

文章编号:1007-5321(2020)03-0118-07

DOI:10.13190/j.jbupt.2019-078

# 一种海量数据快速聚类算法

何倩<sup>1</sup>, 李双富<sup>1,2</sup>, 黄焕<sup>1</sup>, 徐红<sup>1</sup>

(1. 桂林电子科技大学 卫星导航定位与位置服务国家地方联合工程研究中心, 桂林 541004;

2. 广西交科集团有限公司, 南宁 530007)

**摘要:**为满足海量数据处理要求,提出了一种基于网格的  $K$ -means 快速聚类算法(SPGK)。设计基于网格质心的聚类簇个数选取算法,对数据进行网格划分得到每个网格的质心,将质心作为  $K$ -means 聚类的样本点,从而减少  $K$ -means 的欧氏距离计算次数。该算法基于 Spark 平台实现并行计算,进一步地提高了算法的运行效率。SPGK 不但能够获得良好的聚类效果,而且缩减了欧氏距离计算次数,适用于海量数据的快速聚类。在千万级数据集上的实验结果表明,SPGK 的性能明显优于现有的  $K$ -means++ 和基于  $K$  均值聚类的递归划分方法。

**关键词:**快速聚类; Spark; 最佳聚类初始点; 网格划分

中图分类号: TP311

文献标志码: A

## A Fast Clustering Algorithm for Massive Data

HE Qian<sup>1</sup>, LI Shuang-fu<sup>1,2</sup>, HUANG Huan<sup>1</sup>, XU Hong<sup>1</sup>

(1. State and Local Joint Engineering Research Center for Satellite Navigation and Location Service,

Guilin University of Electronic Technology, Guilin 541004, China;

2. Guangxi Jiaoke Group Company Limited, Nanning 530007, China)

**Abstract:** To meet the requirements of massive data processing, a grid-based  $K$ -means fast clustering algorithm (SPGK) is proposed. Selection for optimal clustering initial point and the number of clusters algorithm is presented. The grids of different clusters are meshed to obtain the centroid of each grid. These centroid points are used as sample points for  $K$ -means clustering, thereby reducing the number of Euclidean distance calculations of  $K$ -means. SPGK realizes parallel computation based on Spark platform, which further improves the running efficiency of the algorithm. SPGK not only obtains good clustering effect but also greatly reduces the number of Euclidean distance calculations, which is suitable for fast clustering of mass data. With 10 millions of data, the experiments show that SPGK is superior to the existing  $K$ -means++ and recursive partition based  $K$ -means clustering algorithms obviously.

**Key words:** fast clustering; Spark; best initial clustering point; grid generation

随着数据的大规模增长和信息系统的云服务化,如何利用海量数据(笔者认为数据量至少达到  $10^7$  以上)进行聚类挖掘从而获取有用价值,成为目前企业取得竞争优势的重点<sup>[1]</sup>。为适应  $10^7$  级别的

数据处理,不仅需要很大的存储空间,而且分析、处理和检索操作都非常困难<sup>[2]</sup>。如何实现海量数据的快速聚类成为当前聚类算法的研究重点。

在目前的聚类算法中,有基于密度的带噪声的

收稿日期: 2019-05-11

基金项目: 国家自然科学基金项目(61661015, 61967005); 广西创新驱动重大专项项目(AA17202024); 广西科技创新团队项目(2019GXNSFGA245004)

作者简介: 何倩(1979—),男,教授,博士生导师, E-mail: heqian@guet.edu.cn.

密度聚类方法<sup>[3]</sup> (DBSCAN, density-based spatial clustering of applications with noise) 算法、基于空间网格的统计信息网格<sup>[4]</sup> (STING, statistical information grid) 算法、层次聚类以及距离划分的  $K$ -means<sup>[5]</sup> 算法. 这些算法尽管理论上可以实现数据的聚类分析工作, 但是由于算法本身的特性以及技术的限制还不能适用于海量数据的处理.

$K$ -means 聚类速度快, 具有算法简单、易于实现等优势, 如今仍然广泛使用, 被列为机器学习的十大聚类算法<sup>[6]</sup> 之一. 然而, 随着数据量的不断增长,  $K$ -means 本身的计算特点严重影响了它的聚类效果以及执行效率. 首先,  $K$ -means 算法和  $K$ -means++<sup>[7]</sup> 等其他改进算法都面临着一个共同问题:  $K$  值的选取是随机性的. 因此导致聚类无法实现快速收敛而需要多次迭代计算, 从而大幅度降低了算法执行效率. 其次,  $K$  值的选择存在盲目性, 完全凭借经验或者使用其他方法, 都不高效.

近年来, 数据处理平台性能不断提升. Apache Spark<sup>[8]</sup> 使用弹性分布式数据集 (RDD, resilient distributed datasets) 的抽象数据结构以及基于有向无环图的抽象计算流程, 极大提升了复杂机器学习和数据挖掘分析的计算性能. He 等<sup>[9]</sup> 提出了一种基于 Spark 的并行化移动对象聚集模式挖掘的 RDD-DBSCAN+ 聚类算法, 很好地提高了聚类算法的效率.

针对当前  $K$ -means 以及相关改进算法存在的不足, 笔者提出一种基于网格的  $K$ -means 快速聚类算法 (SPGK, Spark based parallel grid  $K$ -means). 设计聚类簇的个数确定和最佳聚类初始点选取算法 (GCM, grid centroid method), 在 GCM 基础上依据不同类簇之间的最小距离进行网格划分, 计算每个网格的质心作为  $K$ -means 聚类的样本点. 针对  $K$ -means 需要大量计算欧氏距离的问题, 优化  $K$ -means 算法聚类初始点、 $K$  值的随机选取以减少欧氏距离的计算量. 基于 Spark 平台实现整个快速聚类算法.

## 1 相关工作

$K$ -means 系列算法优化的研究工作主要集中在降低计算复杂度方面, 包括基于确定  $K$  值的优化方法、基于减少欧氏距离计算次数的优化方法和基于并行化的优化方法等. 基于确定  $K$  值的优化方法通过提前计算获得合适的  $K$  值, 从而避免枚举  $K$  值的尝试和过多的迭代计算, 达到减少计算量的目标.

确定  $K$  值的方法有手肘法<sup>[7]</sup>、Gap Statistic<sup>[10]</sup>、基于混合距离的方法 (MBHD, method based on hybrid distance)<sup>[11]</sup> 等算法. 基于减少欧氏距离计算次数的优化方法注意到在对海量数据进行聚类时,  $K$ -means 聚类算法的计算量增长很快, 其中衡量样本点间相似度的欧氏距离计算所占比例较高. 为了降低  $K$ -means 的算法复杂度, Capo 等<sup>[12]</sup> 从减少欧氏距离计算次数的角度出发, 结合网格聚类的特点, 将网格内的样本点近似成一个网格质心进行  $K$ -means 聚类, 实现  $K$ -means 在海量数据下的快速聚类. 基于并行化的优化方法针对大数据背景下串行  $K$ -means 聚类效率低下的问题, 通过对聚类过程中每一次迭代的聚类中心相关计算的并行化, 提高了  $K$ -means 的可靠性和效率. Wu 等<sup>[13]</sup> 将  $K$ -means 并行化编程后在 Hadoop 分布式系统中运行, 提高了算法的运行效率. Wang 等<sup>[14]</sup> 通过研究  $K$ -means 的聚类边界问题, 使用 Spark 实现数据样本点间的距离计算和类簇中心点更新的并行计算.

与  $K$ -means 不同, 网格聚类是一种空间数据结构聚类算法, 因而计算效率很高. 通过结合网格实现海量数据的快速聚类是一种有效降低原有聚类算法复杂度的途径. 徐晓等<sup>[15]</sup> 在密度峰值聚类算法基础上结合网格聚类, 极大地降低了计算复杂度. 于彦伟等<sup>[16]</sup> 提出了一种基于网格的 DBSCAN, 以实现 DBSCAN 的快速聚类. 因此, 笔者提出方法将网格聚类思想引入  $K$ -means 聚类, 提高了  $K$  值和聚类初始点选取效率, 降低了欧氏距离计算次数.

## 2 基于网格的 $K$ -means 算法

### 2.1 问题描述

$K$ -means 算法时间复杂度问题可以描述为

$$t = m \left[ ndK + \sum_{i=1}^K \sum_{j=1}^n (x_j - c_i)^2 \right] \quad (1)$$

其中:  $m$  为迭代次数,  $n$  为数据量,  $d$  为数据集维数,  $K$  为类簇个数,  $x_j, c_i$  分别为数据集样本点和类簇中心点. 由式 (1) 可以得出, 当数据量级别达到  $10^7$  以上时,  $K$ -means 对于欧氏距离的计算量达到  $10^8$  以上, 相当于  $10^{10}$  级别以上的加减操作计算量. 同时,  $K$ -means 的聚类效果也非常依赖于  $K$  值以及聚类初始点的选取. 不合适的  $K$  值和过于接近的聚类初始点都会导致迭代次数增加, 从而导致更多的计算量.

网格聚类算法复杂度只与网格个数  $r$  相关. 因

为  $r \ll n$ , 所以网格聚类相比  $K$ -means 算法速度更快。但是, 网格聚类也存在着网格划分大小选择的问题, 当划分的网格足够大时, 会出现不同类簇的样本点划分到同一簇类中心, 将增大聚类误差。

基于网格的距离划分必须考虑权重问题, 在密度不同的网格中, 聚类中心应当偏向密度较大的网格而远离密度较小的网格, 并且权值系数与密度有关。设聚类中心为  $c_k$ , 网格  $g_1, g_2$  的密度为  $|g_1|, |g_2|$  且  $|g_1| \geq |g_2|$ , 则聚类中心的计算公式为

$$c_i = (g_2 + g_1) \frac{|g_1|}{|g_1| + |g_2|} \quad (2)$$

## 2.2 相关定义

**数据集合:** 在一个未知簇的个数的数据集样本  $U$  中, 有  $U = \{u_1, u_2, \dots, u_n\}$ , 其中  $u_n$  代表一个类簇集合,  $n$  为集合中数据总个数。

**网格集合:** 将数据集合  $U$  的数据区域划分成大小相对的网格区域而得到网格集合  $G = \{g_1, g_2, \dots, g_n\}$ , 其中  $g_x$  表示在网格集合  $G$  中的坐标位置且  $g_x \subseteq G, x \in \{1, 2, \dots, n\}, |g_x|$  表示  $g_x$  的密度大小。

**网格质心:** 对于网格集合  $G$  中的任意网格  $g_x$ , 其网格质心计算公式为

$$c_i = \frac{1}{|g_x|} \sum_{j=1}^n x_j \quad (3)$$

其中  $n$  为网格  $g_x$  的样本点个数。

**簇:** 在集合  $U$  中, 由属于同一类的样本点  $x_j (x_j \subseteq U)$  组成的样本点集合叫作簇。而在网格集合  $G$  中, 网格的簇  $g_k$  是由属于同一类网格  $g_x$  的质心点构成的, 有  $g_k = \{c_1, c_2, \dots, c_n\}$ 。

**质心中心区域:** 对于任意 2 个相邻的质心  $c_{n-1}, c_n$  都存在一个密度加权的中心距离区域, 称为质心中心区域  $a_c$ 。其计算公式为

$$a_c = (c_{n-1} + c_n) \frac{|c_n|}{|c_{n-1}| + |c_n|} \quad (4)$$

其中  $|g_{n-1}| \geq |g_n|$ 。

**类簇中心:** 由属于同一簇的网格质心通过加权求得的值为类簇中心。其计算公式为

$$s_n = \frac{1}{|s_n|} \sum_{x=1}^n |c_x| c_x \quad (5)$$

其中  $|s_n|$  为该簇的质心总数。

**距离均方误差:** 由各个数据集集中的样本点  $x_j$  与选取中心  $c_i$  的距离平方和误差得到, 用于分配样本点到最小距离的类簇中心点, 计算公式为

$$E = \sum_{i=1}^K \sum_{j=1}^n \min \|x_j - c_i\|^2 \quad (6)$$

其中  $K$  为聚类个数。

## 2.3 网格 $K$ 值和聚类中心选取算法

基于网格的  $K$  值和聚类中心选取算法 GCM 的核心思想是: 将所有数据样本点都映射到网格中, 去除密度低于阈值的网格, 合并相似的网格, 剩下的网格数即为  $K$  值。算法步骤如下:

- 1) 将数据映射到初始大网格中, 求网格质心;
- 2) 选择任意一个未被标记的质心并标记, 搜索该质心所在网格的密度是否不低于阈值, 是则进行步骤 3), 否则进行步骤 4);
- 3) 将该网格通过四叉树算法分裂成 4 个新网格, 重新计算它们的质心, 然后跳转到步骤 2);
- 4) 计算质心与其相邻的质心距离, 并通过权值距离得到中心位置区域, 判断该中心位置区域密度是否不低于密度阈值, 如果满足则标记这 2 个质心属于同一个簇, 否则标记为不同簇的质心;
- 5) 重复步骤 2), 直到所有质心遍历完毕;
- 6) 计算得到标记为同一样本的质心之间的权值距离中心点, 得到的中心点个数为  $K$  值, 中心所在的位置为初始聚类中心点。

算法伪代码如算法 1。

### 算法 1 网格 $K$ 值和聚类中心选取算法

输入: 未知样本个数数据集  $D$

输出: 质心和  $K$

```

1 centroids  $c_i \leftarrow g_i; g_i \leftarrow D$ 
2 While  $g_i$  hasn't been mark
3 foreach caculate  $c_i \leftarrow g_i$ 
4 if  $c_i \neq \emptyset$ , center  $\leftarrow c_i \cup c_{i+1}$ 
5 if center  $\neq \emptyset, c_i, c_{i+1} \in \text{cluster}_i$ 
6 else quadtree  $g_{i \in (1,2,3,4)} \leftarrow g_i$ , update  $c_i \leftarrow g_i$ 
7 else  $c_i, c_{i+1} \notin \text{cluster}_i$ 
8 endfor
```

## 2.4 网格 $K$ -means 聚类算法

GCM 得到  $K$  值和初始聚类中心之后, 利用初始聚类中心单次聚类可能将样本点错误划分。如图 1(a) 所示, 在同一网格中属于不同簇的样本点容易被识别为同一簇, 其中  $x_j$  为  $S_2$  的样本点, 但是被划分成了  $S_1$  的样本点。此外, 大网格划分也存在无法识别噪声点、对数据边缘识别效果差等问题。

为提高聚类的准确性, 需要在原来的基础上进行更加细致的网格划分, 笔者采取基于最小质心的边界划分策略。如在图 1(a) 中,  $x_j$  被错误划分到  $S_1$  中, 而以  $x_j$  中最大值为边界进行划分时,  $x_j$  便能够

被正确地划分到  $S_2$  中, 如图 1(b) 所示.

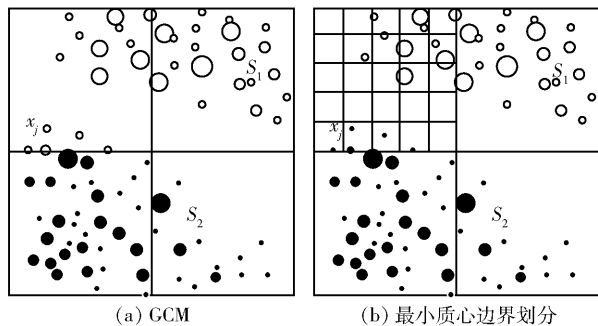


图 1 网格大小划分

基于网格的  $K$ -means 聚类算法步骤如下:

- 1) 将数据样本点映射到网格中, 取得每个网格的质心点和每个网格的密度;
- 2) 选取最小质心网格并进行网格划分;
- 3) 根据选取的聚类初始中心点, 计算每个网格质心到聚类中心的欧氏距离;
- 4) 根据权值距离分配网格到距离最近的聚类中心点;
- 5) 根据权值距离重新计算新的聚类中心点;
- 6) 迭代, 直到所有网格到对应聚类中心的权值欧氏距离最小.

算法伪代码如算法 2.

**算法 2** 基于网格的  $K$ -means 聚类算法

输入: 轨迹数据集  $D$

输出: 类簇中心和所属类簇中心网格集合

```

1 centroids  $g_i \leftarrow g_i; g_i \leftarrow D$ 
2 While  $g_i \neq \emptyset$ 
3 choose  $K$  and centroids  $c_i$ 
4 repeat
5  $c_i \leftarrow g_i$ 
6 update  $E$ 
7 until  $E$  do not change
8 endfor
```

## 2.5 算法分析

2.4 节设计的算法以网格质心进行  $K$ -means 聚类, 减少了距离的计算次数. 以下证明以质心为代表的聚类效果与基于样本点的聚类效果是一致的.

设数据样本点集合为  $X = \{x_1, x_2, \dots, x_n\}$ , 类簇中心点聚类集合为  $C = \{c_1, c_2, \dots, c_k\}$ , 网格质心集合为  $G = \{g_1, g_2, \dots, g_n\}$ , 样本点和网格质心均方误差分别为  $E_x, E_g$ .

$$E_g = \sum_{i=1}^k \sum_{j=1}^n (g_j - c_i)^2 = \sum_{i=1}^k \sum_{j=1}^n (g_j^2 - 2g_j c_i + c_i^2) \quad (7)$$

$$E_x = \sum_{i=1}^k \sum_{j=1}^n \|x_j - c_i\|^2 = \sum_{i=1}^k (\|x_1 - c_i\|^2 + \|x_2 - c_i\|^2 + \dots + \|x_n - c_i\|^2) = \sum_{i=1}^k (x_1^2 - 2x_1 c_i + c_i^2 + \dots + x_n^2 - 2x_n c_i + c_i^2) = \sum_{i=1}^k \sum_{j=1}^n (x_j^2 - 2x_j c_i + c_i^2) \quad (8)$$

可得  $E_g = E_x$ .

## 2.6 算法复杂度分析

1) GCM 算法. 与 Yang 等<sup>[17]</sup>提出的 MBHD 算法和  $K$ -means++ 算法中的手肘法进行对比. 假设在一个数据集  $U$  中, 样本点个数为  $n$ , 数据维度为  $d$ , 算法执行次数为  $m$ , 则 3 种算法的计算复杂度均可表示为  $O = m(ndk)$ . 假设数据集  $U$  的样本点个数  $n$  的数量级达到  $10^7$ ,  $K$  值为 5, 维度为 2, 以欧氏距离计算次数衡量算法复杂度, 则 MBHD 计算次数为  $6.0 \times 10^7$ , 手肘法在每次都能取到最小  $E$  的情况下最小计算次数为  $1 \times 10^9$ , 而 GCM 的最大计算次数仅为 48 次, 不到 MBHD 的百万分之一, 手肘法的两千万分之一.

2) 网格  $K$ -means 聚类算法. 与 Capo 等<sup>[12]</sup>提出的基于  $K$  均值聚类的递归划分方法 (RPKM, recursive partition based  $K$ -means) 以及  $K$ -means++ 算法进行对比. 对于一个样本点个数为 10 000, 维度为 2 的集合,  $K$ -means++ 运算次数为  $6.42 \times 10^5$ , RPKM 运算次数为  $2.68 \times 10^4$ , 而所提出的算法运算次数为  $3.97 \times 10^2$ , 是  $K$ -means++ 算法的 0.062%, RPKM 算法的 1.5%.

## 3 实验及分析

### 3.1 实验环境

为验证所提出的基于网格的  $K$ -means 算法的有效性, 分别采用人工数据集和 UCI 数据集进行实验. 此外, 由于 Spark 是基于 Scala 语言设计的, 并且与其他语言相比兼容性更好, 因此实验采用 Scala 编程实现. 实验用到 1 台普通计算机和 4 台工作站, 机器配置如表 1 所示. 实验使用 Scala 2.11.8 进行开发, 集群环境为 Spark 2.2.0, 运行模式为 Spark on yarn.

表 1 机器配置表					
类型	数量	CPU 型号	CPU 频率/GHz	CPU 核数	总内存大小/GB
工作站	4	AMD 6272	1.4	8	128
计算机	1	Intel i5-6500	3.2	4	8

3.2 GCM 算法评估

基于  $K$ -means++ 的手肘法、MBHD 算法均执行 10 次以上并选取最小的  $E$  作为实验数据,  $K$ -means++ 手肘法的值从 2 开始选取, 实验结果如图 2 所示.

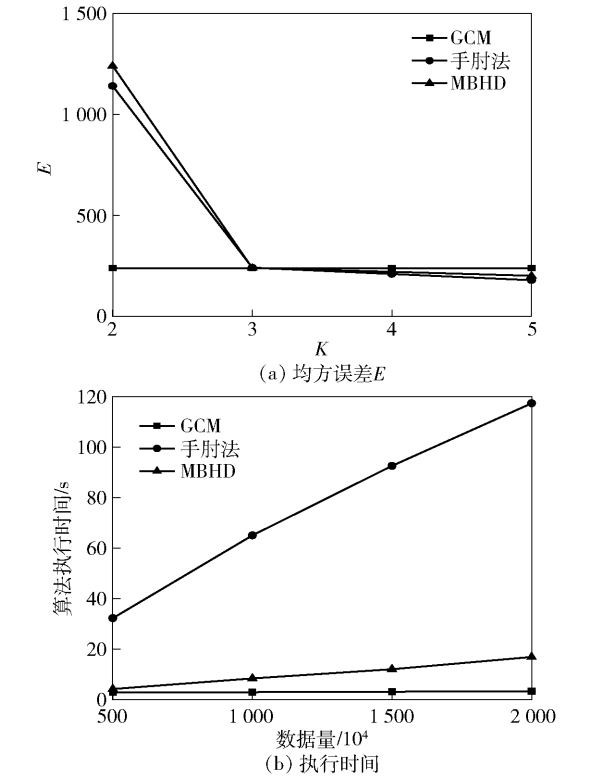


图2 GCM、手肘法、MBHD 算法性能对比

由图 2(a) 可知, 在手肘法和 MBHD 的折线中,  $K=3$  时  $E$  的值快速下降, 选取  $K$  值为 3. 而 GCM 算法在  $K=3$  时与手肘法相交,  $K$  值也为 3, 从而验证了该算法的准确性. 变化数据量分别对 500 万、1 000 万、1 500 万以及 2 000 万进行实验, 由图 2(b) 可知 GCM 执行时间低于手肘法以及 MBHD. 所以, GCM 算法与手肘法、MBHD 算法对比, 在准确度相同的情况下缩减了计算时间, 提高了效率.

3.3 SPGK 性能分析

为验证所提出的 SPGK 算法性能, 分别采用人工数据集以及 UCI 库中的葡萄酒数据进行实验, 而

且将 2 种数据都扩大到  $\{10^4, 10^5, 10^6, 10^7\}$  条. 首先在人工数据集下进行实验, 结果如图 3 所示.

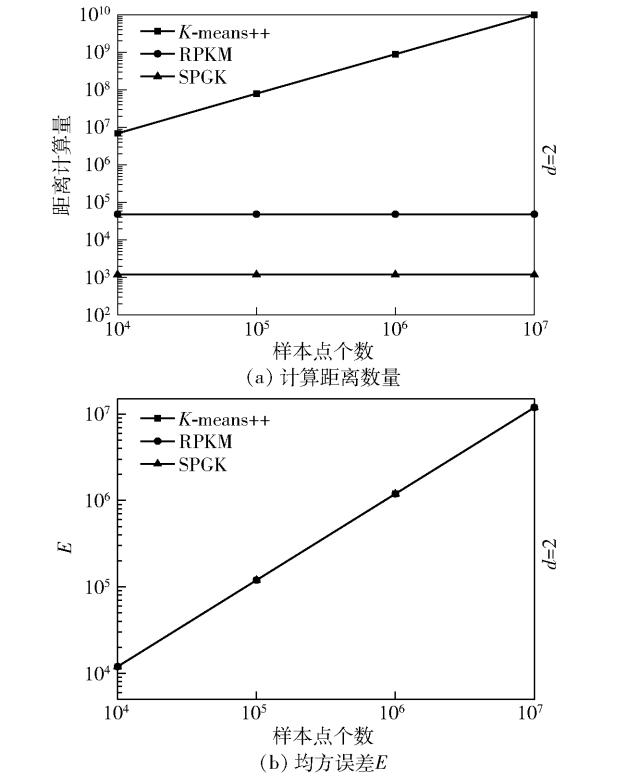


图3  $K$ -means++、RPKM、SPGK 性能对比(人工数据集)

由图 3(a) 可知, SPGK 的欧氏距离计算量要小很多, 同时 RPKM 算法和 SPGK 算法都能够在数据大量增长的情况下依然保持欧氏距离计算次数不变. 这是由于这 2 种算法都是以网格质心为代表进行  $K$ -means 聚类的, 网格聚类的复杂度只与网格个数有关, 因而不影响算法的执行效率.

如图 3(b) 所示, 对 RPKM 和  $K$ -means++ 算法进行了 10 次实验, 并取其中最小一次的  $E$  作为实验结果, 而 SPGK 算法只执行了一次, 就取得了与执行多次 RPKM、 $K$ -means++ 算法一致的实验结果. 其原因是 SPGK 算法在 GCM 中已经获得了准确的簇的个数和最佳聚类初始中心点, 所以可以极大地减少  $K$ -means 在计算欧氏距离时的次数. 在图 3 中, 当数据量为  $10^7$  个时,  $K$ -means++ 需要运行 10 次以上, 欧氏距离的计算次数达到  $10^8$  以上; 尽管 RPKM 也采用了网格划分的方式, 在一定程度上减小了计算欧氏距离的次数, 但是随着不断地进行二分裂划分, 计算次数呈指数级增长, 计算次数也达到了  $10^4$  次; 而 SPGK 算法一直维持在  $10^2$  级数上. 该实验结果说明, RPKM 和 SPGK 算法的时间复杂

度都不会随着实验数据的增长而受影响, 两者的时间复杂度只与网格的个数有关, 因此具备处理海量数据的能力, 同时 SPGK 算法性能更优. 值得一提的是, 笔者也对 SPGK 进行了多次实验, 结果发现 SPGK 的  $E$  值不会改变, 并且  $E$  值与 RPKM、 $K$ -means++ 的  $E$  值相等 (在  $10^{-13}$  的位数上可能不同). 这也进一步验证了 SPGK 算法在提高计算速度的同时, 仍保持了良好的聚类效果.

进一步使用 UCI 中的数据集进行验证, 实验结果如图 4 所示. SPGK 的计算效率依然比 RPKM 和  $K$ -means++ 算法高, 和人工数据集分析结果一致.

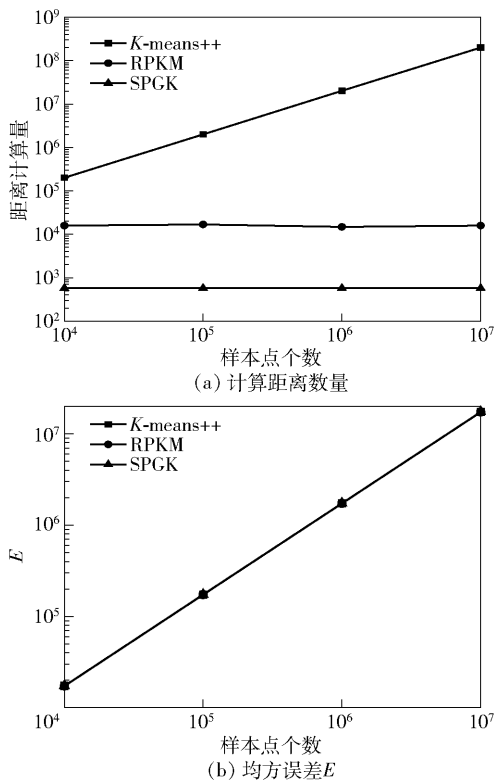


图 4  $K$ -means++、RPKM、SPGK 性能对比 (UCI 数据集)

## 4 结束语

针对  $K$ -means 系列算法在海量数据条件下存在运行效率低、得到的聚类结果可能不是局部最优解等问题, 提出了一种海量数据快速聚类算法. 提出聚类簇的个数确定和选取最佳聚类初始点算法, 提高了  $K$ -means 的聚类效果; 依据不同类簇之间的最小距离对数据进行网格划分, 计算每个网格的质心作为  $K$ -means 聚类的样本点, 提高了  $K$ -means 算法的运行速度, 且算法的复杂度与原始数据的数量无

关. 基于 Spark 平台实现了整个快速聚类算法, 实验结果表明, SPGK 不仅能够处理海量数据并且得到了良好的聚类效果, 明显优于现有的  $K$ -means++ 和 RPKM 聚类算法, 在大数据分析中具有广阔的应用前景.

## 参考文献:

- [1] Gahar R M, Arfaoui O, Hidri M S, et al. An ontology-driven MapReduce framework for association rules mining in massive data[J]. Procedia Computer Science, 2018, 126: 224-233.
- [2] Hidri M S, Zoghalmi M A, Ayed R B. Speeding up the large-scale consensus fuzzy clustering for handling big data[J]. Fuzzy Ets and Systems, 2018(348): 50-74.
- [3] Ester M, Kriegel H P, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise[C] // International Conference on Knowledge Discovery & Data Mining. New York: ACM, 1996: 226-231.
- [4] Wang W. STING: a statistical information grid approach to spatial data mining[J]. Proc of Very Large Database Conf, 1997(15): 186-195.
- [5] Hartigan J A, Wong M A. A  $K$ -means clustering algorithm[J]. Journal of the Royal Statistical Society: Series C (Applied Statistics), 1979, 28(1): 100-108.
- [6] Wu X, Kumar V, Ross J, et al. Top 10 algorithms in data mining [J]. Knowledge And Information Systems, 2007 (14): 1-37.
- [7] Arthur D, Vassilvitskii S.  $K$ -means++: the advantages of careful seeding[J]. Proceedings of Theghteenth Annual Acm Siam Symposiumon Discrete Algorithms Society for Industrial & Applied Mathematics, 2007, 11(6): 1027-1035.
- [8] Shmeis Z, Jaber M. Fine and coarse grained composition and adaptation of spark applications[J]. Future Generation Computer Systems, 2018: 629-640.
- [9] He Qian, Chen Yiting, Dong Qinghe, et al. A parallel clustering and test partitioning techniques based mining trajectory algorithm for moving objects[C] // 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). Guilin: Cuilin University of Electronic Technology, 2017: 455-462.
- [10] Tibshirani R, Hastie W T. Estimating the umber of clusters in a data Et via the gap statistic[J]. Journal of the Royal Statistical Society, 2001, 63(2): 411-423.
- [11] Ishioka T. Extended  $K$ -means with an efficient estima-

- tion of the number of clusters[C]//Intelligent Data Engineering & Automated Learning-ideal, Data Mining, Financial Engineering, & Intelligent Agents, Second International Conference. HongKong: Morgan Kautmann Publishs Inc, 2000.
- [12] Capo M, Perez A, Lozano J A. An efficient approximation to the  $K$ -means clustering for massive data[J]. Knowledge-Based Systems, 2017, 117(2): 56-69.
- [13] Wu Kehe, Zeng Wenjing, Wu Tingting, et al. Research and improve on  $K$ -means algorithm based on hadoop[C]//IEEE International Conference on Software Engineering & Service Science. Piscataway: IEEE, 2015: 334-337.
- [14] Wang Bowen, Yin Jun, Hua Qi, et al. Parallelizing  $K$ -Means Based Clustering on Spark[C]//International Conference on Advanced Cloud and Big Data. Piscataway: IEEE, 2016: 31-36.
- [15] 徐晓, 丁世飞, 孙统风, 等. 基于网格筛选的大规模密度峰值聚类算法[J]. 计算机研究与发展, 2018, 55(11): 79-89.
- Xu Xiao, Ding Shifei, Sun Tongfeng, et al. Large-scale density peaks clustering algorithm based on grid screening[J]. Journal of Computer Research and Development, 2018, 55(11): 79-89.
- [16] 于彦伟, 贾召飞, 曹磊, 等. 面向位置大数据的快速密度聚类算法[J]. 软件学报, 2018, 29(8): 2470-2484.
- Yu Yanwei, Jia Zhaoifei, Cao Lei, et al. Fast density-based clustering algorithm for location big data[J]. Journal of Software, 2018, 29(8): 2470-2484.
- [17] Yang Jie, Ma Yan. Zhang Xiangfen, et al. An initialization method based on hybrid distance for  $K$ -means algorithm[J]. Neural Computation, 2017, 29(11): 3094-3117.

(上接第 111 页)

- [7] Lei H J, Zhang H, Ansari I S, et al. On secrecy outage of relay selection in underlay cognitive radio networks over Nakagami- $m$  fading channels[J]. IEEE Transactions on Cognitive Communications and Networking, 2017, 3(4): 614-627.
- [8] Ho-Van K, Do-Dac T. Analysis of security performance of relay selection in underlay cognitive networks[J]. IET Communications, 2018, 12(1): 102-108.
- [9] Liu Y W, Mousavifar S, Deng Y S, et al. Wireless energy harvesting in a Cognitive Relay Network[J]. IEEE Transactions on Wireless Communications, 2016, 15(4): 2498-2508.
- [10] Nguyen D K, Jayakody D N, Chatzinotas S, et al. Wireless energy harvesting assisted two-way cognitive relay networks: protocol design and performance analysis[J]. IEEE Access, 2016, 5: 21447-21460.
- [11] Xu Chi, Zheng Meng, Liang Wei, et al. Outage performance of underlay multihop cognitive relay networks with energy harvesting[J]. IEEE Communications Letters, 2016, 20(6): 1148-1151.
- [12] Li Mu, Yin Hao, Huang Y Z, et al. Physical layer security in overlay cognitive radio networks with energy harvesting[J]. IEEE Transactions on Vehicular Technology, 2018, 67(11): 11274-11279.
- [13] Maji P, Prasad B, Roy S D, et al. Secrecy outage of a cognitive radio network with selection of energy harvesting relay and imperfect CSI[J]. Wireless Personal Communications, 2018, 100(2): 571-586.
- [14] Zou Yulong, Wang Xianbin, Shen Weiming, et al. Security versus reliability analysis of opportunistic relaying[J]. IEEE Transactions on Vehicular Technology, 2014, 63(6): 2653-2661.
- [15] Gradshteyn I S, Rythik I M, Jeffrey A, et al. Table of integrals, series, and products[M]. New York: Academic Press, 2007.