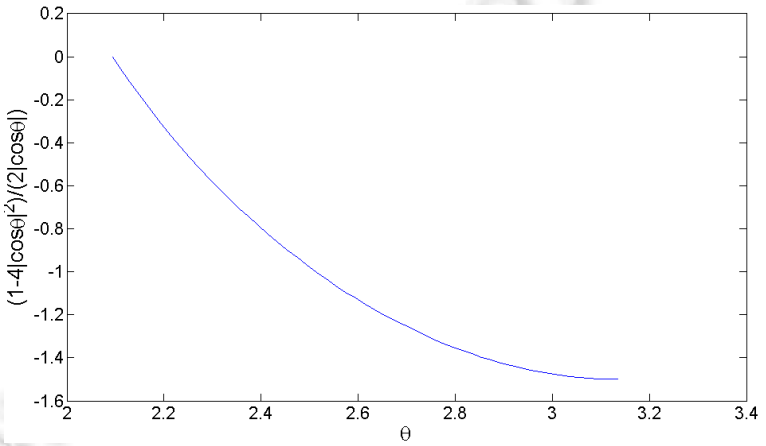


Fig.3 θ curve graph

图 3 θ曲线图

2.2 *K*-medoids空间加速

虽然加速引理 I(引理 3.1)已经证明,但想要起到加速效果,就需要先计算 $dist(P,Q_i)$, $dist(P,O'_j)$.在加速引理 II(引理 3.2)及加速定理 I(定理 3.1)中,应用它们的前提也需要事先计算 $dist(O_i,O'_j)$,$dist(P,Q_i)$,$dist(O_i,Q_j)$, $dist(O_j,O'_j)$.对于每一个非代表对象 P 来说,我们可以注意到:$dist(P,Q_i)$, $dist(P,O'_j)$ 是需要重新计算的;而 $dist(O_i,O'_j)$,$dist(O_i,O_j)$,$dist(O_j,O'_j)$ 与 P 本身无关,只跟其所属簇的中心点相关.对于每一次中心点交换之后,如果先计算中心点之间的 $K^2/2$ 个距离,那么在应用引理 3.2、定理 3.1 时,只需要再计算一次 $dist(P,Q_i)$,而引理 3.1 还需要多计算 $dist(P,O'_j)$.而 $dist(P,Q_i)$代表上一次聚类中,非代表对象 P 与其所属簇中心点之间的距离,实际上在前一次聚类中已经计算.我们可以增加相应的空间开销来保存其距离值,而避免其再次计算.这也是本文空间加速的基础.

以牺牲空间开销来降低时间消耗,是一种常见的做法.Chu 等人[22,23]在提出引理 3.2 中距离不等式时,也考虑以空间开销换取时间消耗的方法,只不过,他们是在聚类之前计算所有中心点与样本点距原点的距离 $\sqrt{\sum_{i=1}^{d}o_{3i}^2}$ 及 $\sqrt{\sum_{i=1}^{d}x_i^2}$,对于一个数据集大小为 n 及簇个数为 K 的聚类来说,其需要额外空间开销 $O(n+K)$.但是对于本文的引理 3.1、引理 3.2、定理 3.1 来说,更多涉及非代表对象与其所属簇中心点之间的距离及中心点组之间的 $K^2/2$ 个距离,因此,本文需要额外增加 $O(n+K^2)$的空间开销,以使引理 3.1、引理 3.2 及定理 3.1 更充分地发挥其加速效应.而非代表对象与其所属簇中心点之间的距离在聚类中是必须计算的,只需要将其保存即可.

3 SPAM 算法流程

上文中已经引入了提高 *K*-medoids 聚类效率的引理 3.1、引理 3.2,同时提出了定理 3.1、定理 3.2,也阐述了

内存空间与引理、定理之间的结合.在 SPAM 算法中,由于只涉及单个中心点的交换,因此只使用引理 3.1、引理 3.2、定理 3.1,它们的组织顺序如图 4 所示,其中,条件判断中规定:如果满足定理(引理)条件,则为 Yes,否则为 No.

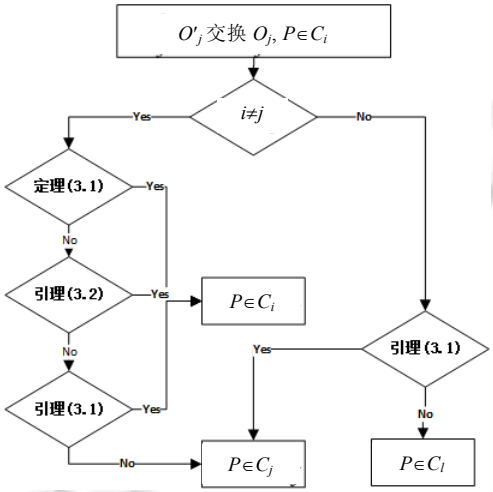


Fig.4 Organization sequence of lemma and theorem in SPAM algorithm

图 4 SPAM 算法中,引理、定理组织顺序

结合 PAM 算法流程,本文所提出的 SPAM 算法流程如下.

输入:数据集 D,簇个数 K;

输出:K 个簇的集合 D'.

1) 从 D 中随机选择 K 个样本点作为 K 个簇代表对象;
2) Repeat
3) 将每一个非代表对象分配到最近的代表对象所代表的簇;
4) 随机选择一个非代表对象 O_{random} 及代表对象 O_j 进行交换;
5) 对于每一个非代表对象 P,依次根据定理 3.1、引理 3.2、引理 3.1 判别 O_{random} 交换 O_j 后 P 所属簇并计算该交换产生的代价;
6) 计算所有样本的交换总代价 S;
7) If S<0, then O_{random} 替换 O_j,形成新的 K 个代表对象(中心点组);
8) Until 代表对象(中心点组)不发生改变
9) 输出 K 个簇的集合 D'

4 算法复杂度分析

从 PAM 与 SPAM 的算法流程上来看,这两者相差很小,都是在每一次聚类迭代(中心点组更新)中有 $K(n-K)$ 个 O_{random},O_j 交换对,并且每一个交换对都需要重新分派 $n-K$ 个非代表对象的所属簇标号,TPAM 算法也是如此,因此这 3 种方法的聚类总代价为 $O(K(n-K)^2)$.相比之下,FKMEDOIDS 聚类算法每一次聚类迭代的复杂度低得多,仅为 $O(nK)$.我们进一步分析复杂度,因为数值的比较、加减法运算相对于数值乘除法、开平方运算要快非常多,因此在耗时方面,前者相对于后者几乎可以忽略,那么整个聚类过程的时间消耗主要集中于样本点距离之间的计算.在 PAM 算法中,以引理 3.1 为基础,该引理中的情形(1)、情形(2)至少需要计算两次距离,分别为 $dist(P,Q_i)$, $dist(P,O'_j)$,按照距离计算次数且忽略比较运算,复杂度仍为 $O(K(n-K)^2)$.在 SPAM 方法中,首先引入了空间加速方法,以 $O(n+K^2)$的额外空间保存了 $K^2/2$ 个中心点之间的距离及 n 个样本点与其所属簇中心点之间的

距离,那么与 PAM 方法相比,SPAM 方法可以节省计算 $dist(P,Q_i)$,在引理 3.2、定理 3.1 中,$dist(P,Q_i)$, $dist(O_t,O'_j)$, $dist(O_i,Q_j)$, $dist(O_j,O'_j)$ 都无需重新计算,就可以确定非代表对象 P 在交换 O_{random},O_j 后所属簇;对于被交换中心点 O_j 所代表簇 C_j 来说,簇内样本点很难满足定理 3.1,但是有一些样本点仍可以满足引理 3.2 的条件,如果连引理 3.2 都无法满足,那就需要转到引理 3.1 中计算,仍只需计算 $dist(P,O'_j)$.如果按平均的方式认为 $n–K$ 个非代表对象平均分配到 K 个簇中,即每个簇的非代表对象样本点为$(n–K)/K$,按照距离计算次数且忽略比较运算,SPAM 算法复杂度近似为 $O(K(n–K)\cdot(n–K)/K)=O((n–K)^2)$.而 TPAM 复杂度应在 $O((n–K)^2)$与 $O(K(n–K)^2)$之间.虽然 FKMEDOIDS 每一次迭代的复杂度仅为 $O(nK)$,然而却需要在聚类之前计算出所有样本之间的距离矩阵及优化初始中心点时的样本密度排序,距离矩阵计算过程的时间复杂度为 $O(n^2)$,同时还需额外的 $O(n^2)$的空间开销以保存该距离矩阵,这比 SPAM 的 $O(n+K^2)$空间开销要大一个量级;同时,排序的最快算法也需要 $O(n\log n)$的时间复杂度.如果设聚类过程迭代了 t 次,根据上述分析,那么整个算法的复杂度及额外空间开销见表 1.

Table 1 Complexity and memory overhead of four algorithms

表 1 4 种算法复杂度与空间开销

算法	PAM	TPAM	FKMEDOIDS	SPAM
时间复杂度	$O(tK(n–K)^2)$	$O(tK(n–K)^2)$	$O(n^2+n\log n+tnK)$	$O(t(n–K)^2)$
额外空间开销	0	$O(n+K^2)$	$O(n^2)$	$O(n+K^2)$

表 1 中第 3 栏的额外空间开销指以 PAM 算法的空间开销为基础,我们将比 PAM 算法多出来的内存空间开销称为额外空间开销.从整个算法的时间复杂度来看,SPAM 要比 PAM,TPAM 低,当迭代次数 t 较小时,FKMEDOIDS 与 SPAM 差不多;反之,SPAM 复杂度要高于 FKMEDOIDS.然而,从额外空间开销来看,FKMEDOIDS 是最大的,SPAM 与 TPAM 是同等量级的.

5 实验结果及讨论

本节里将在实际及人工数据集上进行算法性能测试,并与 PAM,TPAM,FKMEDOIDS 算法进行比较.程序代码采用 C++实现,在 64G 内存的服务器上运行.

5.1 数据集

本次实验中,我们共使用 6 个数据集,其中 4 个数据集来自 UCI[25],并且人工生成两个随机数据集,数据集的具体描述见表 2.其中,6 个数据集的样本量、属性个数、类别个数均不同,对于人工数据集 RandData,我们以二维高斯分布为基础,分别以均值向量[10,10],[10,30],[10,50],[30,10],[30,30],[30,50],[50,10],[50,30],[50,50]及协方差矩阵[10,0;0,10]随机生成 900 个样本,每个类别均为 100 个样本.对于该二维人工数据集,我们可以直观地从图上看出其分布情况,如图 5 所示(图例 C_i 代表第 i 类数据).对于人工数据集 RandData2,与 RandData 的生成方式类似,只是该数据集生成 10 维 10 个类别的 1 000 个数据,每一个类别的数据,都是以一个维度上均值 10 其余为 0,协方差矩阵为 5(对角线上元素为 5),每个类别均为 100 个样本.

Table 2 Data set description

表 2 数据集描述

Data name	Size	Attributes	Class
Banknote	1372	5	5
Bupa	345	6	2
Iris	150	4	3
Libras	360	90	15
RandData	900	2	9
RandData2	1 000	10	10

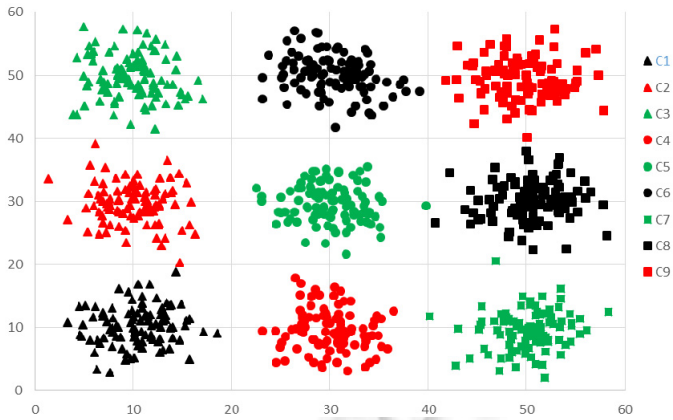


Fig.5 Randomly generated dataset RandData

图 5 随机生成数据集 RandData

5.2 性能评价指标

虽然本文致力于提高 *K*-medoids 聚类效率,但不以牺牲聚类准确性为代价,为此,首先考虑聚类总代价 *S*.

$$S=\sum_{i=1}^{K}\sum_{j=1}^{n_i}dist(x_j,O_i).$$

其次就是考虑聚类总时间(该时间不包括从文本中读入数据及结果输出到文本中),然而 PAM,TPAM,SPAM 都是随机初始化聚类中心,可能导致算法迭代次数(中心点更换次数)不一样而影响总时间的比较,文中将采用 10 次重复实验的平均值作为讨论结果.此外,还将给出聚类总迭代次数与平均迭代的运行时间.本文 SPAM 算法采用引理 3.1、引理 3.2、定理 3.1 及空间加速来提高聚类效率,本质上就是节省样本距离的计算次数.因此,文中也同样给出平均迭代样本距离的计算次数.

5.3 结果与讨论

对 PAM,TPAM,FKMEDOIDS,SPAM 这 4 种方法在表 2 中的 6 个数据集进行 10 次重复测试,首先,表 3 给出了 10 次重复实验的最小值、均值、最大值.在 3 种方法中,最小值、最大值几乎相等,libras 数据的最大值有微小差异,这说明 SPAM 方法与 PAM,TPAM 相比不会降低聚类精度.

Table 3 Minimum, mean, maximum clustering total cost of ten repeated experiments of dataset

表 3 数据集 10 次重复实验聚类总代价的最小值、平均值、最大值

代价	Banknote			Bupa		
	PAM	TPAM	SPAM	PAM	TPAM	SPAM
Min	5025.39	5025.39	5025.39	79.5649	79.5649	79.5649
Mean	5042.863	5046.665	5029.723	79.75918	79.69442	79.8887
Max	5066.57	5077.73	5055.96	80.2125	80.2125	80.2125
代价	Iris			Libras		
	PAM	TPAM	SPAM	PAM	TPAM	SPAM
Min	98.1312	98.1312	98.1312	334.994	334.88	334.88
Mean	98.20494	98.64738	98.4999	335.2584	335.3013	335.2522
Max	98.8686	98.8686	98.8686	335.697	335.745	335.697

将聚类总代价画成直观的直方图并注上相应数值,如图 6 所示,除了 FKMEDOIDS 算法外,PAM,TPAM,SPAM 在 6 个数据集上的聚类总代价没有明显差异,均能达到聚类最优结果,略微差异可以参考柱图顶端的 Cost 具体值;同时还说明,改进的 SPAM 算法保持了 PAM 算法的鲁棒性.FKMEDOIDS 算法在数据集 RandData 及 RandData2 上的聚类总代价明显高于其他 3 个算法,说明 FKMEDOIDS 算法较易陷入局部极值,这跟它中心点的优化方式及中心点的更新方式紧密相关,这种中心点的优化方式降低了 *K*-medoids 算法的鲁棒性,牺牲了聚

类质量.总体上来说,FKMEDOIDS 算法性能可以接受,但 PAM,TPAM,SPAM 这 3 种算法的聚类能力要更胜一筹.

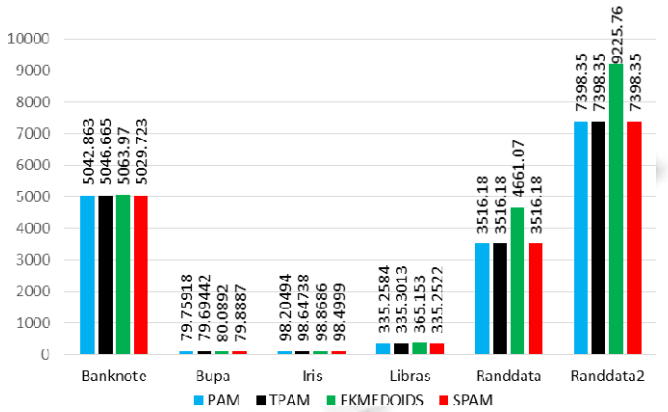


Fig.6 Culstering total cost

图 6 聚类总代价

表 4 给出了 4 种方法的聚类时间,SPAM 在所有数据集上都要比 PAM,TPAM 快.与 FKMEDOIDS 算法相比,加速效率与数据集本身特性还是相关的,这两个方法的差异也较大.尤其是在占用内存空间方面,FKMEDOIDS 要比 SPAM 多一个量级.

Table 4 Average clustering time of ten repeated experiments

表 4 10 次重复实验平均聚类时间

时间	PAM	TPAM	FKMEDOIDS	SPAM
Banknote	22.925	20.448	16.069	9.619
Bupa	0.373	0.299	0.461	0.132
Iris	0.066	0.063	0.033	0.037
Libras	116.690	102.247	0.582	56.403
Randdata	14.089	10.977	3.177	5.772
Randdata2	35.274	26.172	4.575	13.710

虽然聚类结果相差无几,但聚类过程中的迭代次数却有差异,如图 7 所示,尤其是在 Libras 数据集上,PAM,TPAM,SPAM 算法的迭代次数将近是 FKMEDOIDS 算法的 3 倍.而在 Bupa,Iris 数据集方面,SPAM 的迭代次数均比 FKMEDOIDS 算法要少.这说明 4 种算法的迭代次数与数据集本身的特性及初始中心点相关.因此,在聚类效率的比较方面,仅仅比较聚类总时间(见表 4)不足以反映 4 种方法之间的优劣,故需再考虑聚类平均迭代的运行时间.

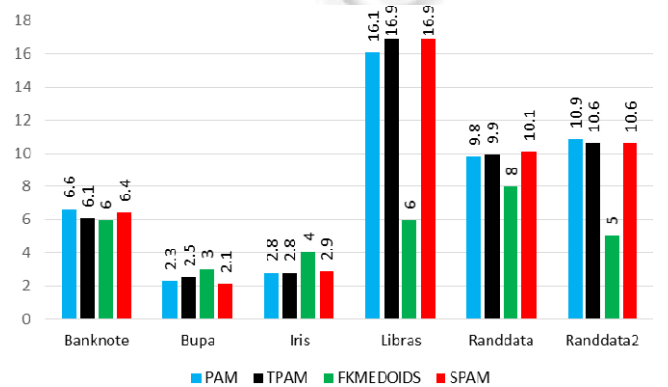


Fig.7 Clustering iteration number

图 7 聚类迭代次数

4 种算法的聚类平均迭代时间如图 8 所示,具体值参看柱图上端数字.从图中可以看出:SPAM 算法比 PAM 算法几乎快一倍(Banknote:1.311 倍,Bupa:1.580 倍,Iris:0.828 倍,Libras:1.172 倍, RandData:1.516 倍,RandData 2:1.502 倍);同时,比 TPAM 算法的聚类速度也快很多;维数较多时,效果会更明显(如 Libras,RandData2 两个数据集).我们还可以注意到:在 Iris,Libras,Randdata,RandData2 数据集上,FKMEDOIDS 算法比 PAM,TPAM,SPAM 都要快;然而在其他两个数据集上,SPAM 要快于 FKMEDOIDS 算法,因此,SPAM 与 FKMEDOIDS 算法的性能跟数据集特性相关,两种算法的性能都有可能超越对方.但是, FKMEDOIDS 算法额外开销了 $O(n^2)$的内存空间,而 SPAM 仅仅额外开销了 $O(n+K^2)$内存空间,比 FKMEDOIDS 少一个量级,当 SPAM 聚类速度低于 FKMEDOIDS 时,数值上也相差较小.综合起来,SPAM 利用了更少资源提高聚类效率.

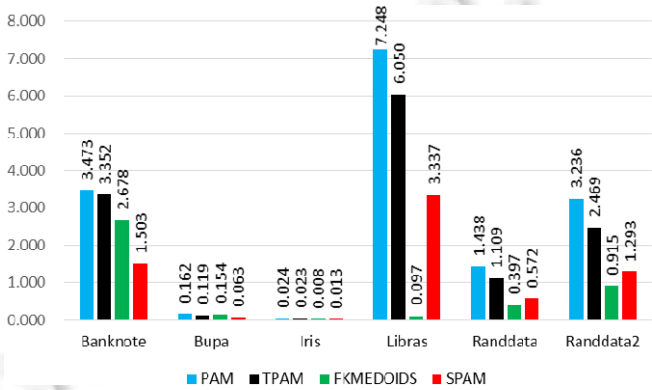


Fig.8 Clustering average iteration time

图 8 聚类平均迭代时间

聚类平均迭代时间仅仅是从结果方面进行的评价,本文提出采用引理 3.1、引理 3.2、定理 3.1 及空间加速方法加速 *K*-medoids 算法,本质上是节省大量的重复样本间距离计算,因此可以统计每次迭代过程中样本距离的计算次数,来反映本文方法的提高效率的实质,如图 9 所示.

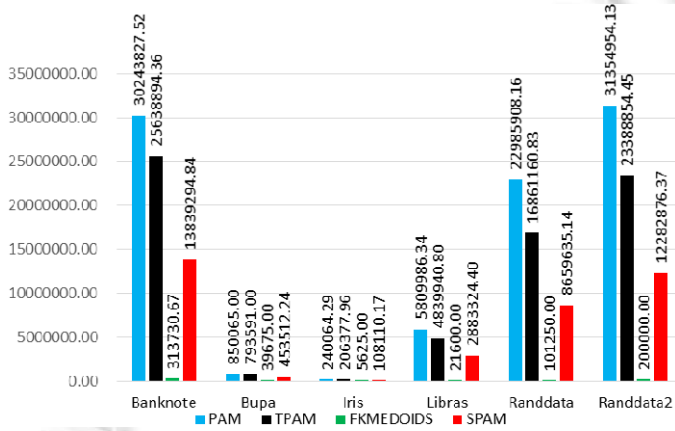


Fig.9 Average number of distance calculation in each iteration

图 9 每次迭代平均距离计算次数

首先分析 PAM,TPAM,SPAM 这 3 种算法,从图 9 中可以明显反映出:SPAM 算法计算样本距离次数要比 PAM,TPAM 算法少非常多,这就可以解释为什么 SPAM 算法可以提高聚类效率.图中显示,FKMEDOIDS 算法的样本距离计算次数几乎看不见.这是因为该算法在聚类之前进行了样本点的距离矩阵计算,在后面的聚类过程

中,只需要调用距离矩阵的样本点距离值即可,无需重新计算,使得该算法需要额外的 $O(n^2)$内存空间,这也合理解释了在算法时间复杂度上需要加上 $O(n^2)$,变成 $O(n^2+tnK)$.虽然图 9 中反映 FKMEDOIDS 计算距离次数最少,但是表 4 中的聚类总时间及图 8 中的聚类平均迭代时间都反映出,SPAM 算法在数据集 Banknote,Bupa 上的效率要高于 FKMEDOIDS.

上述实验数据集均比较小(样本最多的 Banknote 仅 1 372),只能说明在较小样本集情形下,SPAM 算法性能比 PAM,TPAM 优秀,而与 FKMEDOIDS 性能难分伯仲;对于不同的数据集,性能上均有可能超越对方.为了更全面地验证 SPAM 算法性能,我们随机生成一个 10 000 个样本的数据集,分别聚成 5,10,15,…,100 个簇,比较聚类平均迭代时间,见图 10.图中反映出 PAM,TPAM,SPAM 这 3 个算法随着聚类数目的增加,耗时也随之增加,但是 3 个算法的耗时增率却依次递减,即:SPAM 算法随着聚类数目的增多,耗时的增加量要明显小于 PAM,TPAM 算法.对于 FKMEDOIDS 聚类算法来说,总体趋势是随着聚类数目的增加,聚类时间降低,这跟其聚类之前需要计算距离矩阵与中心点优化的样本密度排序消耗时间相吻合.而与 SPAM 相比较而言,小于 50 个簇时,SPAM 效率要高于 FKMEDOIDS;大于 50 个簇时,性能反转.总体而言,对于大数据集来说,SPAM 性能要优于 PAM,TPAM;当簇个数非常大时,也无法保证 SPAM 性能超越 FKMEDOIDS,此时,有待于进一步提高效率.

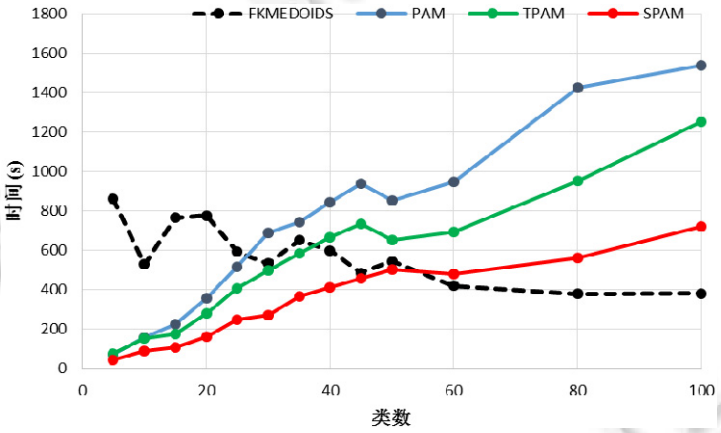


Fig.10 Average iteration time of big dataset clustering

图 10 大数据集聚类平均迭代时间

总体上来说,在聚类的质量方面,PAM,TPAM,SPAM 极其相近,且略优于 FKMEDOIDS 算法;在聚类效率方面,SPAM 算法完全优于 PAM,TPAM 算法,而与 FKMEDOIDS 算法互有优劣.这种结论上的差异,可以从 SPAM 与 FKMEDOIDS 原理上得到印证.FKMEDOIDS 算法事先计算并保存所有样本的距离矩阵,而 SPAM 算法无需执行这一步骤;在初始中心点选择方面,FKMEDOIDS 算法需要计算样本密度值并排序,选择前 K 个样本点作为初始中心点,而 SPAM 随机选择 K 个样本点;还有一个显著差异在中心点的更新方式上,这也导致了定理 3.1 无法优化 FKMEDOIDS 算法,而应该采用定理 3.2 的结论

6 结 论

本文针对 K-medoids 算法效率较低的问题,在 PAM,TPAM 算法的基础上,利用距离不等式,并结合 $O(n+K^2)$空间开销,提出了 SPAM 算法,进一步提升了 K-medoids 聚类的效率;在时间复杂度方面,在一次中心点更换过程中,即 t=1 时,从 PAM,TPAM 算法的 $O(K(n-K)^2)$降至 SPAM 算法的 $O((n-K)^2)$;同时,本文的改进保持了 K-medoids 算法的一切优点,并可以推广到现有的很多优化算法中,如基于抽样、中心点优化、并行化等.在 UCI 实际数据集及人工数据集的实验均表明,SPAM 算法的效率高于 PAM,TPAM 算法.与 FKMEDOIDS 算法的实验对比还表明:在比其少消耗内存空间的情况下,SPAM 算法效率并不一定比 FKMEDOIDS 差;与此同时,SPAM 算法在诸如

Libras 属性个数特别多的数据集上效率提升较少,仍需进一步加以改进.

References:

[1] Xu R, Wunsch D. Survey of clustering algorithms. IEEE Trans. on Neural Networks, 2005,16(3):645−678. [doi: 10.1109/TNN.2005.845141]

[2] Sun J, Liu J, Zhao LY. Clustering algorithms research. Ruan Jian Xue Bao/Journal of Software, 2008,19(1):48−61 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048]

[3] Frey BJ, Dueck D. Clustering by passing messages between data points. Science, 2007,315(5814):972−976. [doi: 10.1126/science.1136800]

[4] Frey BJ, Dueck D. Response to comment on "clustering by passing messages between data point". Science, 2008,319(5864):726. [doi: 10.1126/science.1151268]

[5] Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014,344(6191):1492−1496. [doi: 10.1126/science.1242072]

[6] Arora P, Deepali D, Varshney S. Analysis of *K*-means and *K*-medoids algorithm for big data. Procedia Computer Science, 2016,78:507−512. [doi: 10.1016/j.procs.2016.02.095]

[7] Peker M. A decision support system to improve medical diagnosis using a combination of *k*-medoids clustering based attribute weighting and SVM. Journal of Medical Systems, 2016,40:116. [doi: 10.1007/s10916-016-0477-6]

[8] Kaufman L, Rousseeuw PJ. Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley and Sons, 1990.

[9] Ng RT, Han JW. Clarans: A method for clustering objects for spatial data mining. IEEE Trans. on Knowledge and Data Engineering, 2002,14(5):1003−1016. [doi: 10.1109/TKDE.2002.1033770]

[10] Barioni MCN, Razente HL, Traina AJM, Jr CT. Accelerating *K*-medoids-based algorithms through metric access methods. The Journal of Systems and Software, 2008,81:343−355. [doi: 10.1016/j.jss.2007.06.019]

[11] Park HS, Jun CH. A simple and fast algorithm for *K*-medoids clustering. Expert Systems with Applications, 2009,36:3336−3341. [doi: 10.1016/j.eswa.2008.01.039]

[12] Zadegan SMR, Mirzaie M, Sadoughi F. Randed *K*-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets. Knowledge-Based Systems, 2013,39:133−143. [doi: 10.1016/j.knosys.2012.10.012]

[13] Kashef R, Kamel MS. Efficient bisecting *K*-medoids and its application in gene expression analysis. In: Proc. of the 5th Int'l Conf. on Image Analysis and Recognition. Varzim, 2008. 423−434. [doi: 10.1007/978-3-540-69812-8_42]

[14] Lai PS, Fu HC. Variance enhanced *K*-medoids clustering. Expert Systems with Applications, 2011,38:764−775. [doi: 10.1016/j.eswa.2010.07.030]

[15] Jiang YB, Zhang JM. Parallel *K*-medoids clustering algorithm based on hadoop. In: Proc. of the 5th IEEE Int'l Conf. on Software Engineering and Service Science (ICSESS). 2014. 649−652. [doi: 10.1109/ICSESS.2014.6933652]

[16] Yue J, Mao SJ, Li M, Zou XS. An efficient PAM spatial clustering algorithm based on MapReduce. In: Proc. of the 22nd Int'l Conf. on Geoinformatics. 2014. 1−6. [doi: 10.1109/GEOINFORMATICS.2014.6950803]

[17] Han LS, Xiang LS, Liu XY, Luan J. The *K*-medoids algorithm with initial centers optimized based on a *P* system. Journal of Information and Computational Science, 2014,11(6):1765−1773. [doi: 10.12733/jics20103217]

[18] Han LS, Xiang LS, Liu XY. *P* system based on the MapReduce for the most value problem. Journal of Information and Computational Science, 2014,11(13):4697−4706. [doi: 10.12733/jics20104502]

[19] Zhang Q, Couloigner I. A new and efficient *K*-medoids algorithm for spatial clustering. In: Proc. of the Int'l Conf. on Computational Science and Its Applications. LNCS 3482, Singapore: Springer-Verlag, 2005. 207−224. [doi: 10.1007/11424857_20]

[20] Chu SC, Roddick JF, Ran JS. An efficient *K*-medoids-based algorithm using previous medoid index, triangular inequality elimination criteria, and partial distance search. In: Proc. of the 4th Int'l Conf. on Data Warehousing and Knowledge Discovery, Vol.2454. Aixen-Provence, 2002. 63−72. [doi: 10.1007/3-540-46145-0_7]

[21] Chu SC, Roddick JF, Chen TY, Pan JS. Efficient search approaches for *K*-medoids-based algorithms. In: Proc. of the Int'l Conf. on Computers, Communications, Control and Power Engineering. 2002. 712a−715a. [doi: 10.1109/TENCON.2002.1181751]

[22] Chiang CS, Chu SC, Roddick JF, Pan JS. New search strategies and new derived inequality for efficient *K*-medoids-based algorithm. Chinese Journal of Electronics, 2007,16(1):82−87.

[23] Chu SC, Roddick JF, Pan JS. Improved search strategies and extensions to *K*-medoids-based clustering algorithms. Int'l Journal of Business Intelligence and Data Mining, 2008,3(2):212−231. [doi: 10.1504/IJBIDM.2008.020520]

[24] Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2012.

[25] Lichman M. UCI Machine Learning Repository. Irvine: University of California, School of Information and Computer Science, 2013. http://archive.ics.uci.edu/ml

附中文参考文献:

[2] 孙吉贵,刘杰,赵连宇.聚类算法综述.软件学报,2008,19(1):48−61. http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048]

附录

定理 3.2 的简要证明.

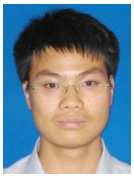
需要分 4 种情形.

1) $P\in C_i,O_i\in O\cap O'$,若 $dist(P,O_i)<\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$,此时说明在所有的新交换中心点中,点 P 与新中心点的距离均大于 $dist(P,Q_i)$,因此,$P\in C_i$;

2) $P\in C_i,O_i\in O\cap O'$,若 $dist(P,O_i)<\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$ 不成立,此时说明在所有的新交换中心点中,存在新中心点,使得点 P 与该新中心点的距离小于 $dist(P,Q_i)$,因此,P 需要重新分配到该新中心点(最小距离)所代表的簇中,即,满足 $dist(P,O'_l)=\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$;

3) 当 $P\in C_i,O_i\notin O'$,说明点 P 所在的簇 i 的中心点已经更换,若 $dist(P,O_i)>\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$,则说明存在新中心点,使得点 P 到该新中心点的距离要小于原始簇的距离 $dist(P,Q_i)$,此时需要 P 重新分配到该新中心点(最小距离)所代表的簇中,即,满足 $dist(P,O'_l)=\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$;

4) 当 $P\in C_i,O_i\notin O'$,说明点 P 所在的簇 i 的中心点已经更换,若 $dist(P,O_i)>\min_j\{dist(P,O'_j)\mid O'_j\in O'-O\}$ 不成立,则说明所有新更换的中心点到 P 的距离均大于原始簇的距离 $dist(P,Q_i)$,此时需要比较所有中心点,以确定 P 所属的簇,即,满足 $dist(P,O'_l)=\min_j\{dist(P,O'_j)\mid O'_j\in O'\}$.



余冬华(1988－),男,江西赣州人,博士生,主要研究领域为机器学习,生物信息学.

任世军(1962－),男,博士,教授,主要研究领域为聚类分析,图论.

郭茂祖(1966－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为机器学习,生物信息学.

刘晓燕(1963－),女,博士,副研究员,主要研究领域为生物信息学,数据挖掘.

刘扬(1976－),男,博士,副教授,CCF 专业会员,主要研究领域为机器学习,图像处理,计算机视觉.

刘国军(1979－),男,博士,讲师,CCF 专业会员,主要研究领域为机器学习,计算机视觉,模式识别.