doi：10.13413/j.cnki.jdxblxb.2025062

基于启发式交叉策略优化的 *K*-Means 聚类算法

张立娜[1]，张兴瑞[1]，马 丽[2]，于合龙[1]，宋欣怡[1]
（1. 吉林农业大学 信息技术学院，长春 130118；2. 无锡学院 物联网工程学院，江苏 无锡 214105）

摘要：针对传统 *K*-Means 算法对初始质心敏感、易陷入局部最优以及未能充分挖掘聚类结果潜在语义特征的问题，提出一种基于启发式交叉策略优化的 *K*-Means 聚类算法. 首先，该算法通过密度驱动的启发式交叉初始化策略，筛选高密度区域的代表性父代点，并引入交叉系数动态生成多样性初始质心，以降低随机初始化导致的聚类结果波动性；其次，在聚类迭代过程中，结合父代点信息与簇内均值更新规则，通过交叉操作动态调整质心位置，解决了传统算法因局部最优导致的簇间重叠问题；最后，将优化后的聚类结果输入多层感知机，利用其非线性映射能力挖掘潜在特征，实现了聚类结果与深层语义特征的深度融合. 实验结果表明，该算法的轮廓系数、Davies-Bouldin 指数和调整 Rand 指数分别达 0.634，1.398，0.621，显著优于其他改进算法，有效提升了算法的聚类准确性、稳定性和可解释性.

关键词：启发式交叉策略；*K*-Means 聚类算法；多层感知机；特征融合
中图分类号：TP399 **文献标志码：**A **文章编号：**1671-5489(2025)06-1663-10

K-Means Clustering Algorithm Based on Heuristic Crossover Strategy Optimization

ZHANG Lina[1], ZHANG Xingrui[1], MA Li[2], YU Helong[1], SONG Xinyi[1]
(1. *College of Information Technology, Jilin Agricultural University, Changchun* 130118, *China*;
2. *College of Internet of Things Engineering, Wuxi University, Wuxi* 214105, *Jiangsu Province, China*)

Abstract: Aiming at the problems that the traditional *K*-Means algorithm was sensitive to initial centroids, prone to local optima, and failing to fully mine the potential semantic features of clustering results, we proposed a *K*-Means clustering algorithm based on heuristic crossover strategy optimization. Firstly, the algorithm used a density-driven heuristic crossover initialization strategy to screen representative parent points in high-density regions, and introduced a crossover coefficient to dynamically generate diverse initial centroids to reduce the volatility of clustering results caused by random initialization. Secondly, during the clustering iteration process, by combining the information of parent points with the intra-cluster mean update rule, the centroid positions were dynamically adjusted through crossover operations, which solved the problem of inter-cluster overlap caused by the local optima of the traditional algorithm. Finally, the optimized clustering results were input into a multi-layer perceptron, which utilized its nonlinear mapping ability to mine potential features and achieved deep fusion of clustering results with deep semantic features. Experimental results show that

收稿日期：2025-02-26.

第一作者简介：张立娜(1982—)，女，汉族，硕士，副教授，从事大数据分析和深度学习的研究，E-mail：zhangln@jlau.edu.cn.
通信作者简介：于合龙(1974—)，男，汉族，博士，教授，从事机器学习和农业工程的研究，E-mail：yuhelong@jlau.edu.cn.

基金项目：吉林省教育科学规划课题项目(批准号：ZD21044).

the contour coefficient, Davies-Bouldin index, and adjusted Rand index of the algorithm reach 0.634, 1.398 and 0.621, respectively, which are significantly superior to other improved algorithms, effectively improving clustering accuracy, stability, and interpretability of the algorithm.

Keywords: heuristic crossover strategy; *K*-Means clustering algorithm; multi-layer perceptron; feature fusion

启发式交叉策略是一种基于问题特点指导交叉操作的方法，在遗传算法、进化算法等领域应用广泛. 该策略通过利用问题的特定知识或启发式信息，对传统的交叉操作进行改进，使算法在搜索过程中能更有效地探索解空间，从而提高找到最优解或近似最优解的概率.

K 均值(*K*-Means)算法作为一种经典的聚类算法备受关注. 传统 K-Means 算法因其简单、高效被广泛应用，但其对初始质心敏感的问题限制了聚类精度[1]. 何嘉伦等[2]针对 *K*-Means＋＋算法的初始质心选择进行了改进，在传统基于距离选择的基础上，融入了数据点密度信息，从而提升了聚类质量和算法的稳定性. 孙林等[3]进一步结合粒子群优化算法，解决了复杂数据分布下的局部最优问题. Huang 等[4]将模拟退火算法引入聚类过程，通过动态调整温度参数避免局部最优解. Xu 等[5]利用蚁群算法优化特征选择，提升了高维数据的聚类效率.

但上述算法的目标都只针对初始质心选择优化或全局搜索策略改进，而忽视了数据内在分布的多模态特性. 针对上述问题，本文提出一种基于启发式交叉策略优化的 *K*-Means 聚类(HC-KM)算法，通过对数据点的分布特征进行分析，利用启发式规则选择有代表性的数据点作为初始聚类中心，并在聚类过程中引入交叉操作，使聚类结果更符合数据的实际分布，从而有效提高聚类的准确性.

1 *K*-Means 聚类算法和多层感知机

1.1 *K*-Means 聚类算法

K-Means 聚类算法作为一种经典的无监督聚类算法，旨在将给定的数据集划分为 *K* 个不同的簇，使同一簇内的数据点有较高的相似度，而不同簇之间数据点的相似度较低[6]. 其核心思想基于“物以类聚”的原则，通过不断迭代更新簇中心点，逐步优化聚类结果，适合处理大规模数据，处理大数据集时高效，有良好的伸缩性. 算法原理如图 1 所示.

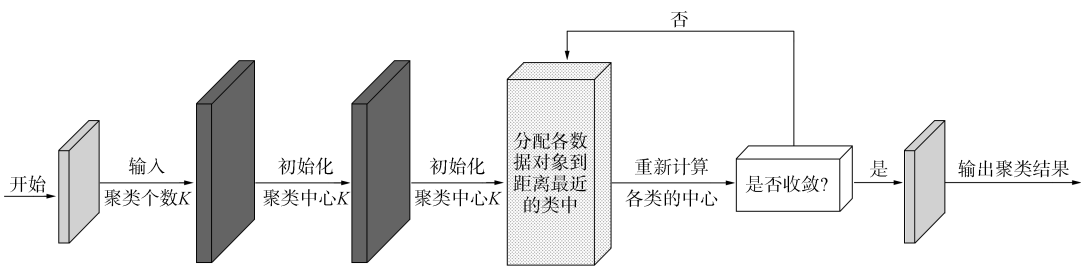

图 1 *K*-Means 聚类算法原理

Fig. 1 Principle of *K*-Means clustering algorithm

K-Means 聚类算法的目标是最小化簇内平方误差和(within-cluster sum of squares, WCSS)[7]，即每个点到其所属簇中心距离的平方和，计算公式为

$$\mathrm{WCSS}=\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in C_i}\|\boldsymbol{x}-\boldsymbol{\mu}_i\|^2, \tag{1}$$

其中 K 为簇的数量，C_i 为第 i 个簇的点集，$\boldsymbol{x}$ 为属于 C_i 的数据点，$\boldsymbol{\mu}_i$ 为第 i 个簇的质心，$\|\boldsymbol{x}-\boldsymbol{\mu}_i\|^2$ 表示数据点 $\boldsymbol{x}$ 与簇中心 $\boldsymbol{u}_i$ 之间的欧氏距离平方.

1.2 多层感知机

多层感知机(multilayer perceptron, MLP)[8]是一种经典的前馈神经网络，主要由输入层、隐藏层和输出层构成，其结构如图 2 所示.

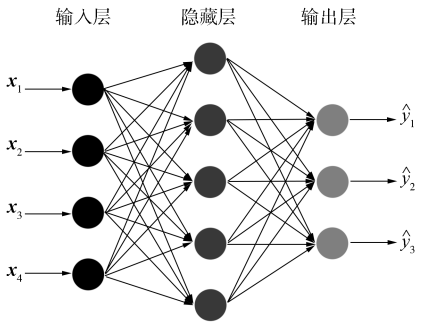

图 2　多层感知机结构

Fig. 2　Structure of multi-layer perceptron

输入层负责接收外部输入的数据，这些数据通常是经过预处理的特征向量. 隐藏层是 MLP 的核心部分，它可以包含一个或多个层次，每个隐藏层由若干个神经元组成. 神经元之间通过权重连接，且每个神经元都有一个偏置项. 隐藏层对输入数据进行特征提取和非线性变换，将原始数据映射到一个更高维的特征空间，从而挖掘数据中的潜在模式和特征. 不同隐藏层的神经元数量可根据具体任务和数据特点进行调整. 输出层则根据任务的具体类型，将隐藏层的输出映射为最终的预测结果. MLP 工作原理如下：

$$\boldsymbol{z}^{[l]}=\boldsymbol{W}^{[l]}\boldsymbol{a}^{[-1]}+\boldsymbol{b}^{[l]}, \tag{2}$$

$$\boldsymbol{a}^{[l]}=g^{[l]}(\boldsymbol{z}^{[l]}), \tag{3}$$

$$L(\boldsymbol{a}^{[L]},\boldsymbol{y})=-[\boldsymbol{y}\log(\boldsymbol{a}^{[L]})+(1-\boldsymbol{y})\log(1-\boldsymbol{a}^{[L]})], \tag{4}$$

$$\frac{\partial L}{\partial \boldsymbol{z}^{[L]}}=\frac{\partial L}{\partial \boldsymbol{a}^{[L]}}\cdot g^{[L]}(\boldsymbol{z}^{[L]}), \tag{5}$$

$$\frac{\partial L}{\partial \boldsymbol{W}^{[l]}}=\frac{\partial L}{\partial \boldsymbol{z}^{[l]}}\boldsymbol{a}^{[l-1]\mathrm{T}}, \tag{6}$$

$$\frac{\partial L}{\partial \boldsymbol{b}^{[l]}}=\frac{\partial L}{\partial \boldsymbol{z}^{[l]}}, \tag{7}$$

$$\boldsymbol{W}^{[l]}=\boldsymbol{W}^{[l]}-\eta\frac{\partial L}{\partial \boldsymbol{W}^{[l]}}, \tag{8}$$

$$\boldsymbol{b}^{[l]}=\boldsymbol{b}^{[l]}-\eta\frac{\partial L}{\partial \boldsymbol{b}^{[l]}}, \tag{9}$$

其中 $\boldsymbol{W}^{[l]}$ 和 $\boldsymbol{b}^{[l]}$ 分别为第 l 层的权重和偏置，$\boldsymbol{a}^{[l-1]}$ 为前一层的激活输出，$\boldsymbol{z}^{[l]}$ 为激活函数，L 为最后一层，$\boldsymbol{y}$ 为真实标签，η 为学习率.

1.3　HC-KM 算法

HC-KM 算法是一种从初始质心选择和迭代过程动态调整两方面优化的算法. 先通过密度计算筛选高密度区域的数据点作为父代候选，利用交叉系数 α 动态融合父代点生成多样性初始质心，然后将数据点分配至最近质心形成初始簇，并在质心更新阶段结合父代点信息动态调整质心位置，通过交叉操作平衡局部收敛与全局探索；若质心变化量未达到收敛阈值，则重复分配与调整步骤，直至质心稳定. 算法流程如图 3 所示.

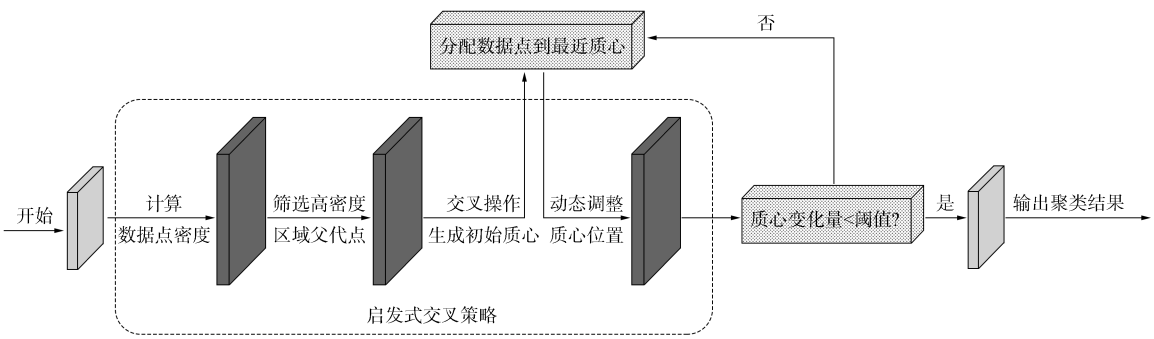

图 3　HC-KM 算法流程

Fig. 3　Flow chart of HC-KM algorithm

传统 K-Means 算法对初始质心敏感，易受随机初始化影响[9]，而 HC-KM 算法基于密度驱动与交叉操作的初始质心选择策略. 通过计算数据点周围数据点的分布情况反映数据点的密度[10]，密度越高表明该点周围数据越密集，计算数据点密度的公式为

$$\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{in}),\tag{10}$$

$$\boldsymbol{x}_j=(x_{j1},x_{j2},\cdots,x_{jn}),\tag{11}$$

$$\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2=(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\cdots+(x_{in}-x_{jn})^2=\sum_{k=1}^{n}(x_{ik}-x_{jk})^2,\tag{12}$$

$$d_{ij}=\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2,\tag{13}$$

$$D(\boldsymbol{x}_i)=\sum_{j=1}^{N}\exp\left\{\frac{d_{ij}}{t}\right\},\tag{14}$$

$$t=2\sigma^2,\tag{15}$$

其中：$D(\boldsymbol{x}_i)$表示数据点 $\boldsymbol{x}_i$ 的密度值；$\|\boldsymbol{x}_i-\boldsymbol{x}_j\|$ 表示欧氏距离；t 为控制参数，控制密度敏感度；σ 为带宽参数，决定密度计算的局部性. 密度驱动的初始质心选择策略通过核密度估计(kernel density estimation，KDE)量化数据点的局部密度分布. 采用指数核函数而非高斯核函数，因其有更陡峭的衰减特性，能更敏感地捕捉局部密集区域的差异，通过指数衰减特性赋予近邻点更高权重，使高密度区域的数据点有更大的累积贡献值. 密度值越大，表明该点所处区域数据越密集.

按密度值降序排列所有数据点，从中选取前 M 个高密度候选点代表数据分布的核心区域，计算候选点间的欧氏距离矩阵，剔除间距小于阈值的冗余点. 优先选择特征突出的数据点，最终保留 P 个父代点$\{p_1,p_2,\cdots,p_p\}$. 通过算术交叉操作融合父代点信息，生成初始质心：

$$\boldsymbol{c}_k=\alpha\cdot\boldsymbol{p}_a+(1-\alpha)\cdot\boldsymbol{p}_b,\tag{16}$$

其中 $\boldsymbol{p}_a$ 和 $\boldsymbol{p}_b$ 为随机选取的两个父代点，α 为交叉系数，交叉系数的引入类似于遗传算法中的基因重组. 重复上述操作，直至生成 K 个初始质心$\{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k\}$. 通过密度筛选与交叉策略，动态平衡权重，生成多样性初始质心，实际上在高密度区域之间构建虚拟连接，既覆盖了密集区，又引入了多样性，从而降低了聚类结果对随机初始化的敏感性.

传统 K-Means 算法迭代过程中易陷入局部最优，导致簇间重叠[11]，HC-KM 算法在此基础上进行改进. 利用生成的初始簇中心$\{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k\}$，计算数据点 $\boldsymbol{x}_i$ 到各簇中心的欧氏距离，将 $\boldsymbol{x}_i$ 分配至距离最近的簇，形成簇集合$\{S_1,S_2,\cdots,S_k\}$：

$$\boldsymbol{c}=(c_{k1},c_{k2},\cdots,c_{kn}),\tag{17}$$

$$d_m=n,\tag{18}$$

$$d(\boldsymbol{x}_i,\boldsymbol{c}_k)=\sqrt{\sum_{m=1}^{d_m}(x_{im}-c_{km})^2}.\tag{19}$$

在质心更新阶段，引入贡献因子，调整簇间距离：

$$d_{\text{agri}}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\omega\cdot d_{\text{comp}}(\boldsymbol{x}_i,\boldsymbol{x}_j)+(1-\omega)\cdot d_{\text{agri-course}}(\boldsymbol{x}_i,\boldsymbol{x}_j),\tag{20}$$

其中 ω 为权重，$d_{\text{agri-course}}$为欧氏距离. ω 的取值采用熵值法对特征重要性进行量化分析，结合层次分析法(AHP)构建判断矩阵.

基于当前簇成员重新计算质心位置，引入父代点信息优化质心位置，避免局部最优：

$$\boldsymbol{c}_k^{\text{new}}=\frac{1}{|S_k|}\sum_{\boldsymbol{x}_i\in S_k}\boldsymbol{x}_i,\tag{21}$$

$$\boldsymbol{c}_k^{\text{final}}=\beta\cdot\boldsymbol{c}_k^{\text{new}}+(1-\beta)\cdot\boldsymbol{p}_{\text{near}},\tag{22}$$

其中 $\boldsymbol{p}_{\text{near}}$为距离 $\boldsymbol{c}_k^{\text{new}}$ 最近的父代点，β 为衰减系数. 该操作将当前簇均值向父代点方向偏移，避免了迭代陷入局部极值. 其本质是通过引入历史高密度区域的先验知识，平衡局部收敛与全局探索.

若当前簇中心与上一轮簇中心的变化量小于设定阈值，则认为聚类过程收敛，停止迭代，输出聚类结果；否则，返回分配数据点步骤继续迭代，直至满足收敛条件：

$$\Delta=\sum_{k=1}^{K}\|\boldsymbol{c}_k^{\text{final}}-\boldsymbol{c}_k^{\text{old}}\|^2.\tag{23}$$

通过动态融合父代点信息，算法能跳出局部最优解，提升对多模态分布及簇重叠数据的适应性，同时增强簇间分离度.

将优化后的聚类结果输入到多层感知机[12]中，隐藏层使用激活函数 ReLU 对输入数据进行非线性变换，挖掘潜在语义特征，ReLU 函数能有效解决梯度消失问题，可极大增强模型对数据特征的学习能力：

$$\boldsymbol{x}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n), \tag{24}$$

$$\boldsymbol{y}=(y_1,y_2,\cdots,y_n), \tag{25}$$

$$y_i=\begin{cases} x_i, & x_i>0,\\ 0, & x_i\leqslant 0,\end{cases} \tag{26}$$

$$\frac{\partial y_i}{\partial x_i}=\begin{cases} 1, & x_i>0,\\ 0, & x_i\leqslant 0.\end{cases} \tag{27}$$

经过隐藏层的多次非线性变换后，先在输出层输出潜在特征预测结果，进一步挖掘聚类结果中的隐藏信息，再融合聚类结果与 MLP 输出的潜在特征预测结果.

2　实验及分析

2.1　实验设置

2.1.1　数据集

本文实验采用的数据集整合了学习通、中国计算机学会计算机软件能力认证(CCF-CSP)及力扣的多源数据[13]，构建 3 200 名用户的综合能力画像，包含 15 个特征维度，如平均绩点、算法完成度、CSP 平均分、力扣周赛排名、在线时长、资源交互频率、项目参与度及代码提交量. 数据集保留原始特征语义，未进行降维处理，以验证算法对多模态数据的特征融合能力及实际业务场景的适用性.

2.1.2　实验环境及参数

实验在配备 Intel Core i7 处理器、32 GB 内存、NVIDIA GeForce RTX 3080 GPU 的计算机上进行，操作系统为 Windows11，Python 版本为 3.8，深度学习框架为 PyTorch 1.10.0，在 PyCharm 环境下进行实验. 在 HC-KM 算法中，控制参数 $t=1.5$，交叉系数 $\alpha=0.6$. MLP 的隐藏层神经元数量分别设为 128 和 64，学习率设为 0.001，训练轮数为 100.

2.1.3　评价指标

实验选取轮廓系数(silhouette score，SS)、Davies-Bouldin 指数(Davies-Bouldin index，DBI)[14]和调整 Rand 指数(adjusted Rand index，ARI)[15]测试算法的聚类效果.

1) 轮廓系数用于衡量簇内紧密度与簇间分离度，值域为$[-1,1]$，其值越大表示聚类效果越好，是用于描述聚类后各类别轮廓清晰度的指标：

$$S(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}, \tag{28}$$

$$a(i)=\frac{1}{n-1}\sum_{j\neq i}^{n}\mathrm{distance}(i,j), \tag{29}$$

其中 j 表示与样本 i 在同一个类内的其他样本点，distance 表示 i 与 j 间的距离. $a(i)$越小说明该类越紧密. $b(i)$的计算方式与 $a(i)$类似，但需要遍历其他类簇得到多个值$\{b_1(i),b_2(i),\cdots,b_m(i)\}$从中选择最小的值作为最终结果. 所以原 $S(i)$改为

$$S(i)=\begin{cases} 1-\dfrac{a(i)}{b(i)}, & a(i)<b(i),\\ 0, & a(i)=b(i),\\ \dfrac{b(i)}{a(i)}-1, & a(i)>b(i).\end{cases} \tag{30}$$

当 $a(i)<b(i)$时，即类内的距离小于类间距离，则聚类结果更紧凑；当 $a(i)>b(I)$时，类内的距离大于类间距离，说明聚类的结果很松散.

2) Davies-Bouldin 指数表征簇间相似度与簇内离散度，其值越小表示聚类效果越优：

$$\mathrm{DBI}=\frac{1}{N}\sum_{i=1}^{N}\max\frac{\overline{S}_i+\overline{S}_j}{\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2},\tag{31}$$

其中 $\overline{S}_i$ 为第 i 类样本到其类中心的平均欧氏距离，$\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2$ 为第 i 类与第 j 类的类中心欧氏距离.

3) 调整 Rand 指数通过对比真实标签与聚类结果的一致性评估聚类准确性，值域为[−1,1]，其值越大表示与真实分布越吻合：

$$\mathrm{ARI}=\frac{2(ad-bc)}{(a+b)(b+d)+(a+c)(c+d)},\tag{32}$$

其中 a 表示在真实和实验情况下都属于同一簇的点对数目，b 表示在真实情况下属于同一簇而在实验情况下不属于同一簇的点对数目，c 表示在真实情况下不属于同一簇而在实验情况下属于同一簇的点对数目，d 表示在真实和实验情况下都不属于同一簇的点对数目.

2.2 实验结果及分析

2.2.1 对比算法

为验证本文方法的有效性，将 HC-KM 算法与以下几种具有代表性的算法进行性能比较.

1) 传统 K-Means(KM)算法：基于随机初始簇中心的经典聚类算法，通过迭代更新质心划分数据为 K 个簇，适用于基础聚类任务，但对初始值敏感，易陷入局部最优.

2) K-Means++(KM++)算法：改进 K-Means 的初始中心选择策略，通过优先选取距离较远的点作为初始质心，降低局部最优风险，提升聚类稳定性与准确性.

3) PSO-KM 算法：融合粒子群优化(PSO)的混合算法，利用 PSO 算法全局搜索能力优化初始簇中心，增强对复杂数据的适应性，显著提高聚类精度[16].

4) Mini-batch KM 算法：采用分批处理数据的 K-Means 变体，通过减少单次计算量提升大规模数据聚类效率，适用于高维或海量数据场景[17].

5) GA-KM 算法：结合遗传算法[18]的 K-Means 算法变体，通过选择、交叉、变异操作优化初始质心[19].

6) FCM 算法：模糊 C 均值聚类算法，基于隶属度矩阵实现软聚类，适应数据重叠场景[20].

7) 深度嵌入聚类(DEC)算法：利用自编码器提取特征后聚类，适合高维非线性数据[21].

2.2.2 结果分析

图 4 为各算法聚类数的肘部法则曲线. 由图 4 可见，传统 K-Means 算法因随机初始化的敏感性，在 $K=4$ 时 WCSS 下降趋缓且 SS 值仅为 0.521，聚类效果松散；KM++算法通过优化初始质心，在 $K=5$ 时 WCSS 拐点显著，SS 提升至 0.583，稳定性增强但多模态适应性不足；PSO-KM 算法利用粒子群全局搜索，在 $K=5$ 时 SS=0.602，但未充分考虑数据密度分布；Mini-batch KM 算法因分批次处理导致 WCSS 与 SS 趋势脱节，$K=5$ 时 SS 仅为 0.498，精度受限；GA-KM 算法在 $K=5$ 时 WCSS=1 050，SS=0.589，较 KM++算法提升但弱于 HC-KM 算法；FCM 算法允许软聚类，$K=5$ 时 SS 仅为 0.571，适用重叠数据但精度偏低；DEC 算法通过自编码器在 $K=5$ 时 SS=0.608，接近 HC-KM 算法，但训练效率低；HC-KM 算法在 $K=5$ 时 WCSS 与 SS 双指标均最优，分别为 980 和 0.634，其密度驱动的初始质心选择与动态交叉策略显著提升了多模态数据适应性，WCSS 较 DEC 算法降低 2.0%，SS 提升 4.3%，验证了其在复杂分布场景下的优越性，是兼顾效率与精度的最佳选择.

为证明本文算法的有效性，基于上述肘部法则对最优聚类数的分析，在 $K=5$ 时对比 HC-KM 算法与其他算法的性能差异，实验结果列于表 1.

表 1 不同算法的实验结果

Table 1 Experimental results of different algorithms

评价指标	算法							
	KM	KM++	PSO-KM	Mini-batch KM	GA-KM	FCM	DEC	HC-KM
SS	0.521	0.583	0.602	0.498	0.589	0.571	0.608	0.634
DBI	1.873	1.642	1.521	1.942	1.605	1.673	1.554	1.398
ARI	0.456	0.528	0.567	0.432	0.541	0.503	0.582	0.621

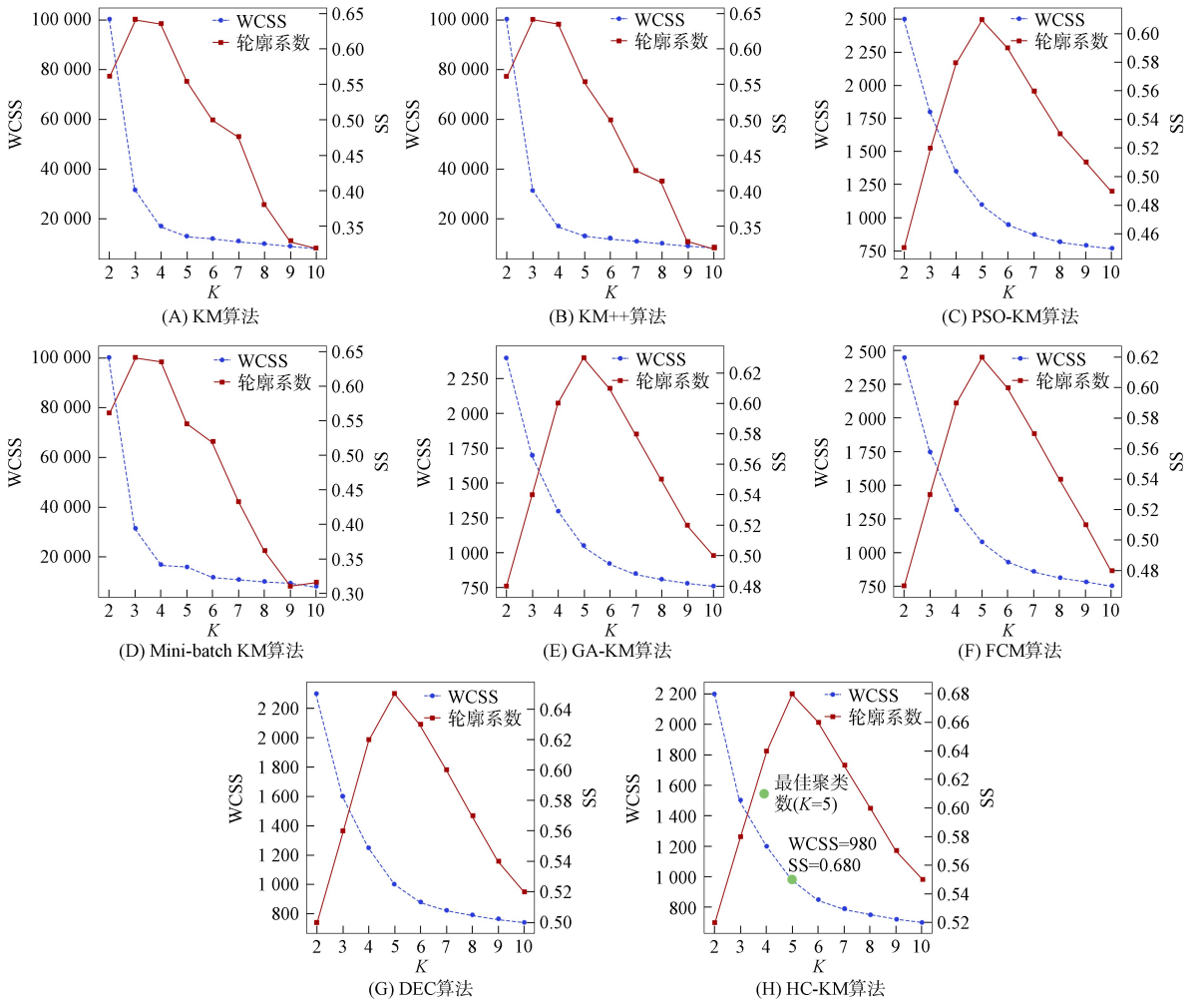

图 4　各算法聚类数的肘部法则曲线

Fig. 4　Elbow rule curves of clustering numbers for each algorithm

由表 1 可见，HC-KM 算法的 SS 值为 0.634，明显高于另外 7 种对比算法. SS 值越大，表明簇内数据点越紧密，簇间分离度越高. HC-KM 算法通过启发式交叉策略优化初始质心选择，并结合动态调整机制，使聚类结果更贴合数据真实分布，从而有效提升了簇结构的紧密度和分离度. HC-KM 算法的 DBI 值为 1.398，显著低于其他 7 种对比算法. DBI 值越小，说明簇内离散度越低且簇间相似度越低. HC-KM 算法在减少簇内距离的同时，增强了簇间差异性，从而验证了启发式交叉策略在避免局部最优和提升聚类稳定性方面的优势. HC-KM 算法的 ARI 值达 0.621，远高于对比模型. ARI 值越高，说明聚类结果与真实标签的一致性越强. HC-KM 算法通过融合密度分析与交叉操作，能更精准捕捉学生数据的潜在特征分布，从而显著提高聚类结果的可解释性和实用性.

2.2.3　消融实验

消融实验的目的是验证本文方法的有效性和创新性. 在相同数据集下通过改变启发式交叉策略，对实验结果进行对比验证，实验结果列于表 2.

表 2　改变启发式交叉策略后的消融实验结果

Table 2　Ablation experiment results after changing heuristic crossover strategy

模型	SS	DBI	ARI
完整	0.634	1.398	0.621
无交叉	0.593	1.578	0.563
随机交叉	0.605	1.512	0.587

由表 2 可见：移除启发式交叉策略后(无交叉模型)，算法各性能均显著下降，SS 从 0.634 降至

0.593，DBI从1.398升至1.578，ARI从0.621降至0.563，表明交叉策略对模型性能具有关键作用；初始质心仅依赖密度选择，未引入交叉操作生成多样性中心点，导致初始质心代表性不足，聚类过程中易陷入局部最优，簇间重叠度增加，SS和ARI下降，DBI升高；以随机系数(而非启发式规则)生成初始质心时，SS=0.605，DBI=1.512，ARI=0.587，虽略优于无交叉模型，但仍显著弱于完整模型，随机交叉缺乏对数据分布规律的针对性，难以平衡簇内紧密度和簇间分离度.

2.2.4 参数对模型的影响

在HC-KM算法中，参数的选择直接决定了模型的初始化策略和迭代过程中的动态调整能力，从而对聚类结果的稳定性和精度产生显著影响.

控制参数t通过调节密度计算的敏感性，决定了初始质心对数据局部分布的捕捉能力；交叉系数α平衡了父代点间的信息继承与多样性探索，影响质心生成的全局代表性. 本文取控制参数t分别为1.0,1.5,2.0，交叉系数α分别为0.3,0.6,0.9作为研究对象，实验结果列于表3. 当$t=1.5$时不同α值对HC-KM算法性能的影响如图5所示. 当$\alpha=0.6$时不同t值对HC-KM算法性能的影响如图6所示. 由图5可见，当$t=1.5$时，SS和ARI达到峰值，DBI最低. t过小会导致密度计算过于局部化，而t过大则忽略了局部特征，降低聚类精度. 由图6可见，当$\alpha=0.6$时，交叉操作平衡了父代点的继承与多样性探索. α过低或过高均会导致初始质心分布不均衡，影响聚类稳定性. 而本文考虑二者之间的平衡，得到了最优参数组合.

表3 不同参数t和α对HC-KM算法性能的影响

Table3 Effects of different parameters t and α on performance of HC-KM algorithm

评价	$t=1.5$					$\alpha=0.6$				
指标	$\alpha=0.3$	$\alpha=0.45$	$\alpha=0.6$	$\alpha=0.75$	$\alpha=0.9$	$t=1.0$	$t=1.25$	$t=1.5$	$t=1.75$	$t=2.0$
SS	0.55	0.62	0.634	0.61	0.59	0.61	0.63	0.634	0.62	0.58
DBI	1.65	1.45	1.398	1.50	1.61	1.52	1.42	1.398	1.45	1.65
ARI	0.54	0.59	0.621	0.57	0.56	0.58	0.61	0.621	0.58	0.53

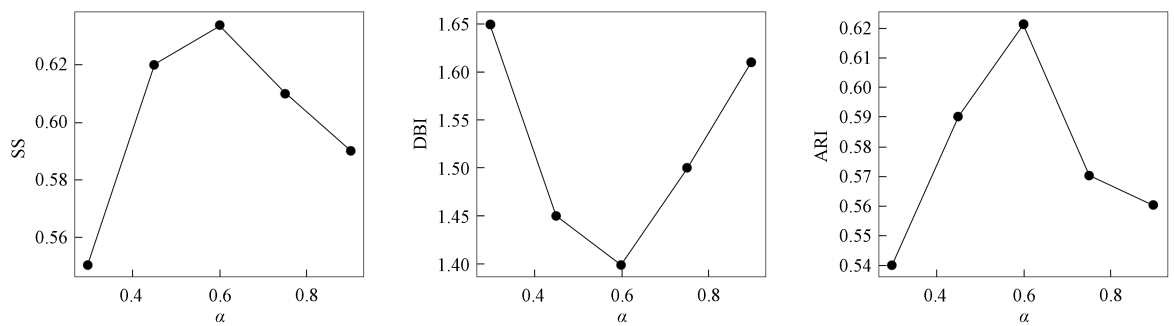

图5 当$t=1.5$时不同α值对HC-KM算法性能的影响

Fig. 5 Effects of different α values on performance of HC-KM algorithm when $t=1.5$

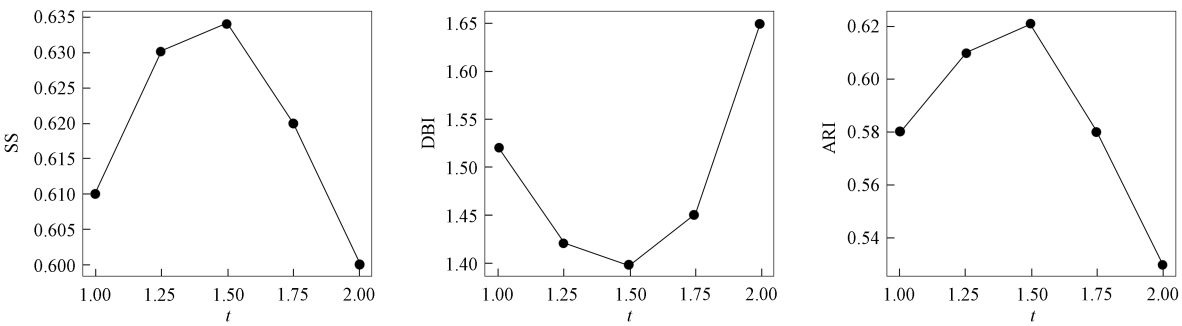

图6 当$\alpha=0.6$时不同t值对HC-KM算法性能的影响

Fig. 6 Effects of different t values on performance of HC-KM algorithm when $\alpha=0.6$

为验证多层感知机中隐藏层神经元数量与学习率设置的合理性，分别调整隐藏层神经元数量为[32,16],[64,32],[128,64],[256,128]，学习率为 0.01,0.001,0.000 1，实验结果列于表 4.

增加隐藏层神经元数量能提升聚类精度，但训练时间会随之大幅度延长；而较小的隐藏层虽训练速度更快，但聚类指标显著下降，表明模型复杂度不足会导致特征提取能力受限. 本文算法隐藏层神经元数量设置为[128,64]，在 SS=0.634，DBI=1.398，ARI=0.621 的优异性能下，每轮仅需 45 s 的训练时间，即能较好地平衡性能与效率. 学习率设置需兼顾稳定性与收敛速度，过高会导致评价指标波动，而过低虽延长了训练时间但却未显著提升精度. 当学习率为 0.001 时，模型在各结构下性能均稳定. 实验结果表明，隐藏层神经元数量设置为[128,64]与学习率 0.001 的组合能在保证聚类精度的同时控制训练成本.

表 4　不同隐藏层结构及学习率对 HC-KM 算法性能的影响

Table 4　Effects of different hidden layer structures and learning rates on performance of HC-KM algorithm

隐藏层结构	学习率	SS	DBI	ARI	每轮训练时间/s
[32,16]	0.01	0.548	1.725	0.502	18
[32,16]	0.001	0.572	1.653	0.536	22
[32,16]	0.000 1	0.563	1.687	0.524	26
[64,32]	0.01	0.592	1.583	0.567	28
[64,32]	0.001	0.613	1.512	0.588	35
[64,32]	0.000 1	0.598	1.547	0.574	38
[128,64]	0.01	0.621	1.432	0.603	42
[128,64]	0.001	0.634	1.398	0.621	45
[128,64]	0.000 1	0.605	1.489	0.591	49
[256,128]	0.01	0.627	1.415	0.612	68
[256,128]	0.001	0.631	1.402	0.618	75
[256,128]	0.000 1	0.623	1.438	0.607	80

综上所述，针对传统 K-Means 聚类算法对初始质心敏感的问题，本文提出了一种 HC-KM 算法. 该算法将启发式交叉策略应用到 K-Means 聚类中，能动态调整聚类中心，使聚类结果更符合数据的实际分布，有效提高了聚类的准确性和稳定性. 对比实验结果表明，HC-KM 算法在 SS,DBI 和 ARI 多个评价指标上都获得了较大的性能提升，在 SS 指标上获得了最多 27.31%的提升，在 DBI 指标上优化最多 28.01%，在 ARI 指标上提升最多 43.75%. 消融实验进一步验证了 HC-KM 算法的有效性，移除启发式交叉策略或采用随机交叉生成初始质心时，SS 指标最多下降 6.46%，DBI 指标最多下降 12.87%，ARI 指标最多下降 9.33%，性能弱于完整的 HC-KM 算法，实验结果表明本文模型优于其他对比模型.

参　考　文　献

[1] 袁逸铭，刘宏志，李海生. 基于密度峰值的改进 K-Means 文本聚类算法及其并行化 [J]. 武汉大学学报(理学版)，2019，65(5)：457-464.（YUAN Y M，LIU H Z，LI H S. An Improved K-Means Text Clustering Algorithm Based on Density Peaks and Its Parallelization [J]. Journal of Wuhan University (Science Edition)，2019，65(5)：457-464.）

[2] 何嘉伦，马冲. 基于初始质心的 K-Means 算法优化 [J]. 长江信息通信，2023，36(6)：69-72.（HE J L，MA C. Optimization of K-Means Algorithm Based on Initial Centroids [J]. Yangtze Information and Communication，2023，36(6)：69-72.）

[3] 孙林，张一曼，张辰珂，等. 基于改进粒子群和 K-means 聚类的优化算法 [J]. 江苏科技大学学报(自然科学版)，2023，37(3)：81-90.（SUN L，ZHANG Y M，ZHANG C K，et al. Optimization Algorithm Based on Improved Particle Swarm and K-means Clustering [J]. Journal of Jiangsu University of Science and Technology (Natural Science Edition)，2023，37(3)：81-90.）

[4] HUANG S H，WEI J J. Student Performance Prediction in Mathematics Course Based on the Random Forest and Simulated Annealing [J]. Scientific Programming，2022，2022(1)：9340434-1-9340434-9.

[5] XU H, KIM M. Combination Prediction Method of Students' Performance Based on Ant Colony Algorithm [J]. Plos One, 2024, 19(3): e0300010-1-e0300010-18.

[6] 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.

[7] ZHAO D D, HU X Y, XIONG S W, et al. *K*-Means Clustering and KNN Classification Based on Negative Databases [J]. Applied Soft Computing, 2021, 110: 107732-1-107732-15.

[8] LIN R Y, ZHOU Z R, YOU S Y, et al. Geometrical Interpretation and Design of Multilayer Perceptrons [J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 35(2): 2545-2559.

[9] 王艳娥, 安健, 梁艳, 等. 基于密度优化初始聚类中心的 *K*-Means 算法 [J]. 计算机技术与发展, 2020, 30(12): 99-105. (WANG Y E, AN J, LIANG Y, et al. *K*-Means Algorithm Based on Density Optimization of Initial Clustering Center [J]. Computer Technology and Development, 2020, 30(12): 99-105.)

[10] SUN L, QIN X Y, DING W P, et al. Nearest Neighbors-Based Adaptive Density Peaks Clustering with Optimized Allocation Strategy [J]. Neurocomputing, 2022, 473: 159-181.

[11] 伍家耀, 周鲲, 徐志强, 等. 粒子群优化算法在电力工程三维数据聚类分析中的应用 [J]. 电子设计工程, 2021, 29(6): 24-28. (WU J Y, ZHOU K, XU Z Q, et al. Application of Particle Swarm Optimization Algorithm in 3D Data Clustering Analysis of Electric Power Engineering [J]. Electronic Design Engineering, 2021, 29(6): 24-28.)

[12] 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.

[13] 王福德, 宋海龙, 孙小海, 等. 多源异构教育大数据挖掘与应用平台 [J]. 吉林大学学报(信息科学版), 2023, 41(5): 922-929. (WANG F D, SONG H L, SUN X H, et al. Multi-source Heterogeneous Educational Big Data Mining and Application Platform [J]. Journal of Jilin University (Information Science Edition), 2023, 41(5): 922-929.)

[14] IDRUS A. Distance Analysis Measuring for Clustering Using *K*-Means and Davies Bouldin Index Algorithm [J]. TEM Journal, 2022, 11(4): 1871-1876.

[15] SUNDQVIST M, CHIQUET J, RIGAILL G. Adjusting the Adjusted Rand Index: A Multinomial Story [J]. Computational Statistics, 2023, 38(1): 327-347.

[16] WANG D S, TAN D P, LIU L. Particle Swarm Optimization Algorithm: An Overview [J]. Soft Computing, 2018, 22(2): 387-408.

[17] ZHU X Y, SUN J, HE Z H, et al. Staleness-Reduction Mini-batch *K*-Means [J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 35(10): 14424-14436.

[18] SAHEED Y K, AROWOLO M O, TOSHO A U. An Efficient Hybridization of *K*-Means and Genetic Algorithm Based on Support Vector Machine for Cyber Intrusion Detection System [J]. International Journal on Electrical Engineering and Informatics, 2022, 14(2): 426-442.

[19] KOUSER K, PRIYAM A, GUPTA M, et al. Genetic Algorithm-Based Optimization of Clustering Algorithms for the Healthy Aging Dataset [J]. Applied Sciences, 2024, 14(13): 5530-1-5530-16.

[20] KUO R J, HSU C C, NGUYEN T P Q, et al. Hybrid Multi-objective Metaheuristic and Possibilistic Intuitionistic Fuzzy *C*-Means Algorithms for Cluster Analysis [J]. Soft Computing, 2024, 28: 991-1008.

[21] WEI Y, GUO H Y, GE Z R, et al. Graph Attention-Based Deep Embedded Clustering for Speaker Diarization [J]. Speech Communication, 2023, 155: 102991-1-102991-10.

(责任编辑: 韩 啸)