华东交通大学学报
Journal of East China Jiaotong University

基于原始几何概念的 Ball k-means 聚类算法

王　森，王悦琳，詹小秦，梁志坚

（华东交通大学理学院，江西 南昌 330013）

摘要：Ball k-means 作为 k-means 的精确加速算法，能在保证聚类结果一致的同时提升效率，但高维数据下加速效果衰减且邻居簇查找复杂度高。为提升其高维数据适应能力并保留其在大 k 聚类中的加速优势，本文提出一种基于原始几何概念的优化算法 Ball k-means-G*。在数据点分配步骤中引入原始几何结构，将数据点与候选质心降维至二维，确定距离下界以跳过冗余高维计算；在邻居簇查找中引入强迫点机制，将质心降至三维，结合原算法判断条件提前排除非邻居簇距离计算。在不同规模、维度的真实数据集实验显示，该算法性能显著优于 5 种对比精确加速 k-means 算法。与原始 Ball k-means 比，各维度数据集效率平均提升约 20.66%。Ball k-means-G*有效缓解了 Ball k-means 在高维数据下的性能衰减问题，适用于各类维度与大 k 聚类任务。

关键词：聚类算法；k-means 聚类；精确加速变体；大规模数据；高维数据

中图分类号：TP391　　**文献标志码：**A

Ball k-means Clustering Algorithm Based on Primitive Geometric Concepts

Wang Sen, Wang Yuelin, Zhan Xiaoqin, Liang Zhijian

（School of Science, East China Jiaotong University, Nanchang 330013, China）

Abstract: Ball k-means, as an accurate acceleration algorithm for k-means, improves efficiency while guaranteeing consistent clustering results. However, its acceleration effect diminishes on high-dimensional data, and the neighbor-searching step involves high complexity. To enhance its adaptability to high-dimensional data while preserving its acceleration advantages in large-k clustering, this paper proposes an optimized algorithm named Ball k-means-G*. In the data point assignment step, a primal-geometry structure is introduced by projecting data points and candidate centroids into two-dimensional space, where a lower bound for distances is established to skip redundant high-dimensional computations. In the neighbor-cluster search, a forced-point mechanism is incorporated: centroids are reduced to three dimensions, and combined with the original algorithm’s pruning conditions to eliminate non-neighbor distance calculations early. Experiments on real datasets with different scales and dimensions show that the proposed algorithm significantly outperforms five existing exact accelerated k-means methods. Compared with the original Ball k-means, it achieves an average efficiency improvement of approximately 20.66% across datasets of various dimensions. The results demonstrate that Ball k-means-G* effectively alleviates the performance degradation of Ball k-means in high-dimensional settings and is applicable to clustering tasks across various data dimensions and with large k values.

Key words: Cluster algorithm; k-means clustering; Accurate acceleration variant; Large scale data; High dimensional data

数据挖掘的主要目标是从收集的数据中提取有意义和有形的信息[2]，但是由于大多数数据是以任意形式和类别收集的，特别是数据对象的特征未知时，使得这些数据很难分析。聚类就是对此类未标记数据进行有意义的分组，该方法致力于确保簇内的数据对象具有高相似性，而簇间的相似性则尽可能低[1]。现实世界数据集具有高密度和高维度的特点，给标准聚类算法带来了巨大挑战。

k-means 聚类算法是数据挖掘中十大聚类算法之一[3][4]，简单性和低计算复杂度使得 k-means[5]算法

在许多领域被广泛接受。Lloyd 于 1982 年发表了该算法，k-means 算法是基于距离函数的计算[6]，对于具有凸分布的数据集，该算法能对其进行较优的划分[7]。原始 k-means 算法中的聚类质心是随机初始化的，这可能会导致每组初始聚类质心的迭代次数不同，聚类结果也不同。原始 k-means 算法时间复杂度为 $O(nkT)$，其中 n 表示数据大小，k 表示聚类数，T 是算法收敛前的迭代总数。在大规模的聚类中，k 的值通常是几千或更多，这可能会在标准 k-means 算法中造成巨大时间开销。

对于 k-means 算法主要有三类研究方向：初始质心的选择，近似实现的加速和精确算法的加速[8]。初始质心选择方法是为了解决原始质心的随机初始化而导致的迭代次数和最终聚类结果的差距[9]。k-means++[10]是初始质心选择改进方法中最突出的一种。Bachem 等人在 k-means++的基础上，用 MCMC（Markov Chain Monte Carlo）采样方法取代 D2-seeding[11]。近似 k-means 算法通过近似聚类结果来加速 Lloyd 的原始 k-means。Chakraborty[21]等人设计了一种基于强一致稀疏中心的聚类方法，适用于高维数据。薛丁文等人使用单 kd 树的近似近邻算法和基于多 kd 树的交叉搜索算法[12]，将 k-means 时间复杂度降为 $O(n\log k)$。Sculley 引入小批量采样方法，在不增加计算成本的情况下提升了收敛效果[13]。Ortega 等人引入质心运动等距阈值的启发式规则，改进了算法在大型数据集上的性能[14]。加速 k-means 的另一种解决方案是精确加速的 k-means 算法。Elkan 根据数据点到质心的距离将三角不等式应用于边界，以避免一些距离计算，从而显著减少距离计算量[15]。Hamerly 通过将每次迭代的下限数量减少到 n 来改进该算法[16]，这导致此算法对“大移动”变得敏感[17]。为改进 Hamerly 算法，Newling 和 Fleuret 提出了 Exponion 算法。此外，他们提出了一种技术，使所有基于边界的方法的边界更加严格，从而消除进一步的冗余距离计算[18]。Xia 等人提出了 Ball k-means 算法，该算法使用活动区域和静止区域减少距离计算的次数，对于大 k 聚类情况很有效[8]。Ismkhan[19]等人提出了 k-means-G*，该方法利用原始几何概念将数据进行降维，提前计算下限以减少高维计算。

针对 Ball k-means 算法的不足，本文提出一种基于原始几何的 Ball k-means 算法(Ball k-means-G*)。该算法在数据分配步骤将数据和候选质心降至二维，可以确定未知距离的下界，从而跳过部分高维距离计算。在邻居簇搜索时进一步引入强制点，减少计算质心与质心之间距离计算，加速其在大 k 高维情况下的搜索速度。研究 Ball k-means 算法各种不同类型的数据集上的缺陷，针对不足之处进行改进，有助于提升算法的性能，对于数据挖掘和聚类分析有较大现实意义。

1 Ball k-means 聚类算法

1.1 k-means 算法核心思想

给定一组 n 个数据点，$\{p_1, p_2, \cdots, p_n\} \subset R^d$，其中每个数据点代表一个 d 维向量，k-means 算法旨在通过最小化每个样本到其最近簇质心的距离平方之和[20]，将这 n 个样本划分为 k 个簇。表示如下：

$$J(p,c)=\sum_{j=1}^{k}\sum_{p_i\in C_j}\left|p_i,c_j\right|^2 \tag{1}$$

公式(1)中 c_j 是簇 C_j 的质心，且为最接近 p_i 的质心，$\left|p_i,c_j\right|$ 表示 p_i 和 c_j 之间的欧几里得距离。为了优化上述的目标函数，k-means 算法迭代进行分配和更新，直到簇质心稳定。

分配：将每个数据点 p_i 分配给最近质心：

$$b(p_i)=\operatorname{argmin}_{j=1,\ldots,k}\left\{\left|p_i,c_j\right|^2\right\} \tag{2}$$

更新：用分配给簇 C_j 的数据点更新质心 c_j：

$$c_j = \frac{1}{|C_j|}\sum_{i=1}^{n}\{p_i | b(p_i) = j\} \tag{3}$$

其中$|C_j|$表示分配给C_j的采样点的数量。

1.2 Ball k-means 算法核心思想

Ball k-means[8]是一种加速精确 k-means，它用球描述每个簇，重点是减少点与质心距离的计算。

设当前查询球簇为C，其质心c与半径r定义如下：

$$c = \frac{1}{|C|}\sum_{i=1}^{|C|} p_i,\ \ r = \max(|p_i, c|) \tag{4}$$

其中p_i是分配给C的点，$|C|$表示C中的样本数。设球簇C_i质心为c_i，若C的半径r满足以下不等式，则C_i是C的邻居簇：

$$\frac{1}{2}|c, c_j| < r \tag{5}$$

若C_i是C的邻居簇，则C中的一些点可以被移动到C_i中，否则C中没有点可以移动到C_i中。

设$\{N_C\}$表示C的相邻簇质心集且$|N_C| = k'$。设$k' \neq 0$（否则C没有邻居），c_i和c_{i+1}分别表示C的i最近和$(i+1)$最近邻簇的质心$(i < k')$，$\forall p \in C$。则C的稳定区域是半径为$\left(\min(|c, c_i|)\right)/2$，质心为$c$的球形区域。若$p$在稳定区域中，即满足：

$$|p, c| \le \frac{1}{2}\min(|c, c_i|) \tag{6}$$

则p在当前迭代中不会移动至其它任何簇中。静止区域以外的称为活动区域，而活动区域被划分为一些环形区域。C的第i个环形面积R_C^i，定义为：

$$R_C^i = \begin{cases} \frac{1}{2}|c, c_i| < |p, c| \le \frac{1}{2}|c, c_{i+1}|, 0 < i < k' \\ \frac{1}{2}|c, c_i| < |p, c| \le r,\ \ i = k' \end{cases} \tag{7}$$

第i个环形区域中的点仅参与簇C及其前i个最近邻簇内的分配。

若质心c_j和c_i间的距离符合以下公式，可以省略其距离计算：

$$|c_i^t, c_j^t| \ge 2r_i^t + \delta(c_i^t) + \delta(c_j^t) \tag{8}$$

其中，c_i^t表示第t次迭代中簇c_i的质心，$\delta(c_i^t)$表示在第t次迭代中的移动距离。表示公式如下：

$$\delta(c_i^t) = |c_i^t, c_i^{t-1}| \tag{9}$$

若查询球簇C的所有相邻球簇在一次迭代中都是稳定的，则C将不会参与下一次迭代的距离计算，因为其稳定区域和活动区域都不会改变。不过 Ball k-means 算法在搜索邻居簇的最差时间复杂度达到了$O(k^2)$，并且在高维度情况下，该算法的优势也有减少。

1.3 通过稳定区域和环形区域过滤点的过程

算法 1 通过稳定区域和环形区域过滤点的过程

输入：查询球簇C及其邻居集N_C

输出：球簇C中数据点的分配结果

Step 1： 若 迭代次数不为 1 且球簇 C 稳定且其所有的相邻球簇未改变且稳定，则跳过当前球簇。
Step 2： 将邻居集 N_C 按从小到大排序。
Step 3： 循环 1（遍历球簇 C 的所有数据点 p ）开始：
Step 4： 若数据点 p 在稳定区域内，则跳过该点。根据公式(7)，判断 p 属于球簇 C 的第 h 个环形，然后计算 p 到它的前 h 个最近邻簇质心的距离，并将 p 分配给最近的簇。
Step 5： 循环 1 结束。

2 优化算法 Ball k-means-G*

为了进一步加速 Ball k-means，提高其在高维数据情况下的效率，本小节提出了优化算法 Ball k-means-G*。其优化体现在两方面：一是在分配步骤中，利用原始几何[19]结构加速寻找数据点所属簇，可以在确定数据点所属环形后进一步减少距离计算；二是在查找邻居簇时，引入强迫点[19]以更快找到非邻居簇，避免不必要的距离计算。

2.1 分配步骤加速

设当前质心为点 M，邻居簇个数为 k'，距离点 M 第 $k'/2$ 近的质心为点 N，O 为其它任意质心。如图 1(a)所示，点 M，N 确定直线 MN，线段 pH_p 和 OH_O 都垂直于直线 MN，θ 为向量 pH_p 和 OH_O 之间的夹角。对于 $\theta\in[0,\pi]$，有 $f(\theta)=|p,O|^2$，且 $f(\theta)$ 为增函数[19]。

定义 1 在 d 维空间中，对于不同的点 M，N，O 和 p，除 $|p,O|$ 以外，每对点之间的所有距离都是已知的，那么如果 p 位于由 M，N，O 表示的平面上，则 $|p,O|$ 最小化，即 $\theta=0$ 时，$|p,O|$ 值最小。

故令 $\theta=0$，且将点 M 视为原点建立坐标系，x 轴方向为从 M 到 N，如图 1(b)所示。这是一个抽象图，因为在实际情况下 H_p 和 H_O 都可能落在线段 MN 上。

通过计算点 O 和 p 在坐标系下的二维坐标，可以算出 $|p,O|$ 的最小值，计算点 q 二维坐标过程如下：

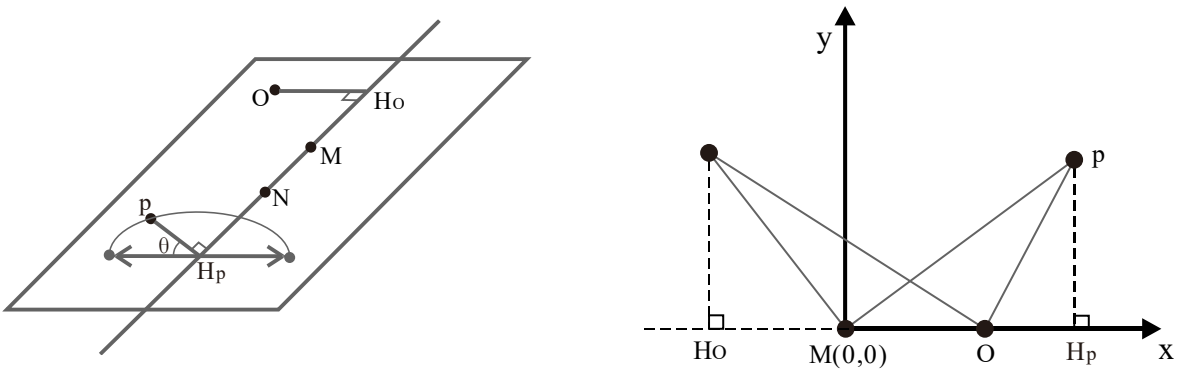
首先，对于边长为 a_1，a_2，a_3 的三角形，μ 为边 a_3 的对角，则通过余弦定理可知：

$$\cos\mu=\frac{a_1^2+a_2^2-a_3^2}{2a_1a_2} \tag{10}$$

根据公式(10)获得 $cos\angle pMN$，易得出点 p 二维坐标计算如下：

$$x_p=|p,M|\cos\angle pMN \tag{11}$$

$$y_p=\sqrt{|p,M|^2-x_p^2} \tag{12}$$



(a) 点 p 在 d 维状态下的位置示意图　　(b) 令 M 为原点的坐标系

图 1 点 p 在 d 维和二维状态的解释图

Fig. 1 Explanation diagram of point p in d-dimension and two-dimensional state

利用原始几何加速分配点的过程如算法 2 中所示。其中选择在遍历数据点之前建立 kd 树，以避免

查找每个数据点时重复建立 kd 树造成额外消耗。

算法 2 利用原始几何加速过滤点的过程

输入：查询球簇 C 及其邻居集 N_C

输出：球簇 C 中数据点的分配结果

Step 1： 若 迭代次数不为 1 且球簇 C 稳定且其所有的相邻球簇未改变且稳定，则跳过当前球簇。

Step 2： 将邻居集 N_C 按从小到大排序。

Step 3： 将球簇 C 质心作为点 M，距离点 M 第 $h/2$ 近的质心为点 N，根据公式(11)(12)计算邻居集 N_C 中每个质心的二维坐标 N_C_2D，并在 N_C_2D 上建立 kd 树便于查找。

Step 4： 循环 1（遍历球簇 C 中的每个数据点 p ）开始：

Step 5： 若数据点 p 在稳定区域内，则跳过该点。根据公式(7)，判断点 p 属于球簇 C 的第 h 个环形。

Step 6： 若 $h \le 5$，则将前 h 个最近邻簇质心作为候选质心集，跳过 Step 7。

Step 7： 根据公式(11)(12)计算点 p 的二维坐标 p_2D，将 p_2D 做为查找点在 kd 树中查找小于 $|p,M|$ 的点，并将查找结果与邻居集 N_C 前 h 个邻居簇的并集作为候选质心集。

Step 8： 遍历候选质心集，计算数据点 p 到候选质心的距离，并将数据点 p 分配给最近的簇。

Step 9： 循环 1 结束。

2.2 查找邻居簇加速

在查找邻居簇中，为进一步减少距离计算量，考虑在上一部分方法的基础上引入强迫点[19]以跳过更多的距离计算。虽然程序复杂度有所增加，但由于簇数 k 在一般情况下远小于数据量 n，整体计算负担仍在可接受范围内；这样既能够保证结果准确性，又实现了更高效的计算。

定义 2 在 d 维空间中，对于不同的点 M，N，O 和 p，除了 $|p,O|$ 以外，每对点之间的所有距离都是已知的，且有另外一点 F，F 与 M，N，O 和 p 之间的距离是已知的，那么说 F 迫使 $\theta>0$，F 被称为一个强迫点[19]。

当引入强迫点 F 时，增加了条件 $|p,F|$，θ 应在范围 $[0,\pi]$ 中进行调整，且从节 2.1 可知距离 $|p,O|$ 随着 θ 增大而增大。另一方面，在范围 $[0,\pi]$ 中的单个点 $\theta=0$ 处的概率始终为零，故 F 迫使 $\theta>0$ 的概率为 1，即点 p 必不在平面 MNO 上，因此坐标系为三维。易知，点 O 坐标可通过算法 2 获得，且 $z_O=0$。

在此情况下，对于点 p 三个坐标计算如下：

$$x_p = |p,M|\cos\angle pMN \tag{13}$$

$$|p,H_p| = \sqrt{|p,M|^2 - x_p^{\,2}} \tag{14}$$

$$y_p = |p,H_p|\sin\theta \tag{15}$$

$$z_p = |p,H_p| - |p,H_p|\cos\theta \tag{16}$$

为了确定角度 θ，过点 F 有平面 α 垂直于平面 MNO，如图 2(a)所示，FH_F 垂直于直线 MN，FF_{MNO} 垂直于平面 MNO，将图 2(a)中的元素映射到平面 α，得图 2(b)。从图 2(b)中易看出：

$$\theta = \angle FH_F p - \angle FH_F O \tag{17}$$

为了获得 $\angle FH_F p$ 和 $\angle FH_F O$，首先要获得 Fp_Map 和 FO_Map 的值，其计算如下：

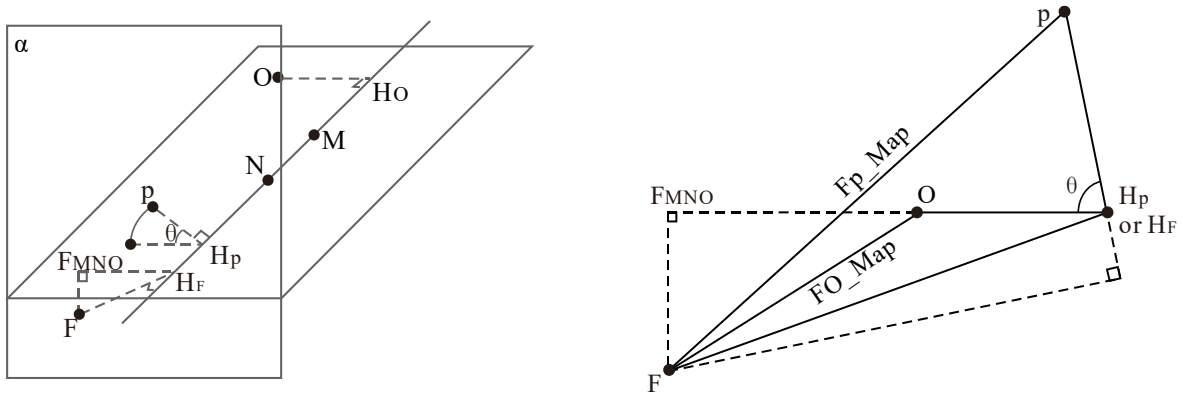
$$Fp\_Map = \sqrt{|p,F|^2 - |x_P - x_F|^2} \tag{18}$$

$$FO\_Map = \sqrt{|O,F|^2 - |x_O - x_F|^2} \tag{19}$$

由图 2(b)和余弦定理，即公式(10)，即可计算出 $\angle FH_F p$ 和 $\angle FH_F O$，如下：

$$\angle FH_F p = \cos^{-1}\frac{{y_F}^2 + {y_p}^2 - Fp\_Map^2}{2y_F y_p} \tag{20}$$

$$\angle FH_F O = \cos^{-1}\frac{{y_F}^2 + {y_O}^2 - FO\_Map^2}{2y_F y_O} \tag{21}$$



(a) 添加垂直平面 α　　(b) 元素映射到平面 α

图 2 推导角 θ 的过程

Fig. 2 The process of deriving the angle θ

利用强迫点更新质心距离矩阵实现过程如算法 3 所示。

算法 3 更新质心距离矩阵

输入：质心集 $\{c_1, c_2, \cdots, c_k\}$，上次迭代质心距离矩阵 *lastDistC*

输出：质心距离矩阵 *distC*

Step 1: 从质心集中随机选择一个质心 c_i 作为点 M，并将 *distC* 所有值置为-1。

Step 2: 计算点 M 到其它所有质心的距离，并将距离点 M 第 $k/2$ 近的质心作为点 N，根据公式(11)(12)计算所有质心的二维坐标集 $\{O\_2D\}$，并在 $\{O\_2D\}$ 上建立 kd 树便于查找。

Step 3: 循环 1（遍历所有质心中的每个 c_j）开始：

Step 4: 根据公式(11)(12)计算质心 c_j 的二维坐标 c_j_2D，并将 c_j_2D 做为查找点在 kd 树中查找小于 $2r_j$ 的点，得到候选质心集。令点 F 为空。

Step 5: 循环 2（遍历候选质心集中的每个 O）开始：

Step 6: 若点 F 不为空，判断公式(8)，若满足则跳过此 O，若不满足则根据公式(13)(15)(16)计算质心 c_j 三维坐标，并计算 $|c_j\_3D, O\_3d|$，若 $|c_j\_3D, O\_3d| > 2r_j$，则跳过此 O。

Step 7: 计算 $|c_j, O\_d|$，若点 F 为空，则将 O_d 作为点 F。

Step 8: 循环 2 结束。

Step 9: 循环 1 结束。

2.3 Ball k-means-G*算法流程

算法 4 完整算法流程

输入：数据集 $\{x_1, x_2, \cdots, x_n\}$，簇数 k

输出：最终聚类结果 $\{C_1, C_2, \cdots, C_k\}$

Step 1: 初始化聚类中心。

Step 2: 循环 1（聚类结果收敛时停止循环）开始：

Step 3: 遍历每个簇，若此簇不稳定，则更新质心，质心移动距离和簇半径。

Step 4: 根据算法 3 更新质心距离矩阵 distC。

Step 5: 循环 2（遍历所有质心中的每个 c_i）开始：

Step 6： 遍历所有质心中每个c_j，若$distC[i][j]$不为-1 且$distC[i][j]<2r_j$，则c_j为c_i邻居簇。

Step 7： 根据算法 2 过滤簇C_i中的数据点。

Step 8： 循环 2 结束。

Step 9： 循环 1 结束。

3 对比实验

为了验证所提出算法的有效性，在 10 个真实数据集上进行了实验比较，其中数据集 1-6 为中小维度数据集，数据集 7-10 为高维数据集。选取 k-means 算法[5]，Hamerly 算法（Ham）**错误!未找到引用源。**，Annuls k-means 算法（Ann）**错误!未找到引用源。**，k-means-G*算法[19]，Ball k-means 算法[8]这 5 种精确 k-means 算法进行比较。表 1 提供了数据集的基本信息，这些数据集都来自 UCI 机器学习存储库（https://archive.ics.uci.edu/）。算法实验环境是 Windows10 bit64，内存 16GB，处理器为 Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz，使用编程语言为 Python 3.8，代码编辑器为 PyCharm。

表 1 实验数据集
Tab.1 The experimental datasets

	Index	Datasets	Capacity	Dimensions
Real datasets	1	Iris	150	4
	2	Breast Cancer Wisconsin	569	30
	3	Wine Quality	4898	11
	4	Spambase	4601	57
	5	Purchasing Intention	12330	17
	6	Codon Usage	13028	67
	7	Darwin	174	450
	8	Asian Biblical Texts	590	8266
	9	Malware	2955	1087
	10	HAR	7351	561

3.1 时间复杂度和空间成本

设查询簇的邻居簇平均数量与活动区域数据点数分别为$M\left(1\le M\le k\right)$和$n'$。为找到各簇邻居，需计算每簇到其余簇的距离，最高花费为$O\left(k^2\right)$。Ball k-means 算法通过公式(8)跳过部分质心距离计算，而 Ball k-means-G*算法进一步引入三维距离提前判断，减少更多计算。因此，实际迭代中搜索邻居簇的时间常低于$O\left(k^2\right)$，且 Ball k-means-G*会比 Ball k-means 算法跳过更多计算。

设算法 2 中 kd 树查询平均返回$m\left(m\le M\right)$个候选质心，则计算活动区域内数据点到候选质心的距离需$O\left(mn'\right)$，计算全部数据点到所属质心的距离需$O\left(n\right)$。此外，与 Ball k-means 算法相同，需对每个簇质心到其邻居簇质心的M个距离进行快速排序，最高花费为$O\left(kM\log\left(M\right)\right)$。

综上，Ball k-means-G*每次迭代的时间复杂度为$O\left(k^2+kM\log\left(M\right)+mn'+n\right)$。随着迭代进行，更多球簇趋于稳定，$M$和$m$逐渐减小，故在后期 Ball k-means-G*算法的时间复杂度会下降。

表 2 对比展示了 Ball k-means-G*与其他算法在时间复杂度和空间成本方面的差异。

表 2 每次迭代的时间复杂度和空间成本
Tab.2 The Time Complexity and Space Cost of Each Iteration

Algorithm	Setup	1st Iteration	Worst-case	Space cost
k-means	-	$O\left(kn\right)$	$O\left(kn\right)$	$O\left(n+kd\right)$

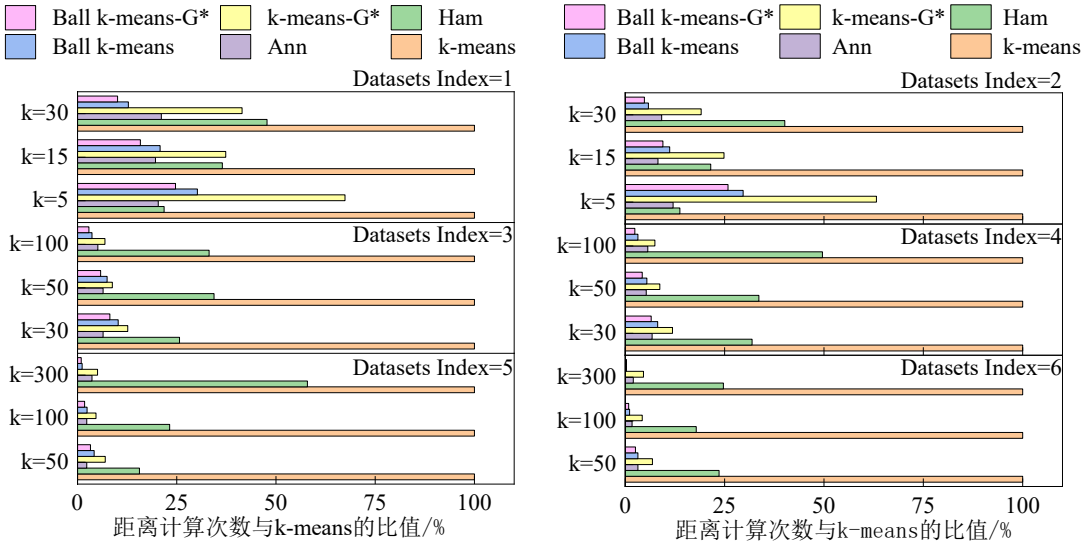
Ham	$O(n)$	$O(kn)$	$O(k^2+kn)$	$O(k+n+kd)$
Ann	$O(k+n)$	$O(kn)$	$O(k\log(k)+n\log(k)+k^2+kn)$	$O(k+n+kd)$
k-means-G*	$O(k^2)$	$O(mn)$	$O(k^2+mn)$	$O(k^2+n+kd)$
Ball k-means	$O(k^2+n)$	$O(kn)$	$O(k^2+kM\log(M)+Mn'+n)$	$O(k^2+kn+kd)$
Ball k-means-G*	$O(k^2+n)$	$O(kn)$	$O(k^2+kM\log(M)+mn'+n)$	$O(k^2+kn+kd)$

3.2 中低维度数据集实验结果

实验中，所有算法均采用相同随机种子与初始化方法。所有的实验算法皆为精确 k-means 算法，因此在相同的迭代次数后，它们会收敛至相同的聚类结果。为了分析算法在不同情况下的效果，选取了 6 个数据集（大中小规模各 2 个），每个规模的数据集均有中维和低维各一个。

图 3 中显示了在中低维度数据集的实验中，各个算法距离计算次数与 k-means 算法的比值。其中 Ball k-means-G*算法在 18 种情况下，有 13 种获得了最低的计算次数，Ann 算法在 5 种情况下表现最佳。与 Ball k-means 算法相比，Ball k-means-G*算法在所有情况下均实现了性能提升。经计算，其距离计算次数的平均减少比例高达 20.18%，这充分证明了算法改进的有效性。

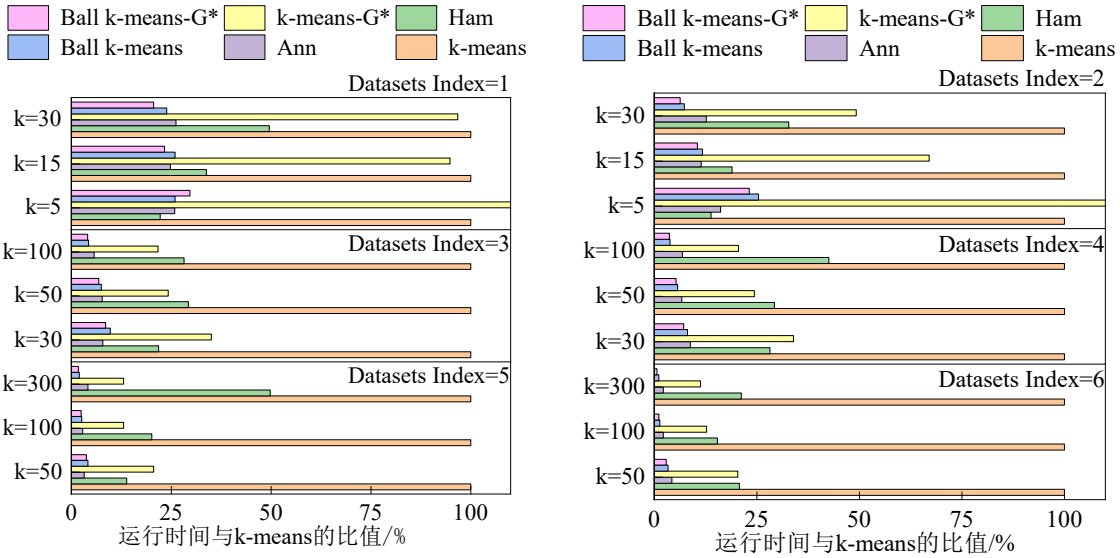
图 4 中显示了每个算法的运行时间与 k-means 算法的比值，对于 18 种情况中的 14 种，Ball k-means-G*算法都获得了更短的运行时间，Ann 算法和 Ham 算法各有 2 种情况时间更短。



(a) 低维度数据集距离计算次数　　(b) 中维度数据集距离计算次数

图 3 中低维度实验中各算法距离计算次数（与 k-means 相比）的比值

Fig.3 Ratio of Distance Calculations (to k-means) by Algorithm in Medium and Low Dimensional Experiments



(a) 低维度数据集运行时间　　(b) 中维度数据集运行时间

图 4 中低维度实验中各算法运行时间（与 k-means 相比）的比值

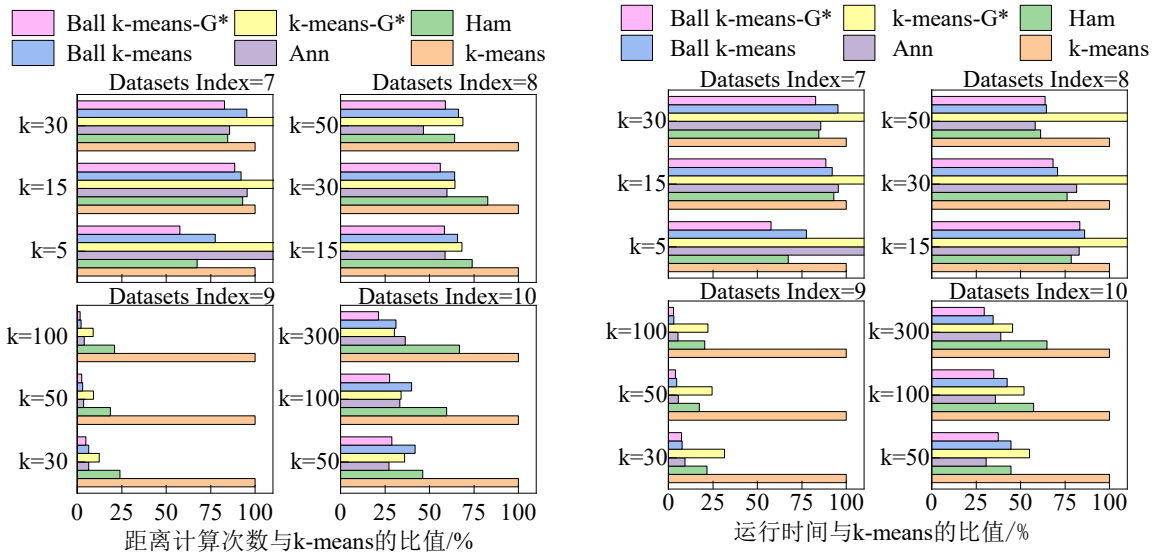
Fig.4 Ratio of Algorithm Execution Time (to k-means) in Medium and Low Dimensional Experiments

3.3 高维度数据集实验结果

为了检验算法在高维空间中的优化效果，在 4 个高维数据集上进行了测试，其中小规模数据集 2 个，中高规模数据集各 1 个。实验的前提条件与上节相同，结果表明，Ball k-means-G*成功地将其中低维度的优势延续到了高维场景。

从图 5(a)中可以看出，在 12 种情况下的 9 种，Ball k-means-G*算法获得了更低的距离计算次数，Ann 算法在另外 3 种情况下表现更佳。在所有情况下 Ball k-means-G*算法相较于 Ball k-means 算法距离计算次数的平均减少比例到约 21.14%。

从图 5(b)中可以看出，运行时间的结果与距离计算次数的趋势基本吻合。在 12 种情况的 9 种 Ball k-means-G*算法都获得了更短的运行时间，Ann 算法 2 种情况时间更短，Ham 算法 1 种情况时间更短。



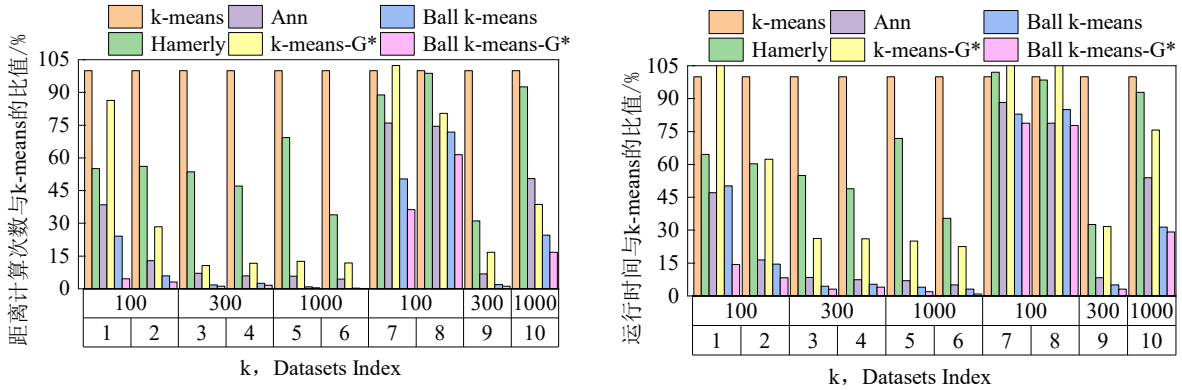
(a) 距离计算次数的比值　　(b) 运行时间的比值

图 5 高维度实验中各算法在两种指标（与 k-means 相比）的比值

Fig.5 Ratio of Two Metrics for Each Algorithm (to k-means) in High Dimensional Experiments

3.4 大 k 条件下实验结果

图 6 中可以看出，Ball k-means-G*在所有 10 个数据集上，均取得了最低的距离计算次数和最短的运行时间。且相较于 Ball k-means 算法，其在中低维度距离计算次数平均减少 48.28%，在高维度情况下平均减少 28.68%。这些结果强有力地证明，Ball k-means-G*算法进一步的优化扩大了 Ball k-means 在大 k 场景下的优势。



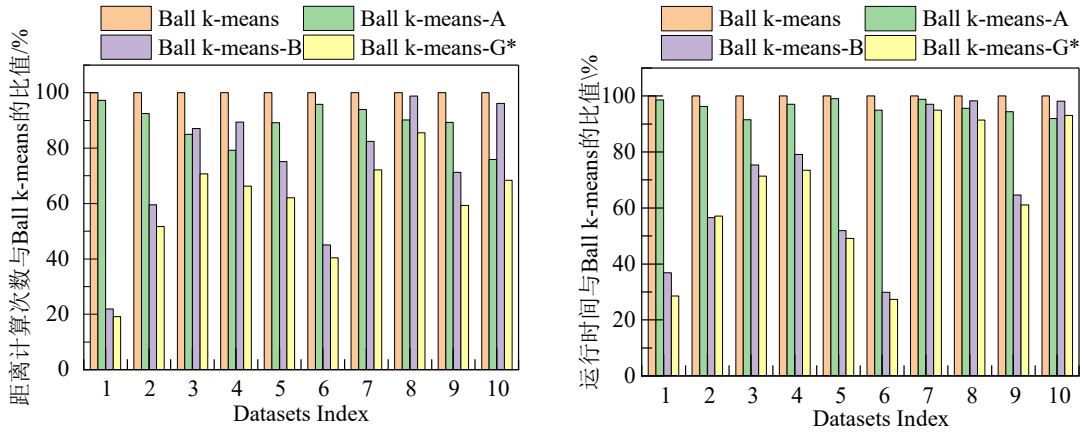
(a) 距离计算次数的比值　　(b) 运行时间的比值

图 6 大 k 实验中各算法在两种指标（与 k-means 相比）的比值

Fig.6 Ratio of Two Metrics for Each Algorithm (to k-means) in Large-k Experiments

3.5 消融实验

图 7 中对各组件进行了消融实验分析，k 值设置与上一节相同。其中 Ball k-means-A 保留了分配步骤加速，Ball k-means-B 保留了查找邻居簇加速。从图中可观察到，在大 k 情况下，Ball k-means-B 的表现是比较好的。Ball k-means-A 的提升效果弱于 Ball k-means-B，但在大数据量场景下表现更优。两者距离计算次数的平均减少比例分别为 11.15%和 27.30%，而完整模型 Ball k-means-G*在所有情况下均取得较优结果，这说明两种策略在不同维度上均能提升算法性能，且结合使用时可进一步优化整体效率。



(c) 距离计算次数的比值　　(d) 运行时间的比值

图 7 消融实验中各算法在两种指标（与 Ball k-means 相比）的比值

Fig.7 Ratio of Two Metrics for Each Algorithm (to Ball k-means) in the Ablation Experiment

4 结论

为了提升 Ball k-means 算法在高维数据集上的表现，同时保留算法在大 k 聚类任务中的固有优势，本文在 Ball k-means 算法的基础上提出了一种基于原始几何概念的 Ball k-means 算法 Ball k-means-G*。本文主要创新工作如下：

1) 在分配步骤中，引入原始几何结构将数据点和候选质心降至二维以加速寻找数据点所属簇，可

以在确定数据点所属环形后进一步减少距离计算。

2) 在查找邻居簇时，引入强迫点将质心降至三维以跳过更多非必要的距离计算，并和原 Ball k-means 算法中判断非邻居条件结合良好。

3) 实验结果表明，Ball k-means-G*算法在大部分的情况下都优于其它精确加速 k-means 算法，该算法在不同数据维度、不同聚类数量下均表现优秀，尤其适合处理大 k 值聚类任务。但是在数据量较小的情况下，聚类效率有所下降。

参考文献：

[1] 王森，邢帅杰，刘琛．密度峰值聚类算法研究综述[J]. 华东交通大学学报,2023,40(1):106-116.DOI:10.16749/j.cnki.jecjtu.20230209.006
WANG S,XING S J,LIU C. Survey of density peak clustering algorithm[J]. Journal of East China Jiaotong University,2023,40(1):106-116.

[2] Abualigah L M Q .Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering[J].Studies in Computational Intelligence, 2018.DOI:10.1007/978-3-030-10674-4.

[3] Ezugwu E S , Agbaje M B , Aljojo N ,et al.A Comparative Performance Study of Hybrid Firefly Algorithms for Automatic Data Clustering[J].IEEE Access, 2020, 8(2020):121089-121118.DOI:10.1109/ACCESS.2020.3006173.

[4] Wu X , Kumar V , Quinlan J R ,et al.Top 10 algorithms in data mining[J].Knowledge and Information Systems, 2008, 14(1):1-37.

[5] Lloyd S P .Least squares quantization in PCM[J].IEEE Trans, 1982, 28(2):129-137.DOI:10.1109/TIT.1982.1056489.

[6] 闫雪婷,张帆,杨斌,等.基于皮尔逊相关系数和信息熵的改进 K-means 聚类算法[J].电子信息对抗技术,2025,40(06):41-46.
Yan X，Zhang F，Yang B. Improved K-Means Clustering Algorithm Based on Pearson Correlation Coefficient and Information Entropy[J]. Electronic Information Warfare Technology，2025，40(06):41-46.

[7] 王紫涵. 聚类分析中 K-means 聚类算法的改进与新聚类有效性指标研究[D]. 安徽大学, 2022.
Wang Z. Research on Improved K-means Algorithm and New Cluster Validity Index in Cluster Analysis[D]. Anhui University, 2022.

[8] Xia S , Peng D , Meng D ,et al.Ball k-Means: Fast Adaptive Clustering With No Bounds[J].IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44:87-99.DOI:10.1109/TPAMI.2020.3008694.

[9] 薛雷,王天放. 基于自适应动态特征加权的 K-means 算法[J]. 吉林大学学报(理学版), 2025, 63(05): 1404-1410.
Xue L, Wang T. K-means Algorithm Based on Ada-ptive Dynamic Feature Weighting[J]. Journal of Jilin University (Science Edition), 2025, 63(05): 1404-1410.

[10] Liu Y , Li X , Sun C ,et al.An Indoor Thermal Comfort Model for Real-Time Prediction of Group Thermal Comfort Based on K-Means++ Algorithm[J]. 2024.

[11] Bachem O , Lucic M , Hamed Hassani S ,et al.Approximate K-Means ++ in Sublinear Time[J]. 2016.DOI:10.1609/aaai.v30i1.10259.

[12] 薛丁文,李建中. 基于 KD 树的 k-means 聚类算法优化[J]. 智能计算机与应用, 2021, 11(11): 194-197.
Xue D, Li J. Optimization of k-means clustering algorithm based on KD-tree[J]. Intelligent Computers and Applications, 2021, 11(11): 194-197.

[13] Sculley D .Web-scale k-means clustering[J].DBLP, 2010.DOI:10.1145/1772690.1772862.

[14] Joaquín Pérez Ortega, Ortega N N A , Ruiz-Vanoye J A ,et al.A-means: improving the cluster assignment phase of k-means for Big Data[J]. 2018(2).

[15] Yu Q , Chen K H , Chen J J .Using a Set of Triangle Inequalities to Accelerate K-means Clustering[J].

2020.DOI:10.1007/978-3-030-60936-8_23.

[16] Hamerly G .Making k-means even faster[J].DBLP, 2010.DOI:10.1137/1.9781611972801.12.

[17] Nicoletti M C , Matte M K .Comparing yinyang and fission-fusion algorithms for accelerating the k-means[J].International Journal of Knowledge and Learning, 2023.DOI:10.1504/ijkl.2023.10053502.

[18] Michelle,Knights,Bruce,et al.Extending BEAMS to incorporate correlated systematic uncertainties[J].Journal of Cosmology and Astroparticle Physics, 2013, 2013(1):367-389.DOI:10.1088/1475-7516/2013/01/039.

[19] Ismkhan H , Izadi M .K-means-G*: Accelerating k-means clustering algorithm utilizing primitive geometric concepts[J].Information Sciences: An International Journal, 2022.DOI:10.1016/j.ins.2022.11.001.

[20] 李冬.聚类分析的快速算法和确定类数的研究[D].西安电子科技大学,2024.DOI:10.27389/d.cnki.gxadu.2024.000197.
LI D. Research on Fast Algorithms and Determining the Cluster Number in Clustering Analysis[D]. Xidian University,2024.

[21] Chakraborty S , Das S .Detecting Meaningful Clusters from High-dimensional Data: A Strongly Consistent Sparse Center-based Clustering Approach.[J].IEEE transactions on pattern analysis and machine intelligence, 2020, PP.DOI:10.1109/TPAMI.2020.3047489.

第一作者：王森（1969—），男，教授，硕士生导师，研究方向为计算机算法与应用。E-mail：515613251@qq.com。

通信作者：王悦琳（2001—），女，硕士研究生，研究方向为聚类分析与数据挖掘， E-mail：2024088085410002@ecjtu.edu.cn。