基于知识诱导驱动无用中心的 K-Means 算法

王 森，刘青阳，詹小秦，陈炼

(华东交通大学理学院，江西 南昌 330013)

摘要：K-Means 是一种被广泛使用的高效无监督聚类算法。研究表明，在处理高维度或非球状结构的数据集时，K-Means 算法在确定聚类数目和选择初始质心方面存在显著局限性。为了深度探究并优化 K-Means 算法的初始质心选取机制与聚类数目确定问题，提出了一种基于知识诱导驱动无用中心的 K-Means 算法。该算法首先引入高密度知识点的检测机制，通过识别数据集中的高密度知识点，构建候选质心集合；继而基于高斯混合模型理论推导最优聚类数；随后采用无用中心筛选策略对候选质心进行优化选择，最终确定最优初始质心集合。在 UCI 标准数据集和人工合成数据集上的对比实验表明，所提出的优化算法聚类性能总体优于其余对比算法。该算法有效解决了非球状数据分布的聚类问题，在复杂数据结构场景下展现出比较优越的聚类性能。

关键词：无监督聚类；K-Means；高密度知识点；高斯混合模型；无用中心

中图分类号： TP391　　**文献标志码：**A

K-Means Algorithm based on Knowledge-Induced Driving Useless Centers

Wang Sen, Liu Qingyang, Zhan Xiaoqin, Chen Lian

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

Abstract: K-Means is a widely used and efficient unsupervised clustering algorithm. Studies have shown that when dealing with high-dimensional or non-spherical data sets, the K-Means algorithm has significant limitations in determining the number of clusters and selecting the initial centroids. In order to deeply explore and optimize the initial centroid selection mechanism and cluster number determination problem of the K-Means algorithm, A K-Means algorithm based on knowledge-induced driving useless centers is proposed. The algorithm first introduces a high-density knowledge point detection mechanism, and constructs a candidate centroid set by identifying high-density knowledge points in the data set; then the optimal number of clusters is derived based on the Gaussian mixture model theory; then the useless center screening strategy is used to optimize the candidate centroids and finally determine the optimal initial centroid set. Comparative experiments on the UCI standard dataset and artificial synthetic dataset show that the clustering performance of the proposed optimization algorithm is generally better than that of the other comparison algorithms. The algorithm effectively solves the clustering problem of non-spherical data distribution and shows relatively superior clustering performance in complex data structure scenarios.

Key words: Unsupervised clustering; K-Means; High-density knowledge points; Gaussian mixture model; Useless centers

在无监督学习领域中，传统 K-Means 算法是数据挖掘和机器学习领域处理数据的重要方法，目的是使得簇内数据点相似度高，簇间相似度低，优化目标是最小化簇内平方误差[1]。在应用层面，K-Means 算法通过发现数据分布特征的结构，广泛应用于数据挖掘、机器学习、模式识别、生物信息处理等诸多领域[2]。

收稿日期：2025−03−25
基金项目：国家自然科学基金项目(12361004)

当前聚类方法体系已形成多维度技术分支，可归纳为以下几类：划分式聚类，密度聚类，层次化聚类，网格聚类，谱聚类，概率聚类。

各类方法在扩展性，参数敏感性和复杂度等维度中呈现显著差异。领域知识驱动的知识发现则是依赖于已有的领域知识、规则、理论框架、知识库或者知识等结构化知识，通过结合现有知识和推理机制，揭示新的关联或者知识。数据知识驱动的发现，是指通过对大规模数据集的深入挖掘，依赖于数据的结构、模式以及统计特征，借助机器学习、数据科学和计算技术等方法，自动化地揭示潜在规律与知识[3]。聚类整合数据知识驱动分析形成一种互补性强、动态优化的知识生成与推理机制，以克服单一依赖路径的局限性。

K-Means 作为聚类领域中的经典算法，由 MacQueen 等首次形式化提出，持续引领划分式聚类方法研究[4]。在此基础上，Hartigan 等对 K-Means 进行了改进，通过构建损失函数，迭代划分数据，证明函数在每次迭代中保持单调递减的特性，确保算法收敛至局部最优解[5]。Selim等进一步系统分析K-Means 算法的收敛特性，通过形式化分析不同距离空间下优化轨迹，证明在欧式距离空间下，算法具备局部收敛性[6]。Comaniciu 等提出了 Mean Shift[7]聚类算法，证明 K-Means 算法是 Means Shift 算法的特例。Mean Shift 引入核函数和权重机制，基于密度梯度上升的思想，迭代逼近高密度区域，提升算法对多模态分布的建模能力。Chinrungrueng等提出了另一种 K-Means 的改进方案，采用多核学习构建超球面交集结构，有效提升处理复杂数据的能力[8]。在计算效率方面，1999 年 Pelleg 等针对 K-Means 在大规模数据集的计算复杂度问题，提出基于 KD 树的加速策略，降低算法的时间复杂度[9]。

作为划分式聚类的基准算法，K-Means 的理论发展始终围绕这三大核心：初始质心，聚类数目和迭代效率。从初始质心分析，大部分研究旨在提高收敛速度并避免局部最优解，例如 K-Means++算法，构建几何概率分布，散布初始质心，从而降低不良聚类结果的可能性。从聚类数目分析，K-Means 算法的聚类数目需要预定，目前的优化方法有 Elbow Method 方法，Silhouette Score 方法，Gap Statisti 方法等。从迭代质心策略分析，大部分研究目的是减少迭代计算量，如果不考虑收敛性，大部分迭代更新质心策略是使用指数加权移动平均策略。

虽然 K-Means 算法思想简单，易于实现，擅长处理紧凑型超球数据集。但在算法的初始化过程中，必须先指定聚类数量，同时随机选择初始中心。此外，算法的性能容易受到初始簇的选择的影响，并且对于复杂数据集，确定开始时的最佳聚类数目变得复杂[10]。从这方面深入研究优化 K-Means 算法，对高效准确地使用 K-Means 算法，具有重大意义。

文中提出一种基于知识诱导驱动无用中心的 K-Means 算法（KA-KIDUC），旨在改进 K-Means 算法初始质心选取机制与聚类数目确定问题。利用数据内在知识驱动的优化理念，分析数据集的内在结构特征[11]，提取数据的高密度知识点，作为候选质心集；通过高斯混合模型拟合数据集的整体趋势，确定聚类数目；采用无用中心策略进行筛选候选质心集，剔除冗余或对聚类效果无益的质心；随后，从剔除无用中心的候选质心集选取 K 个初始质心，并将其引入 K-Means 算法进行聚类。该方法有效提升了聚类的准确性，提高了处理非球状结构的数据集的能力。

研究 K-Means 算法时，主要关注算法初始质心的缺陷，聚类数目的确定问题。这些问题通常表现为：初始质心的随机选择可能导致算法陷入局部最优解，或陷入空簇的情况；确定聚类数目非常依赖先验知识，在处理复杂数据集时，聚类数目难以确定。因此，针对这些局限性，提出了一些改进方法，旨在优化质心初始化策略，提高聚类的准确性，提升 K-Means 算法在非球状结构数据集的聚类性能。

1 K-Means 聚类算法

1.1 K-Means 聚类算法核心思想

K-Means 算法是一种思想简单的无监督聚类方法，其核心思想是通过迭代优化，将数据集划分为 K 个

簇，最小化每个簇内的数据点到簇中心的距离平方和[11]。

数据集 $X=\{x_1,x_2,...,x_N\}$ 作为输入数据，每个数据点有 l 个特征变量，其表示为 $x_i=\{x_{i1},x_{i2},...,x_{il}\}$，聚类集合 $S=\{S_1,S_2,...,S_K\}$ 作为输出数据，其中 $S_i\cap S_j=\varnothing$，$(i\neq j)$。聚类的目标函数是最小化均值的欧式距离平方和，定义为：

$$SSEMD(C)=\sum_{i=1}^{K}SSEMD(C_i) \qquad (1)$$

式(1)中，$SSEMD(C_i)$ 表示聚类中心为 C_i 的簇偏 $SSEMD$，其定义为：

$$SSEMD(C_i)=\sum_{\forall P\in C_i}dis(P,C_i)^2 \qquad (2)$$

式(2)中，$dis(x_i,x_t)$ 表示数据点 x_i 和 x_t 之间的距离。其中使用欧式距离作为两个数据点的距离度量，距离度量定义为：

$$dis(x,y)=\sqrt{\sum_{i=1}^{m}(x_i-y_i)^2} \qquad (3)$$

依据 K-Means 的算法，其时间复杂度与数据数量 N，聚类数目 K，数据维度 d，迭代次数 T 有关。时间复杂度定义为：$O(N\cdot K\cdot T\cdot d)$。

1.2 K-Means 聚类算法流程

传统 K-Means 算法是一个非常典型的划分类聚类算法，思路简单，便于实现，具体算法流程如下：

输入：数据集 $X=\{x_1,x_2,...,x_N\}$；

Step1：聚类数目 K：根据先验知识确定聚类数目 K；

Step2：初始化质心：从数据集中随机选取 K 个数据点作为初始质心 $C=\{C_1,C_2,...,C_K\}$；

Step3：分配划分：计算数据点与聚类中心的欧式距离，将数据点划分到距离最近的簇中。

Step4：质心更新：计算簇内数据点的算术均值向量作为新聚类中心（质心）；

Step5：迭代优化：重复执行分配划分（Step3）和质心更新（Step4），直到算法收敛条件；

输出：聚类结果 $S=\{S_1,S_2,\ldots,S_K\}$,其中 K 是聚类数目，S 是数据集中所有数据聚类集合。

2 改进算法 KA-KIDUC

为了克服 K-Means 算法在初始化阶段存在的局限性，探究初始质心和聚类数目的对算法的影响，提出一种改进算法 KA-KIDUC。

2.1 基于知识点提取预初始质心

与传统的领域知识驱动不同，这里的知识点是采用非人工处理和约束规则的方法，提取数据内在特征。知识点是指在数据集中一些具有高密度的数据点，与其他数据点比较，知识点具有较高的局部相对密度，是具有较高局部相对密度的任何数据点的更大距离的数据点[12]。

假设数据集为 $X=\{x_1,x_2,...,x_N\}$，目标是将数据集其划分成 K 个簇，其中聚类中心表示为 $C=\{C_1,C_2,...,C_K\}$。现在引入平均距离提供数据的密度核，利用最近邻策略[13]计算数据点的局部密度 ρ_i。

定义 1 在数据集 $X=\{x_1,x_2,...,x_N\}$ 中的任意数据点 x_i，其局部密度 ρ_i 定义为：

$$\rho_i = \frac{dis_{average}}{\frac{1}{\varphi}\sum_{j\in S_\varphi} \tilde{dis}_{ij}} \tag{4}$$

式(4)中 S_φ 表示 φ 个最接近 x_i 的数据集合（最近邻集合），其中 φ 是为正整数的阀值，$\tilde{dis}_{ij}$ 表示数据点 x_i 和 x_j 之间的距离。$dis_{average}$ 是数据集中所有数据点的平均距离，定义为：

$$dis_{average} = \underset{i,j\in C}{average}\, \tilde{dis}_{ij} \tag{5}$$

局部密度 ρ_i 使用了在数据点 x_i 周围小区域内的局部信息 S_φ 和相应 $\frac{1}{\varphi}\sum_{j\in S_\varphi} \tilde{dis}_{ij}$，提供了比 DPC 算法中局部密度更加精准细致的处理策略[13]。$dis_{average}$ 考虑所有数据点之间的综合距离信息，基于自然最近邻的思想提出自然最近邻权重值的定义。对于 $x_i \in X$，其到较高密度点的超距离 δ_i 的特征如下。

定义 2 对于 $x_i \in X$，超距离 δ_i 是衡量数据点 x_i 到其他更高密度点的最小距离，其超距离 δ_i 定义为：

$$\delta_i = \min_{j\in I,\rho_j>\rho_i} \tilde{dis}_{ij} \tag{6}$$

若 x_i 是密度最高的数据点，其超距离为所有数据点的平均距离，则超距离 δ_i 定义为：

$$\delta_i = mean\{dis_{ij} \mid i \neq j, j \in I\} \tag{7}$$

在超距离定义公式中，ρ_j 是 x_j 的局部密度，$\tilde{dis}_{ij}$ 是数据点 x_i 和 x_j 之间的距离，I 是数据点的索引集合。

选择一个合适的 φ 值是整个算法的重要问题，为了解决这个问题，提出一个新的公式。

定义 3 对于 N 个数据，阀值 φ 定义为：

$$\varphi = \frac{\sqrt{N}}{h} \tag{8}$$

式(8)中，h 为正整数，且 $h \le \sqrt{N}$，φ 是向下取整。一个数据集中最多包含 $\sqrt{N}$ 个密集数据区域。使用极限思想，N 个数据点均匀分布在这些密集数据区域，因此每个区域有 $\sqrt{N}$ 个数据点。φ 是获得相对密度值的关键参数，φ 的值小于 N。通常参数 h 设置为 1 或者 2，所以 φ 应该是 $\sqrt{N}$ 或者 $\frac{\sqrt{N}}{2}$。一般情况下，h 可以设置为 2，但在数据集太大或者数据集中存在特别小的簇时，则需要设置更大。

定义 4 设 $x_i, x_j \in X$。若 x_j 是 x_i 最近的高密度邻居，则称 x_i 为 x_j 的直联系下级，称 x_j 为 x_i 的直联系上级，用 $x_i \to x_j$ 表示。将每个数据点直接依赖的数据点计算为 η_i，局部密度越大的数据点则具有更多的依赖数据点，η_i 的计算结果为：

$$\eta_i = \sum_{j\in C, j\neq i} \zeta(x_i, x_j) \tag{9}$$

其中 ζ 定义为：

$$\zeta(x_i, x_j) = \begin{cases} 1 & (\rho_j < \rho_i, \delta_j < th) \\ 0 & (\text{其他}) \end{cases} \tag{10}$$

公式(9)中，th 的值是区分密集区和核心区的阀值，直接影响知识点的选择质量，主要由 δ_i 和 ρ_i 共同决定。通常 th 设为 δ_i 的 0.7 到 0.8 之间的分位数。如果是噪声较多的数据集，需要过滤大部分的噪声，例如：密

集区的δ_i集中在 1.0 到 1.5 之间，噪声区的δ_i集中在 2.0 到 5.0 之间，位于密集区上限（1.5）与噪声区下限（2.0）之间，形成自然分界，确保大部分真实聚类中心被保留，同时过滤噪声，一般选取靠近两者中间值，则$th \approx 1.8$。

定义 5 数据集 X 中任意点 x_i 的相对密度 ρ'_i 是 x_i 的直接下属数量，其定义为：

$$\rho'_i = \sum_{i \in C, j \neq i} \xi(x_i, x_j) \tag{11}$$

式(11)中，$\xi\left(x_i, x_j\right)$的计算方式为：

$$\xi(x_i, x_j) = \begin{cases} 1 & (x_j \to x_i, x_j \in C_{\eta i}) \\ 0 & (else) \end{cases} \tag{12}$$

式(12)中的$C_{\eta i}$表示的是数据点的直接依赖数据点η_i所包含的最近邻的子集，最近邻参数φ一般预定 10 到 20 之间[14]。但是在某些数据集中，其范围的选择难以确定，因此，考虑到聚类中心极限分布的思想，提出公式(8)计算最近的参数φ，这有助于最优参数的设置。

整个算法过程中，计算局部密度ρ_i，超距离δ_i和相对密度ρ'_i的时间复杂度都为$O(N^2 d)$,选择高密度知识点的时间复杂度为$O(Nd)$。其中 N 是数据点数量，d 是数据维度。整个算法的时间复杂度表示为：$O(N^2 d)$。

基于知识点提取预初始质心详细算法步骤如下：

输入：数据集 $X = \{x_1, x_2, \ldots, x_N\}$，聚类中心 $C = \{C_1, C_2, \ldots, C_K\}$；

Step1：根据公式(8)计算正整数阀值φ；

Step2：根据公式(4)计算出局部密度ρ_i；

Step3：根据公式(6)计算超距离δ_i；

Step4：根据超距离δ_i，确定th值，一般th是δ_i的 0.7 至 0.8 的分位数；

Step5：根据公式(9)计算数据点的直接依赖数据点直属值η_i；

Step6：根据公式(11)计算数据点的相对密度ρ'_i；

Step7：依据局部密度和相对密度，选择高密度知识点，并存入候选质心集G；

输出：候选质心集G。

2.2 利用高斯分布获取聚类中心数量

在传统K-Means算法框架中，聚类数目的确定通常依赖于人工预设机制。常见的方法论体系主要是基于领域经验和先验知识进行显式参数指定。这些方法在低维特征空间且具备显著可分性特征的数据集上具有可行性。然而当面临复杂场景时，存在空间中类间分离度评估指标的敏感性衰减，数据稀疏性和距离度量失真问题[15]。

在基于同一数据集的聚类实验结果中，尽管输入数据具有同源性，但聚类效能呈现显著差异性。如图1所示（聚类数目为3），其聚类中心的空间分布与数据集的多模态概率密度特征形成强耦合[16]，而图2（聚类数目为1）则完全忽略了数据的子结构特性。该数据集在特征空间中的分布情况，理论上最优聚类数目应设定为3。

高斯混合模型通过极大似然估计的期望最大化算法实现聚类结构优化，但其生成的聚类中心可能因模

型假设与真实数据分布失配而与理想高斯分布存在理论偏差，特别是在数据分布呈现非线性可分或复杂多模态特征时，质心的空间定位可能偏离高斯成分的理论期望值[17]。在参数初始化策略中，聚类数目通常遵循最小复杂度原则，开始递增测试。若存在领域先验知识，可约束聚类数搜索范围，从而显著提升参数收敛速率。

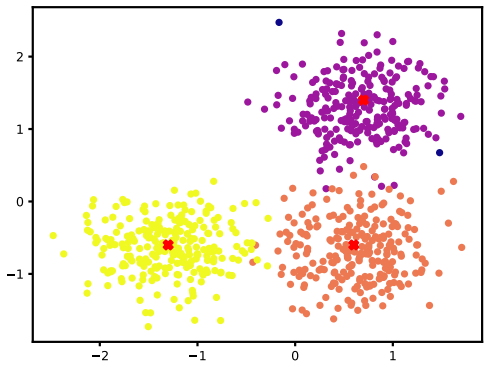


图 1 3 个聚类数量的聚类结果

Fig. 1 Clustering results for 3 cluster numbers

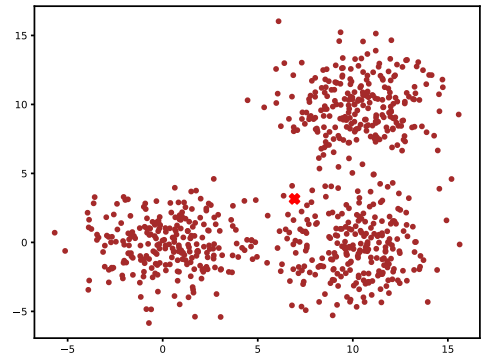


图 2　1 个聚类数量的聚类结果

Fig. 2　Clustering results for 1 cluster number

定义 6 定义 D_0 表示聚类子群的数据服从高斯分布，D_1 表示聚类中心周围的数据分布显著偏离高斯特性。当聚类子群满足 D_0 状态，无需进行分裂操作；当聚类子群满足 D_1 状态，则执行层次化分裂策略，分割成两个新簇，重新估计聚类中心[18]。

定义 7 给定一列数据点 $\{x_i\}_{i=1}^{m}$ 标准化处理，满足均值 0 和方差 1，假设 x_i 为第 i 个有序值，设 $z_i = F(x_i)$，其中 F 为标准正态分布的累计分布函数[19]，其统计函数定义为：

$$A^2(Z) = -\frac{1}{n}\sum_{i=1}^{n}(2i-1)[\ln(z_i) + \ln(1-z_{n+1-i})] - n \qquad (13)$$

定义 8 不可能从属点：假设数据点 x_i 的标签已经被分配，从数据中估计 μ 和 σ，需要根据如下定义：

$$A_*^2(Z) = A^2(Z)(1 + 4/N - 25/(n^2)) \qquad (14)$$

高斯混合模型的时间复杂度受到数据数量 N，聚类数目 K，数据维度 d，迭代次数 T 的共同影响，它的复杂度可以表示为：$O(N \cdot K \cdot T \cdot d)$。

高斯混合模型算法确定聚类数目策略如下：

输入：数据集 $X=\{x_1,x_2,\ldots,x_N\}$，聚类中心 $C=\{C_1,\ldots,C_K\}$，一般初始质心数量设为 1;

Step1：根据传统 K-Means 算法更新质心；

Step2：设置 S_i 为分配给质心 C_i 的数据点集合；

Step3：使用统计检测 S_i 是否服从高斯分布（置信水平 α =0.0001）；

Step4：初始化创建一个数组 A，将未被分配的点的索引加入数组中，如果数据服从高斯分布（满足 D_0 状态），则保留质心 C_i，否则将 C_i 替换成两个质心（满足 D_1 状态）；

Step5：重复执行 Step1 至 Step4 步骤，直至聚类中心不再变化，保存聚类数目；

输出：数据集聚类数目。

2.3 基于无用中心提取初始质心(UCI)

本小节提出基于无用中心提取初始质心的策略。通过此策略，可以有效筛选出对最终聚类结果有贡献的质心，提高算法的效率和准确性。

在传统的 K-Means 算法中，无用中心是指在迭代过程中未被分配到任何数据点的质心，会形成空簇。而文中定义的无用中心是指在初始化阶段，根据数据点与候选中心之间关系筛选出的无效候选质心，剔除无效候选质心（无用中心），减少后续聚类数据划分的计算冗余,错误分配和空簇。设 $X=\{x_1,x_2,...,x_N\}$ 是具有 N 个数据点的数据集，每个数据点有 l 个变量特征向量：$x_i=\{x_{i1},x_{i2},...,x_{il}\}$。

定义 **9** 聚类数目参数 K，输入数据集 X 和聚类集合 S，其中 $S=S_1\cup S_2\cup\ldots\cup S_K$，欧式距离平方和 $SSEDM$ 定义为：

$$SSEDM(S)=\sum_{i=1}^{K}SSEDM(S_i) \tag{15}$$

式(15)中，$SSEDM(S_i)$ 是数据集 X 中 S_i 的 $SSEDM$,定义为：

$$SSEDM(S_i)=\sum_{\forall P\in S_i}dis(x_j,core(S_i))^2 \tag{16}$$

其中，$dis(x_i,x_j)$ 表示数据点 x_i 到 x_j 的距离，其中 $(i\neq j)$，$core(S_i)$ 表示 S_i 的质心。

完全精确最小化 $SSEDM$ 并不能保证聚类解决方案 $SSEDM(S_i)$ 的最佳质量。该目标函数不仅是评估聚类结果质量的常用度量，也被作为 K-Means 的目标函数。在具有线性可分聚类的实例中，K-Means 可以有效的最小化 $SSEDM$，但是它对初始质心非常敏感[20]。

定义 10 随机选择一个数据点 $x\in X$ 作为第 i 个中心，其概率为：

$$P=\frac{MIND(X)^2}{\sum_{x\in I}MIND(X)^2} \tag{17}$$

这里的 $MIND(X)$ 表示数据点到最近聚类中心的最短距离。

定义 11 无用中心的概念：如果中心点 C 对数据点 x 没用，则满足如下定义：

$$dis(x,C_x)<dis(x,C) \tag{18}$$

$$dis(C,C_x) < dis(x,C) \tag{19}$$

定义 12 如果中心点 C 不是数据点 x 的无用中心，则它是数据点 x 的有用中心。

如图 3，在图中蓝色点表示 C_1，绿色点表示 C_2，红色点表示 C_3，黑色点是数据点 x。在图中 C_1 和 C_2 是数据点 x 的有用中心。虽然 C_3 比 C_1 更接近数据对象 x，但是 C_3 不是数据点 x 的有用中心。

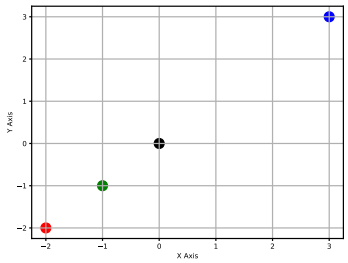


图 3　无用中心效果图

Fig. 3 Useless center rendering

选择数据集中第一个数据属性里具有最小值作为第一个中心，然后在每次迭代中，选取具有最大平均价值的数据点选为下一个中心。

定义 13 在数据集 $X=\{x_1,x_2,...,x_N\}$ 中，其中 $x_i \in X$ 的平均价值为：

$$V=\mu\frac{average(dis(x,C))}{\max\limits_{\forall C\in UNC_x}(dis(x,C))} \tag{20}$$

式(20)中，μ 是数据点距离价值系数，定义为：

$$\mu=P\times\sum\nolimits_{\forall C\in UNC_x}\ln(dis(x,C)) \tag{21}$$

式(21)，$dis(x_i,x_j)$ 是数据点 x_i 到数据点 x_j 的欧式距离，P 是数据点 x 作为第 i 个中心的概率。x 被选为下一个中心，UNC_x 是有用中心的数据点集。

这个拟初始化过程会一直持续到达到所需有用中心的数量时才会终止。在拟初始化过程中，如果最近添加的中心 C 对 x 无用，则直接忽略它；否则，将其加入 UNC_x。在这种情况下，新添有用中心时，也可能会从集合 UNC_x 删除一些数据点。

去除无用中心的时间复杂度与数据点的数量 N，中心数量 K，数据维度 d 有关，它表示为：$O(N\cdot K^2\cdot d)$。

2.4　KA-KIDUC 算法流程

算法核心是提出一种基于 K-Means 的聚类算法，提取数据内在密度知识点驱动的质心初始化框架，提供高质量的初始解。通过预筛选高密度质心，可减少后续约束融合的迭代成本，提高准确率。

考虑到高密度知识点高度依赖数据密度分布的特性，这种依赖关系很容易导致极端情况的出现。因此，通过人工添加约束条件，包括必连约束和勿连约束，来强制某些数据点进行特定分组，以此强化业务逻辑，弥补特征数据存在的不足[21]。

定义 14　数据点 $x_i,x_j \in X$，如果数据点 x_i 和 x_j 存在必连约束，那么数据点 x_i 和 x_j 的直接依赖数据点 η_i 所包含的最近邻的子集也一定存在一个簇中。如果数据点 x_i 和 x_j 存在勿连约束关系，那么数据点 x_i 和 x_j 的直接依赖数据点 η_i 所包含的最近邻的子集也一定不存在一个簇中。其中 η_i 根据公式(9)计算。

KA-KIDUC 算法具体过程如下：

输入：必连约束集合 $Must=\{(x_a,x_b),(x_k,x_t),...\}$，勿连约束集合 $Cannot=\{(x_m,x_r),(x_e,x_v),...\}$，数据集 $X=\{x_1,x_2,\ldots,x_N\}$；

Step1：对数据集 X 进行高斯分布算法，确定数据集聚类数目 K；

Step2：利用基于知识点提取预初始质心算法，计算数据集高密度知识点，构建候选质心集 G；

Step3：根据无用中心策略，在候选质心集 G 中选取 K 个初始质心 $C=\{C_1,C_2,...,C_K\}$；

Step4：更新分配：计算数据点到各个聚类中心的欧氏距离，将数据点划分到最近邻聚类簇；

Step5：对比各个簇，*Must* 集，*cannot* 集中数据点，根据定义 14，重新划分数据点；

Step6：更新质心；根据当前数据点分配情况，计算簇内数据点的算法均值向量作为新质心；

Step7：迭代更新：重复 Step4-Step6，不断重复划分数据点并更新质心，达到收敛条件；

输出：最终聚类结果 $S=\{S_1,S_2,\ldots,S_K\}$。

KA-KIDUC 时间复杂度主要分为两个部分。第一部分是确定初始质心的时间复杂度，第二部分是划分数据点部分。其中数据点的数量 N，聚类数目 K，迭代次数 T，数据维度 d，KA-KIDUC 的时间复杂度表示为：$O(N^2d+NKTd+K^2d)$。

3 实验分析

3.1 实验设计

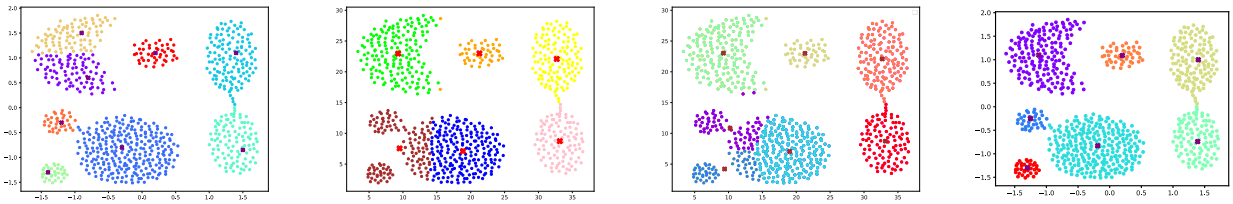
为了验证所提出聚类算法的有效性，在不同人工合成数据集和真实数据集的进行了对比实验和消融实验。在实验结果，将采用聚类算法评价指标和聚类时间：调整后的互信息（AMI），调整后的兰德系数（ARI）和 Fowlkes-Mallows 指标（FMI），轮廓系数（SC），戴维森堡丁指数（DBI），聚类时间（TIME）。

对比实验中采用传统 K-Means 算法，GMM 算法，K-Means++算法为对比算法。K-Means 算法基于距离度量的硬划分原则；K-Means++算法作为 K-Means 的改进变体，通过概率化初始化策略优化簇中心选择；GMM 算法采用概率生成框架，假设数据由若干个高斯分布线性组合生成，通过最大化对数似然函数估计参数。

消融实验主要验证改进算法在聚类数目和约束条件的情况下，聚类效果的影响结果。使用真实数据集里的 Seeds 数据集和 Banknote 数据集。

3.2 实验结果

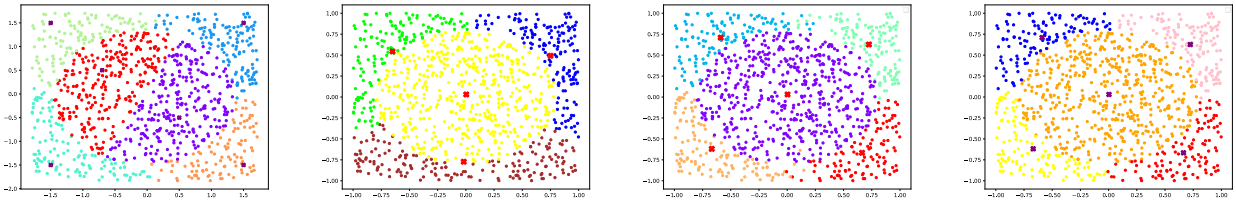
图 4 至图 9 分别可视化了人工数据集和真实数据集上聚类结果，表 1 列出了对比算法在实验中数据集指标。图 10 至图 13 是在 Seeds 和 Banknote 数据集的消融实验测试集上构建基准模型，随后通过逐步移除/替换策略（如去除约束条件等）生成对比实验组，最终通过表 2，表 3 记录的聚类性能，定量分析性能差异的显著性。



(a) GMM　(b) K-Means++　(c) K-Means　(d) KA-KIDUC

图 4　4 种算法在 Aggregation 的聚类结果对比

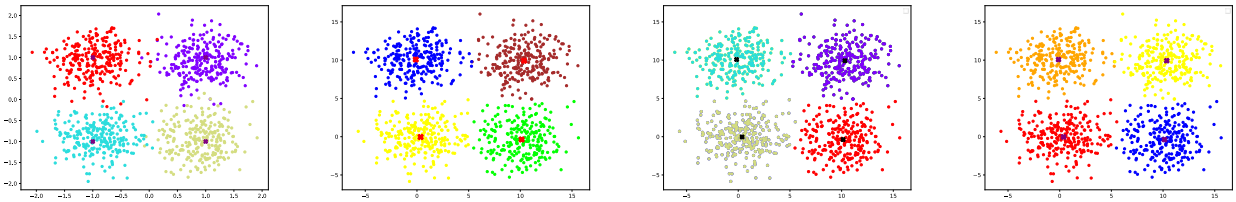
Fig. 4 The comparison of cluster results of 4 algorithms in Aggregation



(a) GMM　(b) K-Means++　(c) K-Means　(d) KA-KIDUC

图 5　4 种算法在 Circle 的聚类结果对比

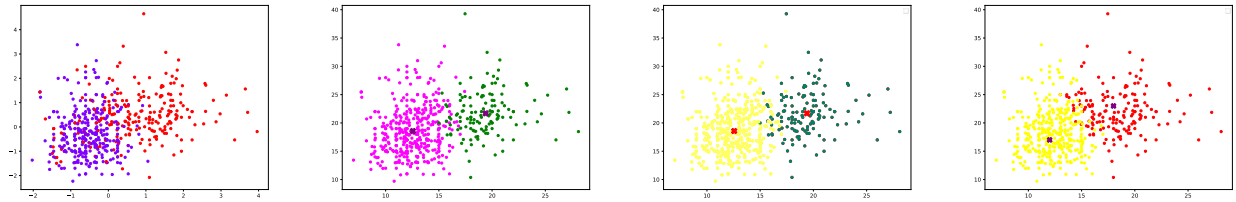
Fig. 5 The comparison of cluster results of 4 algorithms in Circle



(a) GMM　(b) K-Means++　(c) K-Means　(d) KA-KIDUC

图 6　4 种算法在 Square 的聚类结果对比

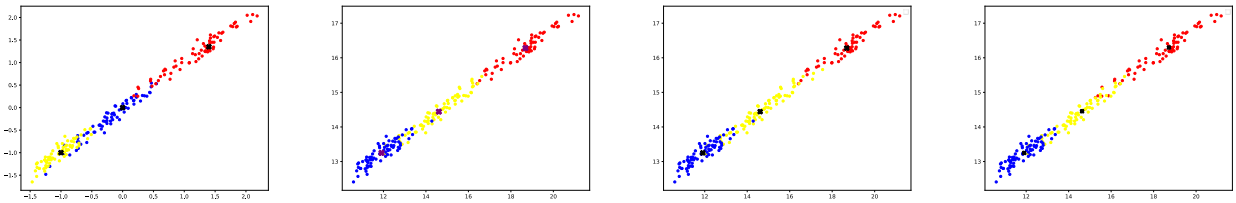
Fig. 6 The comparison of cluster results of 4 algorithms in Square



(a) GMM　(b) K-Means++　(c) K-Means　(d) KA-KIDUC

图 7　4 种算法在 WDBC 的聚类结果对比

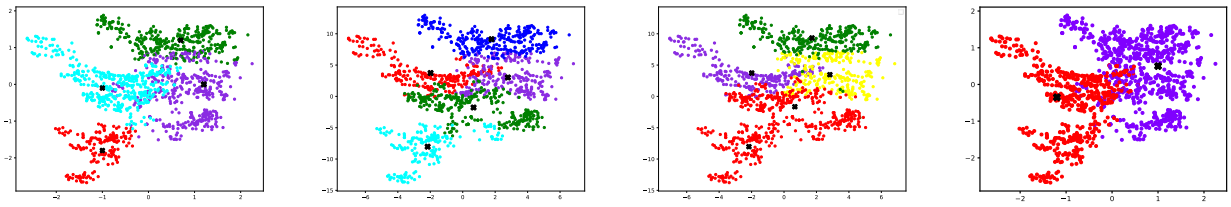
Fig. 7 The comparison of cluster results of 4 algorithms in WDBC



(a) GMM　(b) K-Means++　(c) K-Means　(d) KA-KIDUC

图 8　4 种算法在 Seeds 的聚类结果对比

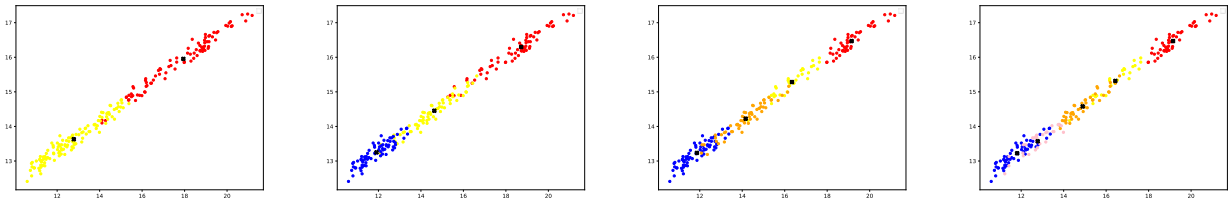
Fig. 8 The comparison of cluster results of 4 algorithms in Seeds



(a) GMM　　(b) K-Means++　　(c) K-Means　　(d) KA-KIDUC

图 9　4 种算法在 Banknote 的聚类结果对比

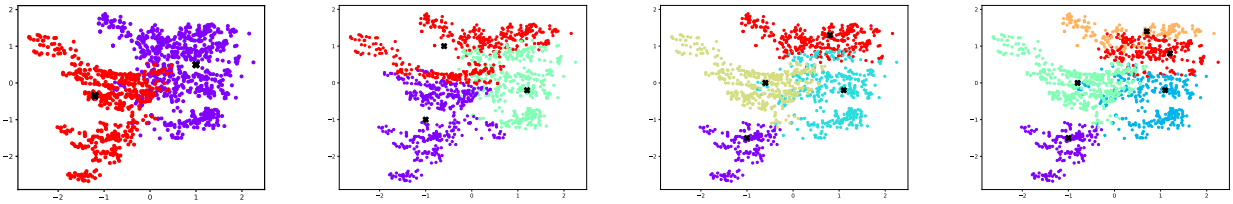
Fig. 9 The comparison of cluster results of 4 algorithms in Banknote



(a) 聚类数量为 2　　(b) 聚类数量为 3　　(c)聚类数量为 4　　(d) 聚类数量为 5

图 10　聚类数量对 Seeds 数据集的消融实验图

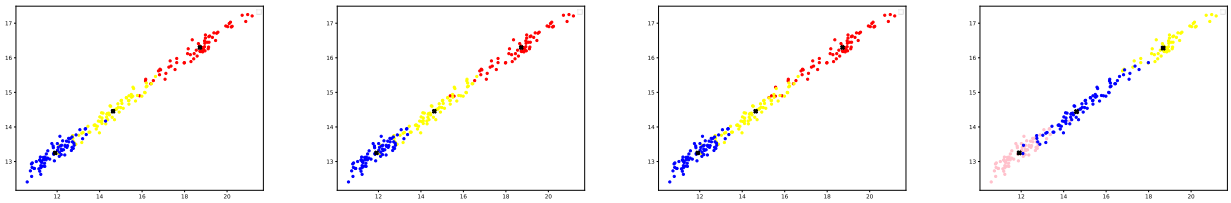
Fig. 10 Experimental diagram of the ablation of the Seeds dataset by cluster number



(a) 聚类数量为 2　　(b) 聚类数量为 3　　(c) 聚类数量为 4　　(d) 聚类数量为 5

图 11 聚类数量对 banknote 数据集的消融实验图

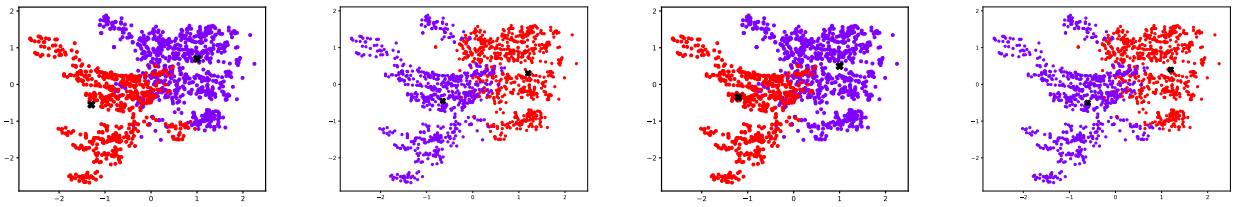
Fig. 11 Experimental diagram of the ablation of the WDBC dataset by cluster number



(a) 添加必连约束　　(b) 添加勿连约束　　(c) 添加必连和勿连约束　　(d) 无约束条件

图 12　约束条件对 Seeds 数据集的消融实验图

Fig. 12 Ablation experiment diagram of the constraints on the Seeds dataset



(a) 添加必连约束　(b) 添加勿连约束　(c) 添加必连和勿连约束　(d) 无约束条件

图 13 约束条件对 Banknote 数据集的消融实验图

Fig. 13 Ablation experiment diagram of the constraints algorithm on the Banknote dataset

表 1　4 种算法在 5 个数据集的评价指标对比

Tab. 1 The Comparison of evaluation indicators of 6 algorithms in 5 datasets

Datasets	Algorithms	AMI	ARI	FMI	SIC	DBI	TIME(s)
Aggregation	K-Means	0.8735	0.8081	0.8503	0.5025	0.7110	**0.0717**
	GMM	0.9571	0.9286	0.9447	0.5005	0.6211	0.4527
	K-Means++	0.8823	0.8124	0.8521	0.5205	0.6368	0.0817
	KA-KIDUC	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.5722**	0.3651
Circle	K-Means	0.5726	0.5622	0.7505	0.4828	**0.6329**	**0.1764**
	GMM	0.5745	0.3864	0.6218	0.4560	0.7987	0.6562
	K-Means++	0.7247	0.6948	0.8336	0.4552	0.8070	0.5682
	KA-KIDUC	**0.7790**	**0.7664**	**0.7722**	**0.8102**	0.6562	9.5668
Square	K-Means	0.9094	0.9425	0.9544	0.6028	0.5118	0.1904
	GMM	0.9331	0.9551	0.9663	0.5981	0.5109	0.6502
	K-Means++	0.9293	0.9525	0.9643	0.6002	**0.5077**	**0.1032**
	KA-KIDUC	**0.9331**	**0.9551**	**0.9663**	**0.6028**	0.5114	8.1332
WDBC	K-Means	0.4640	0.4914	0.7915	0.6973	0.5044	0.2034
	GMM	0.5439	0.6607	0.8391	0.4230	0.9114	0.8208
	K-Means++	0.4640	0.4914	0.7915	0.9673	0.5044	**0.0558**
	KA-KIDUC	**0.5662**	**0.6521**	**0.8465**	**0.6147**	**0.4065**	3.7688
Seeds	K-Means	0.7695	0.7685	0.8455	0.4838	0.7353	0.5635
	GMM	**0.8662**	**0.9020**	**0.9344**	**0.4782**	0.7296	0.8208
	K-Means++	0.7709	0.7908	0.8601	0.4314	**0.7255**	**0.2370**
	KA-KIDUC	0.8005	0.8389	0.8922	0.4708	0.7321	1.8321
Banknote	K-Means	0.7695	0.7685	0.8455	0.4838	**0.7353**	**0.5635**
	GMM	0.6734	0.5342	0.7316	0.3080	1.1497	2.0996
	K-Means++	0.3323	0.2390	0.5087	0.3128	1.0717	0.6793
	KA-KIDUC	**0.8536**	**0.9060**	**0.9538**	**0.9060**	0.9538	8.7602

表 2　聚类数量对两种数据集的性能分析表

Tab. 2 Performance analysis table of the number of clusters for 2 datasets

Datasets	聚类数量	AMI	ARI	FMI	SIC	DBI	TIME(s)
Seeds	2	0.4957	0.4446	0.6833	0.4771	0.7735	1.9716
	3	**0.8005**	**0.8389**	**0.8922**	**0.4708**	**0.7321**	**1.8321**
	4	0.7301	0.7298	0.8142	0.4327	0.8509	2.2609
	5	0.6767	0.6299	0.7429	0.3889	0.9573	2.8515
Banknote	2	**0.8536**	**0.9060**	**0.9538**	**0.9060**	**0.9538**	**8.7602**
	3	0.3719	0.3350	0.6153	0.2215	1.2910	10.8231
	4	0.6734	0.5324	0.7316	0.3080	1.1497	11.0265
	5	0.6056	0.4371	0.6633	0.2762	1.1442	11.1045

表 3　约束条件对两种数据集的性能分析表

Tab. 3 Performance analysis table of constraints for 2 datasets

Datasets	必连约束	勿连约束	AMI	ARI	FMI	SIC	DBI	TIME(s)
Seeds	1	0	0.7724	0.8024	0.8677	0.4844	0.7274	2.2757
	0	1	0.7912	0.8148	0.8761	0.4843	0.7298	2.1083
	0	0	0.7364	0.7163	0.8115	0.4712	0.7463	1.9985
	1	**1**	**0.8105**	**0.8389**	**0.8922**	**0.4708**	**0.7321**	**1.8321**
Banknote	1	0	0.7388	0.8326	0.9173	0.8326	0.9173	8.9061
	0	1	0.6807	0.7171	0.8594	0.1695	2.0280	9.3560
	0	0	0.6389	0.6637	0.8333	0.1766	1.8857	9.7221
	1	**1**	**0.8536**	**0.9060**	**0.9538**	**0.9060**	**0.9538**	**8.7602**

3.3　分析与评估

在对比实验中，Aggregation 数据集，Circle 数据集，Square 数据集是人工数据。其特性明显，干扰噪声少的特点，各个算法在这三种人工数据集上的内在性能和外在性能的能力差距不大，性能数值差距在 0.2 以内。但是可以发现 KA-KIDUC 处理数据边界明显，簇内密度均匀，簇间数据数量不均衡的数据集，具有更好的聚类效果。真实数据集 Seeds 数据集是经典的农业形态测量数据集，呈现一条斜线的分布特征，存在较多的干扰数据点。Banknote 是钞票真伪鉴别的数据集，簇间数据数量不均衡，噪声多。这两个真实数据集簇间没有明显的分界线，数据相互交叉。分析实验数据，KA-KIDUC 的聚类性能指标比其他对比算法指标数值优于 0.1 以上，可以看出 KA-KIDUC 具有更好的抗噪性，聚类效果更好。

在 KA-KIDUC 算法的消融实验中，主要使用 Banknote 和 Seeds 数据集，根据聚类性能分析当聚类数目与数据分类数目一致时，其聚类效果更好；同时添加必连约束和勿连约束，其展现的聚类效果最好。

从整个实验中分析，虽然 KA-KIDUC 算法的聚类性能整体优于其他对比算法，但所需要的聚类时间是最多的，意味着其收敛速率更低。在消融实验中发现，当聚类数目越多，收敛速度越慢。同时有必连约束和勿连约束时，算法的收敛速度会大大提高，并且聚类性能更好。

4 总结

为了降低 K-Means 算法在初始质心选择和聚类数目设定上的局限性，提出了一种基于知识诱导驱动无用中心的 K-Means 算法，主要创新工作包括：

1）引入数据内在知识驱动，识别数据集的高密度知识点。

2）利用高斯混合模型分布模型预测数据集的趋势走向，获取聚类数目。

3）采用无用中心筛选策略，剔除无用中心，筛选初始质心。

4）添加必连约束和勿连约束，有效地减少了后续点聚类过程中标签错误分配的情况。

实验结果表明，KA-KIDUC 算法在处理数据边界明显，簇内密度均匀，簇间数据数量不均衡的数据集更有优势，具有更好的抗噪能力。然而，该优化算法在收敛速度明显下降，KA-KIDUC 时间复杂度与传统的 K-Means 算法相比较，初始化阶段显著增加的时间复杂度，不适用大规模数据集。聚类算法实际应用中需要考虑质量和效率，可以将优化初始化步骤或者寻找轻量的替代方法作为未来探索方向。

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(01)：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(01): 106-116.

[2] 周晓东，董海清，张昆鹏，等. 基于几何的 K-means 初始聚类中心优化算法研究[J]. 仪表技术，2025，(02):66-69+73. DOI:10.19432/j.cnki.issn1006-2394.2025.02.008.
ZHOU X D, DONG H Q,ZHANG K P, el. Research on Optimization Algorithm for Initial Clustering Centers of K-means Based on Geometry[J]. Instrumentation technology, 2025, (02):66-69+73.

[3] 姚苏梅，陆泉. 数据与知识协同驱动的知识发现：概念、机理与模型[J]. 情报学报，2025，44(03):282-295.
Yao S, Lu Q. Knowledge-Discovery Method Driven by the Collaboration of Data and Knowledge: Concept, Mechanism, and Model[J]. Journal of Intelligence, 2025, 44(03):282-295.

[4] MacQueen J. Some methods for classification and analysis of multivariate observations[C]. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. University of California press, 1967, 5: 281-298.

[5] Hartigan J A, Wong M A. Algorithm AS 136: A k-means clustering algorithm[J]. Journal of the royal statistical society. series c (applied statistics), 1979, 28(1): 100-108.

[6] Selim S Z, Ismail M A. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality[J]. IEEE Transactions on pattern analysis and machine intelligence, 1984 (1): 81-87.

[7] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis[J]. IEEE Transactions on pattern analysis and machine intelligence, 2002, 24(5): 603-619.

[8] Chinrungrueng C, Sequin C H. Optimal adaptive k-means algorithm with dynamic adjustment of learning rate[J]. IEEE Transactions on neural networks, 1995, 6(1): 157-169.

[9] Pelleg D, Moore A. Accelerating exact k-means algorithms with geometric reasoning[C]. Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. 1999: 277-281.

[10] Ikotun A M, Ezugwu A E, Abualigah L, et al. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data[J]. Information Sciences, 2023, 622: 178-210.

[11] 孙林，刘梦含，薛占熬. 结合人工蜂群与 K-means 聚类的特征选择[J]. 计算机科学与探索，2024，18(01):93-110.
SUN L, LIU M H, XUE Z A. Feature Selection Combining Artificial Bee Colony with K - means Clustering[J]. Journal of Frontiers of Computer Science and Technology 2024, 18(01):93-110.

[12] Tang Y, Pan Z, Hu X, et al. Knowledge-induced multiple kernel fuzzy clustering[J]. IEEE Transactions on Pattern Analysis and

Machine Intelligence, 2023, 45(12): 14838-14855.

[13] 原泽菲，张正军，姜国林. 基于相对邻近度的自适应谱聚类算法[J/OL]. 计算机科学，1-12[2024-12-28].
YUAN Z F，ZHANG Z J，JIANG G L. Adaptive spectral clustering algorithm based on relative proximity[J/OL]. Computer Science，1-12[2024-12-28].

[14] 张清华，周靖鹏，代永杨，等. 基于代表点与 K 近邻的密度峰值聚类算法[J]. 软件学报，2023，34(12):5629-5648.
ZHANG Q H，ZHOU J P，D Y Y，el. Density Peak Custering Algorithm Based On Representative Points and K-nearest Neighbors[J]. Journal of Software, 2023, 34(12):5629-5648.

[15] Sarmadi H, Entezami A, Magalhães F. Unsupervised data normalization for continuous dynamic monitoring by an innovative hybrid feature weighting-selection algorithm and natural nearest neighbor searching[J]. Structural Health Monitoring, 2023, 22(6): 4005-4026.

[16] 何选森，何帆，徐丽，等. K-Means 算法最优聚类数量的确定[J]. 电子科技大学学报, 2022, 51(06):904-912.
HE X S, HE F, XU L, el. Determination of the Optimal Number of Clusters in K-Means Algorithm[J]. Journal of University of Electronic Science and Technology of China, 2022, 51(06):904-912.

[17] Yan H, Lei Z, Liu C, et al. Gmm-resnext: Combining generative and discriminative models for speaker verification[C]. ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024: 11706-11710.

[18] Patel E, Kushwaha D S. Clustering cloud workloads: K-means vs gaussian mixture model[J]. Procedia computer science, 2020, 171: 158-167.

[19] Neamvonk J, Phuenaree B. Assessment of Anderson-Darling and their Modified Tests for right skewed distribution[J]. Computer Science, 2022, 17(3): 1327-1339.

[20] Ahmed M, Seraj R, Islam S M S. The k-means algorithm: A comprehensive survey and performance evaluation[J]. Electronics, 2020, 9(8): 1295.

[21] 周晨曦，梁循，齐金山. 基于约束动态更新的半监督层次聚类算法[J]. 自动化学报, 2015, 41(07):1253-1263.
ZHOU C X, LIANG X, QI J S. A Semi-supervised Agglomerative Hierarchical Clustering Method Based on Dynamically Updating Constraints[J]. Journal of Automation, 2015, 41(07):1253-1263.

第一作者：王森（1969—），男，教授，硕士生导师，研究方向计算机算法与应用。E-mail: wangsen@ecjtu.edu.cn。

通信作者：刘青阳（2001—），女，硕士研究生，研究方向聚类分析与数据挖掘。E-mail:
2023088085410003@ecjtu.edu.cn。