DOI:10.19322/j.cnki.issn.1006-4710.2022.03.012

基于改进 HHO 与 K-Medoids 的混合聚类算法

李 姣，王秋萍，戴 芳

(西安理工大学 理学院，陕西 西安 710054)

摘要：针对 K-Means 在聚类过程中对离群点敏感以及容易陷入局部最优的不足，本文提出一种基于改进 HHO(IHHO)与 K-Medoids 的混合聚类算法(IHHO-KMedoids)。在 IHHO 中，带有 Logistic混沌扰动的控制参数策略更好地实现了探索与开发之间的平衡，集成变异策略提高了算法的全局搜索能力，翻筋斗觅食策略增强了种群多样性，避免算法陷入局部最优。将所提 IHHO 与 5 种其他群智能算法和 4 种改进的 HHO 算法在 CEC 2014 测试函数上进行对比，实验结果表明 IHHO算法的优化效果较好，求解精度较高。K-Medoids 与 K-Means 相比对噪声点和离群点更鲁棒。IHHO-KMedoids 算法稳定性好，不易陷入局部最优。UCI 数据集和文本数据集上的仿真结果表明 IHHO-KMedoids 算法效率高，聚类精度高。

关键词：Harris 鹰优化算法；Logistic 映射；集成变异策略；翻筋斗觅食策略；K-Medoids 算法

中图分类号：TP301.6 **文献标志码**：A **文章编号**：1006-4710(2022)03-0410-11

Hybrid clustering algorithm based on improved HHO and K-Medoids

LI Jiao, WANG Qiuping, DAI Fang

(Faculty of Sciences, Xi'an University of Technology, Xi'an 710054, China)

Abstract: A hybrid clustering algorithm (IHHO-KMedoids) based on the improved HHO (IHHO) and K-Medoids is proposed in this paper for solving the issues of K-Means which is sensitive to outliers and easy to fall into local optimum. In IHHO, the control parameter strategy with Logistic chaotic disturbance better achieves the balance between exploration and exploitation, the ensemble mutation strategy improves the global search ability of the algorithm, and the somersault foraging strategy increases the diversity of the population and avoids the algorithm falling into local optimum. The proposed IHHO is compared with five other swarm intelligence algorithms and four improved HHO algorithms on the CEC 2014 benchmark functions with the experimental results showing that IHHO has better optimization ability and higher accuracy. K-Medoids is better robust against noise and outliers compared with K-Means. IHHO-KMedoids algorithm has high stability and is not easy to fall into local optimum. The simulation results on UCI datasets and a text dataset show that the IHHO-KMedoids algorithm has higher efficiency and clustering accuracy compared with contrastive algorithms.

Key words: Harris hawks optimization algorithm; Logistic map; ensemble mutation strategy; somersault foraging; K-Medoids

聚类分析是一种常用的无监督学习过程，用于探索或发现数据集中隐藏的模式，被广泛应用于统计学、模式识别、人工智能、图像处理等领域，其目的在于将给定数据集划分为多个簇，其中相同簇中对象的相似性较高，而不同簇中对象的相似性较低。

K-Means 是一种经典的划分式聚类算法，已经在诸多领域得到了广泛应用，但是该算法存在对离群点的依赖性较强和容易陷入局部最优的缺点。

收稿日期：2021-10-17； **网络出版日期**：2022-04-01

网络出版地址：https://kns.cnki.net/kcms/detail/61.1294.N.20220331.0922.002.html

基金项目：国家自然科学基金资助项目(61976176)

第一作者：李姣，女，硕士生，研究方向为智能计算。E-mail:1803987422@qq.com

通信作者：王秋萍，女，博士，教授，研究方向为智能算法理论及应用、决策分析、灰色系统理论及其应用。E-mail:wqp566@sina.com

K-Medoids 也是一种基于划分的聚类算法,它是在 K-Means 的基础上提出来的,其思想简单、容易实现。不同之处在于 K-Means 在迭代过程中采用簇中对象的均值作为簇的中心,而 K-Medoids 采用每个簇中与其他对象最为相似的实际对象作为聚类中心,可以改善 K-Means 对噪声点和异常点敏感的不足,但 K-Medoids 也存在可能会陷入局部最优的问题[1]。因此,近年来研究者们将一些元启发式算法与 K-Medoids 进行融合来解决上述问题。文献[2]在蚁群聚类优化的基础上结合 K-Medoids 对原聚类算法进行扩展,提出一种蚁群 K-Medoids 融合的聚类算法以提高聚类效率和鲁棒性。文献[3]将粒子群算法(PSO)与 K-Medoids 算法进行结合,利用 K-Medoids 为 PSO 过程提供适应度度量,使得火灾探测系统快速,易于实现。文献[4]提出了 K-Medoids 与元启发算法结合的方法,很好地解决了 p-移动枢纽位置分配问题。

Harris 鹰优化(Harris hawks optimizaton, HHO)算法是 Heidari 等[5]从 Harris 鹰追捕猎物时的协作行为中受到启发,于 2019 年提出的一种群智能优化算法,其特点为原理简单、优化性能较强。目前 HHO 已被成功应用于特征选择、图像处理、预测和参数估计等领域。然而,与其他群智能算法一样,HHO 在求解复杂实际问题时可能陷入局部最优、寻优精度低等问题。

因此,本文对 HHO 作出以下改进。

1) 采用带有 Logistic 混沌扰动的控制参数策略有利于更好地平衡算法的探索和开发能力。

2) 探索阶段每次位置更新后引入集成变异策略来增强算法的全局搜索能力,从而进一步提高算法的寻优能力。

3) 引入翻筋斗觅食策略,可以使每个个体移动到当前位置和对称于猎物位置之间区域的任何位置,能有效增强种群的多样性,降低算法陷入局部最优的概率。

CEC2014 测试函数的实验结果表明,改进算法的寻优性能优于其对比算法。

为解决 K-Medoids 容易陷入局部最优的问题,本文将改进的 Harris 鹰优化算法 IHHO 与 K-Medoids 相结合用于求解聚类问题,提出一种基于 IHHO 和 K-Medoids 的混合聚类算法 IHHO-KMedoids,该算法利用 IHHO 较强的全局搜索能力和较好的寻优能力克服了 K-Medoids 的不足,利用 K-Medoids 改善了 K-Means 在聚类过程中对噪声点敏感的缺点。所提的混合聚类算法具有 IHHO 和 K-Medoids 各自的优势,且具有较强的鲁棒性,能够快速有效地找到一组最佳聚类中心。在 UCI 数据集和清华大学中文文本分类数据集上分别进行了仿真实验,并与 K-Means、K-Medoids 和 HHO-KMedoids 进行对比,实验结果表明所提聚类算法的聚类性能较好,能够有效解决聚类问题。

1 改进的 Harris 鹰优化算法

1.1 Harris 鹰优化算法

HHO 中,每一个 Harris 鹰代表优化问题的一个候选解,而猎物代表每一次迭代的最好解。算法模拟了 Harris 鹰捕获猎物时的动态过程,可分为探索,探索到开发的转换,开发阶段。

1) 探索阶段。Harris 鹰随机栖息在一些位置,根据以下两种策略围捕猎物:

$$X(t+1)=\begin{cases}X_{\text{rand}}(t)-r_1\left|X_{\text{rand}}(t)-2r_2X(t)\right|, & q\geqslant 0.5\\ (X_{\text{rabbit}}(t)-X_{\text{m}}(t))-r_3(LB+r_4(UB-LB)), & q<0.5\end{cases}\tag{1}$$

$$X_{\text{m}}(t)=\frac{1}{N}\sum_{i=1}^{N}X_i(t)\tag{2}$$

式中:$X(t)$为第 t 次迭代时的个体位置;$X_{\text{rabbit}}(t)$和 $X_{\text{rand}}(t)$分别为猎物和随机选择的个体的位置;r_i ($i=1,\cdots,4$)和 q 都是(0,1)内服从均匀分布的随机数;UB 和 LB 为搜索空间的上界和下界;$X_{\text{m}}(t)$为当前种群的平均位置;N 为种群规模。

2) 探索到开发的转换。HHO 根据猎物能量 E 实现探索到开发的转换,猎物逃逸过程中能量呈线性递减,计算公式为:

$$E=2E_0(1-t/T)\tag{3}$$

式中:T 为最大迭代次数;E_0 为能量迭代的初始值,每次迭代在(-1,1)内随机变化。当 $|E|\geqslant 1$ 时,Harris 鹰在解空间中探索猎物的位置,执行探索,反之则在解邻域内执行开发。

3) 开发阶段。通过 r 刻画猎物是否成功地逃逸,当 $r<0.5$ 时成功逃逸,反之则失败,同时也根据 E 值大小执行不同的包围策略,包括如下 4 种情形。

a) 软包围。当 $r\geqslant 0.5$ 且 $|E|\geqslant 0.5$ 时,猎物能量充足并采用随机跳跃的方式进行逃逸,但逃逸失败,此时鹰更新位置如下:

$$X(t+1)=\Delta X(t)-E\left|JX_{\text{rabbit}}(t)-X(t)\right|\tag{4}$$

式中:$\Delta X(t)=X_{\text{rabbit}}(t)-X(t)$;$J=2(1-r_5)$ 为随

机跳跃强度;r_5为(0,1)内服从均匀分布的随机数。

b) 硬包围。当 $r\geqslant 0.5$ 且 $|E|<0.5$ 时,猎物能量较低且逃逸失败,鹰的位置更新公式为:

$$X(t+1)=X_{\text{rabbit}}(t)-E|\Delta X(t)| \tag{5}$$

c) 渐进式快速俯冲的软包围。当 $r<0.5$ 且 $|E|\geqslant 0.5$时,猎物有充足的能量来确保逃逸,鹰采取下式进行位置更新:

$$X(t+1)=\begin{cases}Y:X_{\text{rabbit}}(t)-E|JX_{\text{rabbit}}(t)-X(t)|, & f(Y)<f(X(t))\\ Z:Y+S\times LF(D), & f(Z)<f(X(t))\end{cases} \tag{6}$$

式中:f 为适应度函数;D 为问题维数;S 为 D 维随机向量;LF 为 Levy 飞行函数[5]。

d) 渐进式快速俯冲的硬包围。当 $r<0.5$ 且 $|E|<0.5$时,猎物没有足够能量逃逸,在突袭前鹰采取硬包围捕获猎物,鹰位置更新如下:

$$X(t+1)=\begin{cases}Y:X_{\text{rabbit}}(t)-E|JX_{\text{rabbit}}(t)-X_{\text{m}}(t)|, & f(Y)<f(X(t))\\ Z:Y+S\times LF(D), & f(Z)<f(X(t))\end{cases} \tag{7}$$

1.2 带有 Logistic 混沌扰动的控制参数策略

在 HHO 中,控制参数 $2(1-t/T)$的大小直接影响着 E 的变化,且控制参数随迭代次数的增加从 2 线性递减至 0,下降速率不变,即迭代后期 E 的绝对值小于 1,不再进行全局搜索。而对于一些复杂优化问题,在探索后期,无法保证种群已经聚集在全局最优解附近,因此线性更新的控制参数不能反映出 Harris 鹰的实际捕猎过程。

本文采用 Logistic 映射[6]产生混沌序列,进而对控制参数进行混沌扰动,给出一种带有 Logistic 混沌扰动的控制参数策略,对应能量 E 的更新公式如下:

$$E=\left[E_{\text{initial}}|y^t|-(E_{\text{initial}}-E_{\text{final}})\mathrm{e}^{-\left[t/\left(\frac{T}{4}\right)\right]^2}\right]E_0 \tag{8}$$

式中:E_{initial}、E_{final}分别为控制参数的初值和终值;t 和 T 分别为当前迭代次数和最大迭代次数;y^t 为 Logistic 映射产生的混沌序列,Logistic 映射的定义式为:

$$y^t=\mu y^{t-1}(1-y^{t-1}) \tag{9}$$

式中:$\mu\in[0,4]$,$y^t\in[0,1]$。本文取 $\mu=4$。

通过引入带有 Logistic 混沌扰动的控制参数策略,增强了算法全局探索能力,平衡了算法全局探索和局部开发能力,可以避免算法过早收敛。

1.3 集成变异策略

HHO 算法的开发阶段采用多种更新策略进行局部搜索,因而其局部开发能力优于全局探索能力[5]。文献[7]指出集成变异策略可以增强群智能优化算法的探索和开发能力,本文在算法探索阶段引入集成变异策略[7],第 i 个 Harris 鹰个体 X_i的 3 个候选位置 V_{i1}、V_{i2}和 V_{i3}由式(10)~(12)并行产生,这 3 种变异策略分别被称为 DE/rand/1/bin,DE/rand/2/bin 和 DE/current-to-rand/1/bin。

$$V_{i1,j}=\begin{cases}X_{R_1,j}+F_1\times(X_{R_2,j}-X_{R_3,j}), & \text{rand}<C_{R1}\text{或 }j=j_{\text{rand}}\\ X_{i,j}, & \text{其它}\end{cases} \tag{10}$$

$$V_{i2,j}=\begin{cases}X_{R_4,j}+F_2\times(X_{R_5,j}-X_{R_6,j})+F_2\times(X_{R_7,j}-X_{R_8,j}), & \text{rand}<C_{R2}\text{或 }j=j_{\text{rand}}\\ X_{i,j}, & \text{其它}\end{cases} \tag{11}$$

$$V_{i3,j}=\begin{cases}X_{i,j}+\text{rand}\times(X_{R_9,j}-X_{i,j})+F_2\times(X_{R_{10},j}-X_{R_{11},j}), & \text{rand}<C_{R3}\text{或 }j=j_{\text{rand}}\\ X_{i,j}, & \text{其它}\end{cases} \tag{12}$$

式中:$R_k(k=1,2,\ldots,11)\in[1,N]$且为均不等于 i 的不同整数;维数 $j_{\text{rand}}\in[1,D]$;尺度因子 F_i $(i=1,2,3)$的值分别为 1,0.8 和 1;交叉率 C_{R1},C_{R2}和 C_{R3}的值分别为 0.1,0.2 和 0.9;rand 函数产生 0 和 1 之间的服从均匀分布的随机数。

在候选解 V_{i1}、V_{i2}和 V_{i3}更新之后,将其中具有最小适应度值的候选解记为 V_i,然后采用下式进行贪婪选择:

$$X_i=\begin{cases}V_i, & f(V_i)<f(X_i)\\ X_i, & f(V_i)\geqslant f(X_i)\end{cases} \tag{13}$$

1.4 翻筋斗觅食策略

迭代后期,Harris 鹰种群将会聚集在当前猎物附近,导致种群多样性降低、算法容易陷入局部最优。因此,受文献[8]中蝠鲼通过向后翻筋斗围绕浮游生物来回游动以此来获取食物源的启发,本文引入翻筋斗觅食策略来增强种群多样性和避免 HHO

陷入局部最优。该策略将猎物看成一个支点,每一个 Harris 鹰个体都会移动到其当前位置与对称于支点之间的任何位置 X_s,以寻求围捕猎物的最佳区域,公式如下:

$$X_s(t) = X(t) + s(r_6 X_{\text{rabbit}}(t) - r_7 X(t)) \quad (14)$$

式中:s 为空翻因子,决定了 Harris 鹰的空翻范围,取 $s=2$[8];r_6 和 r_7 是(0, 1)内服从均匀分布的随机数。

在迭代时,对翻筋斗后的 Harris 鹰个体进行贪婪选择,即:

$$X(t) = \begin{cases} X_s(t) & f(X_s(t)) < f(X(t)) \\ X(t), & f(X_s(t)) \geqslant f(X(t)) \end{cases} \quad (15)$$

翻筋斗觅食策略可以使 Harris 鹰个体在变化的搜索范围内进行自适应搜索,提高了种群多样性,避免了算法陷入局部最优,从而进一步加快了算法收敛。

1.5 IHHO 算法

IHHO 算法步骤如下。

Step 1:初始化参数,包括种群规模 N,维数 D,最大迭代次数 T,搜索空间上界 UB 和下界 LB。

Step 2:在搜索空间中随机产生初始的 Harris 鹰种群 $X_i(i=1, 2, \ldots, N)$。

Step 3:计算所有个体适应度 $f(X_i)$,将 X_{rabbit} 设置为猎物的位置(当前最好位置)。

Step 4:利用式(8)更新猎物的逃逸能量 E。

Step 5:当 $|E|\geqslant 1$ 时,执行探索阶段:

根据(1)式更新个体位置;执行集成变异策略,采用式(10)~(12)更新 V_{i1}、V_{i2} 和 V_{i3},选取具有最优适应度的候选解记为 V_i,并根据(13)进行贪婪选择。

Step 6:当 $|E|<1$ 时,执行开发阶段:

当 $r\geqslant 0.5$ 且 $|E|\geqslant 0.5$,根据式(4)更新个体位置;当 $r\geqslant 0.5$ 且 $|E|<0.5$,根据式(5)更新个体位置;当 $r<0.5$ 且 $|E|\geqslant 0.5$,根据式(6)更新个体位置;当 $r<0.5$ 且 $|E|<0.5$,根据式(7)更新个体位置。

Step 7:执行翻筋斗觅食策略,采用式(14)更新位置,并根据式(15)进行贪婪选择。

Step 8:判断算法是否达到最大迭代次数,若是则算法结束,输出当前猎物位置,否则返回 Step 3。

1.6 算法复杂度分析

由文献[5]可知,HHO 算法总的复杂度为:

$$O(\text{HHO}) = O(N) + O(T\times N) + O(T\times N\times D) = O(T\times N\times D) \quad (16)$$

IHHO 算法基于 HHO 引入了带有 Logistic 混沌扰动的控制参数、集成变异和翻筋斗觅食 3 种策略,其中带有 Logistic 混沌扰动的控制参数策略并不增加额外的复杂度,集成变异策略的计算复杂度为 $O(3\times T\times N\times D)$,而翻筋斗觅食策略的复杂度为 $O(T\times N\times D)$。因此,IHHO 总的复杂度为:

$$\begin{aligned} O(\text{IHHO}) &= O(\text{HHO}) + O(3\times T\times N\times D) + O(T\times N\times D) = \\ &O(T\times N\times D) + O(3\times T\times N\times D) + O(T\times N\times D) = \\ &O(T\times N\times D) \end{aligned} \quad (17)$$

综上,IHHO 和 HHO 算法相比,运算量增加了一点,复杂度是一样的,但算法的收敛精度和稳定性都得到了提高。

2 数值实验

选取 CEC 2014 基准测试集进行仿真实验,该测试集包括单峰、多峰、混合和复合 4 种类型的 30 个函数。所有的实验均在 MATLAB 2015 下进行,为公平起见,所有算法采用相同的初始化种群,各参数设置为种群规模 $N=30$, 维数 $D=30$,最大迭代次数 $T=1\ 000$, $E_{\text{initial}}=2$, $E_{\text{final}}=0$,各算法在每个测试函数上独立运行 51 次,记录平均值和标准差。

2.1 策略有效性分析

选取单峰函数 F1、F2,多峰函数 F5、F10,混合函数 F18、F19 和组合函数 F26、F30 进行策略有效性分析。将仅采用“带有 Logistic 混沌扰动的控制参数策略”,“集成变异策略”,“翻筋斗觅食策略”的 HHO 分别记为 LHHO,EHHO,SHHO,将 IHHO 与上述三种改进的 HHO 和基本 HHO 进行实验对比,实验结果见表 1,最好结果用粗体表示。

由表 1 可知,采用单一策略的 LHHO、EHHO 和 SHHO 在 8 个测试函数上的平均值和标准差都优于 HHO 算法,IHHO 算法在 7 个函数上的平均值和标准差均达到了最小,在函数 F26 上,IHHO 算法性能优于 HHO、LHHO 和 SHHO,仅比 EHHO 略差。实验结果体现出本文改进策略的有效性,表明 IHHO 的求解精度和稳定性得到了明显提高。

表 1 单策略改进的 HHO 和 IHHO 在 8 个测试函数上的实验结果

Tab. 1 Experimental results of improved HHO by single strategy and IHHO on 8 benchmark functions

函数	指标	HHO	LHHO	EHHO	SHHO	IHHO
F1	平均值	4.565 4E+07	3.563 6E+07	9.402 8E+06	3.099 1E+07	**2.405 7E+06**
	标准差	1.764 1E+07	1.503 5E+07	4.403 0E+06	1.545 8E+07	**9.659 5E+05**
F2	平均值	3.718 3E+07	3.149 0E+07	3.417 8E+06	2.835 2E+07	**2.058 8E+04**
	标准差	1.169 2E+07	1.124 0E+07	1.434 6E+06	5.745 7E+06	**1.005 1E+04**
F5	平均值	5.207 4E+02	5.206 7E+02	5.205 3E+02	5.207 3E+02	**5.200 6E+02**
	标准差	1.618 3E−01	8.140 7E−02	1.275 9E−01	1.291 6E−01	**7.389 8E−02**
F10	平均值	4.514 6E+03	4.098 3E+03	1.502 1E+03	4.057 9E+03	**1.189 2E+03**
	标准差	6.954 2E+02	6.851 4E+02	1.995 3E+02	5.614 4E+02	**1.006 0E+02**
F18	平均值	1.926 4E+06	2.802 2E+05	7.355 8E+04	1.520 7E+05	**1.142 8E+04**
	标准差	3.652 8E+06	4.599 7E+05	1.130 1E+05	1.061 2E+05	**1.117 6E+04**
F19	平均值	1.983 5E+03	1.942 1E+03	1.911 1E+03	1.973 6E+03	**1.910 3E+03**
	标准差	5.375 2E+01	2.616 3E+01	2.199 0E+00	3.192 6E+01	**1.148 2E+00**
F26	平均值	2.760 2E+03	2.740 4E+03	**2.700 4E+03**	2.760 2E+03	2.710 4E+03
	标准差	5.140 4E+01	5.131 0E+01	**7.551 5E−02**	5.136 6E+01	3.146 7E+01
F30	平均值	3.775 8E+04	2.983 9E+04	7.265 4E+03	9.773 0E+03	**6.267 2E+03**
	标准差	7.727 6E+04	3.100 4E+04	1.253 6E+03	1.469 8E+04	**1.233 9E+03**

2.2 IHHO 与其他群智能算法的比较

选取奇数组的 15 个测试函数,将 IHHO 算法与 PSO[9]、BA[10]、SSA[11]、MVO[12]和 MFO[13]进行仿真实验比较,实验结果见表 2,其中最好结果用粗体表示。

表 2 IHHO 与其他群智能算法对比实验结果

Tab. 2 Comparison of experimental results of IHHO and other swarm intelligence algorithms

函数	指标	PSO	BA	SSA	MVO	MFO	IHHO
F1	平均值	5.081 6E+08	3.513 4E+09	2.047 8E+07	1.075 6E+07	2.016 6E+08	**4.072 4E+06**
	标准差	1.583 7E+09	1.204 8E+09	1.085 1E+07	3.223 9E+06	2.187 3E+08	**1.206 8E+06**
F3	平均值	4.603 0E+03	3.674 7E+06	7.792 1E+04	7.354 0E+03	1.102 7E+05	**7.477 7E+02**
	标准差	2.831 5E+03	9.864 9E+06	2.080 2E+04	1.985 1E+03	4.231 0E+04	**5.018 1E+02**
F5	平均值	5.209 5E+02	5.212 5E+02	5.201 0E+02	5.204 7E+02	5.203 4E+02	**5.200 4E+02**
	标准差	6.987 7E−02	1.014 6E−01	1.633 7E−01	1.322 3E−01	2.192 2E−01	**5.584 0E−02**
F7	平均值	7.029 8E+02	1.751 1E+03	**7.000 1E+02**	7.006 9E+02	7.854 6E+02	7.000 2E+02
	标准差	4.790 7E+00	2.060 3E+02	**7.736 1E−03**	1.118 8E−01	7.090 2E+01	1.993 8E−02
F9	平均值	**1.026 3E+03**	1.483 2E+03	1.040 9E+03	1.034 4E+03	1.112 2E+03	1.026 7E+03
	标准差	**1.491 7E+01**	6.893 5E+01	4.201 4E+01	4.332 0E+01	5.199 6E+01	3.189 0E+01
F11	平均值	4.253 9E+03	1.057 3E+04	5.060 4E+03	4.711 6E+03	5.289 6E+03	**3.716 7E+03**
	标准差	4.958 3E+02	4.783 8E+02	5.748 9E+02	3.696 6E+02	7.673 4E+02	**2.492 8E+02**
F13	平均值	1.301 7E+03	1.310 0E+03	1.300 6E+03	1.300 7E+03	1.301 2E+03	**1.300 5E+03**
	标准差	3.897 0E+00	1.323 9E+00	1.221 3E−01	1.181 3E−01	8.033 0E−01	**9.080 4E−02**
F15	平均值	1.513 5E+03	1.417 9E+07	1.513 1E+03	**1.511 6E+03**	2.386 6E+05	1.512 6E+03
	标准差	2.897 9E+00	1.881 3E+07	4.629 4E+00	**2.822 0E+00**	3.601 9E+05	3.100 7E+00

表 2(续)

函数	指标	PSO	BA	SSA	MVO	MFO	IHHO
F17	平均值	8.033 3E+07	2.765 7E+08	1.258 8E+06	6.631 8E+05	5.340 5E+06	**6.523 3E+05**
	标准差	1.460 6E+08	1.205 0E+08	1.019 6E+06	5.437 6E+05	3.401 2E+06	**4.099 9E+05**
F19	平均值	2.046 4E+03	2.804 9E+03	1.915 6E+03	1.912 8E+03	1.975 0E+03	**1.909 4E+03**
	标准差	3.865 5E+02	3.219 4E+02	3.434 5E+00	1.912 3E+00	7.527 8E+01	**9.352 1E−01**
F21	平均值	1.596 0E+05	1.274 1E+08	4.938 4E+05	2.017 9E+05	1.043 0E+06	**1.546 6E+05**
	标准差	**9.481 8E+04**	7.228 1E+07	4.335 2E+05	1.727 1E+05	1.266 0E+06	1.109 8E+05
F23	平均值	2.616 4E+03	4.504 9E+03	2.633 7E+03	2.625 2E+03	2.674 3E+03	**2.500 0E+03**
	标准差	2.014 3E+00	8.158 4E+02	1.038 9E+01	6.255 3E+00	5.611 9E+01	**0.000 0E+00**
F25	平均值	2.712 9E+03	2.948 7E+03	2.717 3E+03	2.706 9E+03	2.721 5E+03	**2.700 0E+03**
	标准差	3.240 9E+00	6.960 6E+01	3.527 7E+00	1.685 6E+00	1.186 0E+01	**0.000 0E+00**
F27	平均值	4.024 6E+03	4.360 0E+03	3.627 5E+03	3.337 8E+03	3.577 0E+03	**2.900 0E+03**
	标准差	7.121 1E+02	1.914 2E+02	1.159 7E+02	1.668 3E+02	2.676 9E+02	**0.000 0E+00**
F29	平均值	7.091 7E+08	1.025 7E+06	7.100 2E+06	1.457 8E+06	2.378 7E+06	**3.789 8E+03**
	标准差	5.090 3E+08	1.196 9E+06	9.635 8E+06	4.528 7E+06	3.781 7E+06	**7.570 9E+02**

从表 2 可以看出,IHHO 算法在 11 个测试函数上都得到了最好的实验结果。在函数 F7、F9 和 F15 上,IHHO 算法取得了较好的结果,其性能优于 4 个对比算法,仅劣于 1 个对比算法;而在函数 F21 上,IHHO 算法的平均值达到了最好。综上所述,IHHO 算法的寻优效果好,与其他对比算法相比具有明显优势。

2.3 IHHO 与其他改进 HHO 算法的比较

选取偶数组的 15 个测试函数,将 IHHO 与基本 HHO 和 4 种改进的 HHO 算法包括 EESHHO[14]、CCNMHHO[15]、QRHHO[16] 和 GCHHO[17] 进行实验对比,实验结果见表 3,最好结果用粗体表示。

由表 3 可知,从平均值来看 IHHO 在 12 个函数上都获得了最好的结果。在函数 F14 上,IHHO 算法结果劣于 HHO 算法;在函数 F26 上,IHHO 算法性能仅弱于 GCHHO 算法,但强于其他 4 种算法;在函数 F30 上,QRHHO 算法获得了最好的实验结果。总体上说,IHHO 算法在 6 种算法中具有最好的寻优性能。

表 3 IHHO 与其他改进 HHO 算法的对比实验结果

Tab.3 Comparison of experimental results of IHHO and other improved HHO algorithms

函数	指标	HHO	EESHHO	CCNMHHO	QRHHO	GCHHO	IHHO
F2	平均值	3.348 0E+07	7.752 5E+09	1.238 4E+07	3.119 7E+07	1.572 4E+04	**1.566 1E+04**
	标准差	7.315 3E+06	6.132 4E+09	4.082 7E+06	5.508 3E+06	1.374 5E+04	**1.336 0E+04**
F4	平均值	6.024 1E+02	8.940 8E+02	6.118 8E+02	5.798 4E+02	5.017 2E+02	**5.009 0E+02**
	标准差	7.075 6E+01	2.591 8E+02	6.451 6E+01	3.476 3E+01	3.041 7E+01	**2.8798E+01**
F6	平均值	6.355 0E+02	6.299 9E+02	6.287 3E+02	6.325 1E+02	6.313 7E+02	**6.181 5E+02**
	标准差	3.263 7E+00	3.033 0E+00	**2.962 0E+00**	3.254 4E+00	4.954 9E+00	3.180 8E+00
F8	平均值	9.439 1E+02	9.515 8E+02	9.268 5E+02	9.402 1E+02	9.176 5E+02	**8.047 3E+02**
	标准差	2.104 5E+01	2.476 3E+01	1.059 9E+01	2.034 5E+01	2.116 9E+01	**1.249 0E+00**
F10	平均值	4.293 0E+03	3.565 2E+03	2.800 7E+03	3.568 3E+03	2.894 3E+03	**1.135 0E+03**
	标准差	7.130 9E+02	5.806 8E+02	7.536 5E+02	4.422 2E+02	5.935 5E+02	**9.937 0E+01**
F12	平均值	1.202 1E+03	1.200 9E+03	1.201 0E+03	1.201 7E+03	1.200 8E+03	**1.200 5E+03**
	标准差	5.265 1E−01	2.973 7E−01	4.141 1E−01	4.059 4E−01	4.724 2E−01	**1.583 3E−01**
F14	平均值	**1.400 3E+03**	1.409 6E+03	1.400 4E+03	1.400 4E+03	1.400 4E+03	1.400 4E+03
	标准差	**4.404 9E−02**	1.071 7E+01	4.976 5E−02	2.413 2E−01	2.399 7E−01	2.177 2E−01

表 3(续)

函数	指标	HHO	EESHHO	CCNMHHO	QRHHO	GCHHO	IHHO
F16	平均值	1.612 3E+03	1.612 3E+03	1.612 0E+03	1.612 2E+03	1.611 9E+03	**1.611 3E+03**
	标准差	5.595 3E−01	**3.194 6E−01**	5.021 2E−01	4.917 8E−01	4.100 6E−01	5.097 4E−01
F18	平均值	2.625 4E+05	1.047 6E+05	7.374 9E+03	1.150 6E+05	6.897 1E+03	**5.240 0E+03**
	标准差	1.199 4E+05	2.022 0E+05	5.947 5E+03	5.924 0E+04	4.894 5E+03	**3.973 6E+03**
F20	平均值	2.980 5E+04	3.125 5E+04	8.911 4E+03	3.203 4E+04	1.453 5E+04	**2.201 2E+03**
	标准差	9.954 4E+03	1.072 5E+04	3.271 8E+03	8.559 5E+03	5.974 2E+03	**2.116 9E+02**
F22	平均值	3.097 3E+03	3.048 1E+03	2.767 1E+03	3.068 6E+03	2.975 1E+03	**2.571 2E+03**
	标准差	2.592 6E+02	2.194 8E+02	2.278 0E+02	2.524 8E+02	2.500 1E+02	**1.874 2E+02**
F24	平均值	**2.600 0E+03**	**2.600 0E+03**	**2.600 0E+03**	**2.600 0E+03**	**2.600 0E+03**	**2.600 0E+03**
	标准差	1.208 3E−04	1.740 1E−04	1.266 1E−04	**0.000 0E+00**	3.785 4E−04	**0.000 0E+00**
F26	平均值	2.760 2E+03	2.770 2E+03	2.753 5E+03	2.780 1E+03	**2.700 5E+03**	2.710 4E+03
	标准差	5.137 7E+01	4.799 0E+01	4.995 4E+01	4.204 0E+01	**1.522 5E−01**	3.147 0E+01
F28	平均值	**3.000 0E+03**	**3.000 0E+03**	3.348 6E+03	**3.000 0E+03**	**3.000 0E+03**	**3.000 0E+03**
	标准差	**0.000 0E+00**	**0.000 0E+00**	2.662 0E+02	**0.000 0E+00**	**0.000 0E+00**	**0.000 0E+00**
F30	平均值	5.230 3E+04	6.214 0E+04	3.804 7E+03	**3.200 0E+03**	1.048 1E+04	6.165 2E+03
	标准差	1.119 3E+05	3.393 8E+04	7.807 0E+02	**0.000 0E+00**	4.612 4E+03	1.898 4E+03

3 基于 IHHO 与 K-Medoids 结合的聚类算法

3.1 K-Medoids 算法

K-Medoids 是一种划分聚类算法，旨在将数据集 X 中 n 个 d 维对象分配到 k 个中心点 $o=\{o_1, o_2, \ldots, o_k\}$ 所在簇 $c_1, c_2, \ldots, c_k$ 中，使得对于 $1\leqslant i, j\leqslant k$，$c_i\subseteq X$ 并且 $c_i\cap c_j=\varnothing\ (i\neq j)$，$\bigcup_{i=1}^{k}c_i=X$，并且簇内对象相似，而不同簇中对象相异[18]。K-Medoids 采用实际对象作为聚类中心，相比 K-Means 可以降低对噪声点和离群点的敏感性，使算法具有鲁棒性。围绕中心点划分(partitioning around medoids，PAM)算法是 K-Medoids 的一种主流实现[18]，其步骤为：随机选取 k 个对象作为初始代表对象，而剩余对象则根据与代表对象之间的欧氏距离分配给最近代表对象所代表的簇，并继续用非代表对象来代替代表对象的迭代过程，直到聚类中心不发生变化。本文代价函数采用下式计算：

$$AE=\sum_{i=1}^{k}\sum_{j=1}^{n_i}\sqrt{\sum_{m=1}^{d}(x_{jm}-o_{im})^2} \tag{18}$$

式中 n_i 表示各个簇中包含对象的个数。

3.2 基于 IHHO 与 K-Medoids 的混合聚类算法

IHHO 算法具有全局搜索能力强，求解精度高的特点，但运行时间成本较大。K-Medoids 收敛速度快，但容易陷入局部最优的不足。为此，本文提出一种基于 IHHO 与 K-Medoids 的混合聚类算法 IHHO-KMedoids。该算法保持了 IHHO 和 K-Medoids 各自的优点，利用 IHHO 高效的寻优性能可以避免算法陷入局部最优，且凭借 K-Medoids 鲁棒性强、不易受噪声和异常点影响的特点改善了 K-Means 存在的不足，从而可以快速且准确地找出一组最佳聚类中心。

IHHO-KMedoids 算法步骤如下。

Step 1：设置算法参数，包括种群规模 N，最大迭代次数 T，数据集中对象的维数 d，簇的个数 k。

Step 2：在搜索空间中随机产生初始种群，每个 Harris 鹰个体 $X_i(i=1, 2, \ldots, N)$的维数为 $k\times d$。

Step 3：在数据集中随机选取 k 个对象作为初始的聚类中心。

Step 4：根据式(18)计算适应度，将具有最好适应度的个体位置记为当前猎物位置，并执行一次 K-Medoids聚类操作。

Step 5：执行 IHHO 算法中的 Step 4～Step 7。

Step 6：判断算法是否满足终止条件，若满足，则算法结束，输出当前猎物位置即为最佳聚类中心点，否则返回 Step 4。

3.3 仿真实验与分析

3.3.1 聚类评价指标

选取 F-Measure[19]、调整 Rand 指数(adjusted Rand index、ARI)[20] 和标准化互信息(normalized

mutual information、NMI)[21]作为评价指标来衡量各算法的聚类性能,其定义见如下。

1) F-Measure 指标

F-Measure 指精度和召回率的调和平均,即:

$$F\text{-}Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{19}$$

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

式中:*Precision* 和 *Recall* 表示精度和召回率,*TP*、*TN*、*FP* 和 *FN* 分别表示真阳性、真阴性、假阳性和假阴性的个数。*F-Measure* 综合了 *Precision* 和 *Recall* 的结果,取值范围为[0,1],其值越大说明聚类结果越准确。

2) ARI 指标

Rand 指数(RI)计算了两个簇之间的相似性度量,其定义如下:

$$RI = \frac{TP + TN}{TP + FP + TN + FN} \tag{22}$$

调整 Rand 指数 ARI 在 RI 基础上对其进行了重新调整,公式如下:

$$ARI = \frac{RI - Expected\_RI}{\max(RI) - Expected\_RI} \tag{23}$$

式中 *ARI* 的取值范围为[−1,1],值越大表明聚类结果与真实情况的吻合度越高。

3) NMI 指标

互信息(MI)可计算聚类后的类标签 X 和真实类标签 Y 的相关程度,NMI 表示 MI 的归一化,计算公式为:

$$NMI = \frac{MI}{F(H(X), H(Y))} \tag{24}$$

式中:$H(X)$和 $H(Y)$分别表示 X 和 Y 的熵;F 为归一化函数,这里采取算术平均的方法;*NMI* 在[0,1]内取值,值越接近于 1 表明聚类效果越好。

3.3.2 UCI 数据集实验

将 IHHO-KMedoids 与 K-Means、K-Medoids 和 HHO-KMedoids 进行实验对比,实验数据采用 UCI 数据库中的 Iris、Wine、Glass 和 Zoo 数据集,表 4 为所有数据集的具体描述。设置实验参数,种群规模为 30,最大迭代次数为 1 000,实验独立运行 30 次,记录适应度函数的最好值、平均值、最差值和标准差,并依据平均值结果进行排序,实验结果见表 5,其中最好结果用粗体表示。

表 4 数据集描述

Tab. 4 Description of datasets

数据集	个数	维数	簇数	各簇包含个数
Iris	150	4	3	50,50,50
Wine	178	13	3	59,71,48
Glass	214	9	6	70,17,76,13,9,29
Zoo	101	17	7	41,20,5,13,4,8,10

由表 5 可知,IHHO-KMedoids 算法在所有数据集上的最好值、平均值、最差值和标准差都达到了最小,在 4 个算法中排名第一。在 Iris 数据集上,IHHO-KMedoids 的最好值、平均值和最差值都为 96.540 3,标准差为 $1.055\ 7\times10^{-10}$。在 Wine 数据集上,IHHO-KMedoids 的最差值 16 292.669 1 和最好值 16 292.186 5 之差仅为 0.482 6,而性能较好的 HHO-KMedoids 的最差值与最好值的差为 6.700 4。而对于 Glass 和 Zoo 来说,IHHO-KMedoids 算法相比其他算法的目标函数值得到了很大的改善。表 6 给出这 4 种算法在所有数据集上的 3 个聚类评价指标结果,其中最好的结果以粗体显示。由表 6 可知,IHHO-KMedoids 在 Iris 数据集上的 F-Measure、ARI 和 NMI 值显著大于 K-Medoids,说明 IHHO-KMedoids 改善了 K-Medoids 容易陷入局部最优的不足。在 Wine 和 Glass 数据集上,IHHO-KMedoids 的聚类指标结果优于 K-Means、K-Medoids 和 HHO-KMedoids。在 Zoo 数据集上,IHHO-KMedoids 的 F-Measure、ARI 和 NMI 值分别为 0.879 8、0.845 0 和 0.868 5,相比其他算法得到了明显的提高,体现出所提算法的优越性能。

表 5 4 种算法在 UCI 数据集上的实验结果

Tab. 5 Experimental results by 4 algorithms on UCI datasets

数据集	指标	K-Means	K-Medoids	HHO-KMedoids	IHHO-KMedoids
Iris	最好值	97.204 6	98.131 2	96.583 5	**96.540 3**
	平均值	97.208 6	103.235 8	96.639 0	**96.540 3**
	最差值	97.224 9	123.654 5	96.692 2	**96.540 3**
	标准差	9.076 4E−03	1.141 4E+01	4.124 9E−02	**1.055 7E−10**
	Rank	3	4	2	**1**

表 5(续)

数据集	指标	K-Means	K-Medoids	HHO-KMedoids	IHHO-KMedoids
Wine	最好值	16 555.679 4	16 375.889 1	16 303.373 4	**16 292.1865**
	平均值	17 232.038 3	17 039.057 9	16 306.322 3	**16 292.5717**
	最差值	18 436.952 1	18 033.811 2	16 310.073 8	**16 292.669 1**
	标准差	9.358 7E+02	9.080 8E+02	2.451 9E+00	**2.153 6E−01**
	Rank	4	3	2	**1**
Glass	最好值	215.677 5	220.594 3	217.382 7	**215.059 7**
	平均值	230.266 3	238.643 4	236.508 5	**224.359 7**
	最差值	245.887 4	261.772 6	249.178 6	**233.801 6**
	标准差	1.274 0E+01	1.771 4E+01	1.721 2E+01	**8.349 9E+00**
	Rank	2	4	3	**1**
Zoo	最好值	119.147 7	121.665 3	120.276 9	**107.309 5**
	平均值	126.536 8	127.814 9	124.160 8	**109.896 3**
	最差值	140.491 1	136.469 7	130.632 6	**112.896 3**
	标准差	8.135 4E+00	6.394 5E+00	4.103 9E+00	**1.984 5E+00**
	Rank	3	4	2	**1**

表 6 4 种算法在 UCI 数据集上的聚类评价指标结果

Tab.6 Clustering evulation index results by 4 algorithms on UCI datasets

数据集	评价指标	K-Means	K-Medoids	HHO-KMedoids	IHHO-KMedoids
Iris	F-Measure	0.811 1	0.655 2	0.818 7	**0.829 3**
	ARI	0.716 3	0.432 9	0.728 7	**0.743 7**
	NMI	0.741 9	0.610 9	0.743 3	**0.766 1**
Wine	F-Measure	0.583 5	0.583 4	0.580 6	**0.595 4**
	ARI	0.371 1	0.371 5	0.367 6	**0.390 7**
	NMI	0.428 8	0.419 3	0.417 9	**0.434 0**
Glass	F-Measure	0.478 7	0.380 5	0.517 8	**0.527 6**
	ARI	0.257 9	0.208 5	0.248 8	**0.267 6**
	NMI	0.399 5	0.326 9	0.419 0	**0.465 2**
Zoo	F-Measure	0.814 4	0.672 4	0.759 4	**0.879 8**
	ARI	0.761 5	0.588 0	0.672 9	**0.845 0**
	NMI	0.803 1	0.726 3	0.767 2	**0.868 5**

3.3.3 文本数据集实验

本节实验选取清华大学中文文本分类数据集 THUCnews 进行测试,从家居、彩票、房产、股票和财经五类中每类各获取 100 篇文档混合,将这 500 篇文档作为文本聚类的测试集。在实验前,对数据集进行文本预处理和文本表示。

1) 文本预处理。包括中文分词、去停用词和特征选择。中文分词是指将给定中文文本切分成以词汇为单位的过程,本文采用 Python 中的 jieba 分词对文本进行分词处理。分词后,文本中还存在大量功能词,通常指一些频繁出现、附带极少信息的助词、介词、语气词等,比如“的,了,是,呢”等,这些词被称为停用词,它们出现频率极高但没有实质意义,因此为减少文本特征维度,通常将这些词进行过滤。经分词和去停用词后,文本中仍存在大量冗余特征,特征维度较大,因此要对高维文本进行特征选择以减少低频词对后续聚类的影响。

2) 基于潜在语义分析(LSA)的文本表示。在聚类前将上述预处理后的非结构化数据表示为数值型数据,采用向量空间模型进行文本向量化,对于包含 n 个文本的文档集,若预处理后的特征集包含 m 个特征项,则根据特征项对应的权重构造出特征项-文档矩阵 $\boldsymbol{X}$,使用 TF-IDF 表示特征项的权重,则矩阵 $\boldsymbol{X}$ 可表示如下:

$$\boldsymbol{X}=\begin{pmatrix} tf\text{-}idf_{1,1} & \cdots & tf\text{-}idf_{1,n} \\ \vdots & \ddots & \vdots \\ tf\text{-}idf_{m,1} & \cdots & tf\text{-}idf_{m,n} \end{pmatrix} \tag{25}$$

式中 $tf\text{-}idf_{i,j}$ 表示第 i 个特征项在第 j 个文本中对应的 TF-IDF 权重值。

LSA 技术利用奇异值分解将文档和特征项的高维表示映射到低维潜在语义空间中,从而实现高维原始矩阵的约减和近似[22]。对特征项-文档矩阵 $\boldsymbol{X}$ 进行奇异值分解:

$$\boldsymbol{X}=\boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{D}^{\mathrm{T}} \tag{26}$$

式中:$\boldsymbol{T}$ 和 $\boldsymbol{D}$ 分别为 $m\times r$ 和 $n\times r$ 的正交矩阵;$\boldsymbol{\Sigma}=\mathrm{diag}(\sigma_1,\cdots,\sigma_r)$ 是由 r 个非零奇异值 $\sigma_1,\cdots,\sigma_r$(其中 $\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_r>0$)构成的 r 阶对角矩阵。

对奇异值从大到小排序,选择保留前 $l(l<r)$ 个最大的奇异值,得到 $\boldsymbol{X}$ 的低秩近似矩阵 $\boldsymbol{X}_l$ 为:

$$\boldsymbol{X}_l=\boldsymbol{T}_l\boldsymbol{\Sigma}_l\boldsymbol{D}_l^{\mathrm{T}} \tag{27}$$

式中:$\boldsymbol{X}_l$ 是一个 $m\times n$ 的矩阵;$\boldsymbol{T}_l$ 和 $\boldsymbol{D}_l$ 分别是 $m\times l$ 和 $n\times l$ 的矩阵,$\boldsymbol{T}_l$ 的列向量所组成的 l 维线性空间构成文本的潜在语义空间。

$\boldsymbol{X}$ 中的文档可通过 $\boldsymbol{T}_l$ 映射到该空间得到其低维表示 $\boldsymbol{A}$,具体公式如下:

$$\begin{cases}\boldsymbol{X}\approx\boldsymbol{X}_l=\boldsymbol{T}_l\boldsymbol{\Sigma}_l\boldsymbol{D}_l^{\mathrm{T}}\\ \boldsymbol{T}_l^{\mathrm{T}}\boldsymbol{X}\approx\boldsymbol{T}_l^{\mathrm{T}}\boldsymbol{T}_l\boldsymbol{\Sigma}_l\boldsymbol{D}_l^{\mathrm{T}}=\boldsymbol{\Sigma}_l\boldsymbol{D}_l^{\mathrm{T}}\\ \boldsymbol{A}=\boldsymbol{T}_l^{\mathrm{T}}\boldsymbol{X}\approx\boldsymbol{\Sigma}_l\boldsymbol{D}_l^{\mathrm{T}}\end{cases} \tag{28}$$

式中:矩阵 $\boldsymbol{A}$ 中第 j 列元素表示第 j 个文本用 l 个特征项表示的向量,即 $\boldsymbol{A}$ 的列向量为各文本的向量表示。本文保留前 4 个最大的奇异值,即 $l=4$。

实验参数设置为种群规模 30,最大迭代次数 1 000,簇的个数 k 为 5,维数 d 为 4。

将 IHHO-KMedoids 算法与 K-Means、K-Medoids和 HHO-KMedoids 进行实验对比,在给定适应度函数上独立运行 30 次,记录其最好值、平均值、最差值、标准差和平均值的排序结果,实验结果见表 7,其中最好的结果用粗体表示。

表 7 4 种算法在文本数据集上的实验结果

Tab. 7 Experimental results by 4 algorithms on text datasets

算法	最好值	平均值	最差值	标准差	Rank
K-Means	45.211 4	49.649 9	66.166 1	6.897 8E+00	3
K-Medoids	45.356 7	49.358 9	62.602 7	5.309 0E+00	2
HHO-KMedoids	44.752 8	50.982 6	54.994 0	4.310 6E+00	4
IHHO-KMedoids	**44.664 8**	**44.664 8**	**44.664 8**	**1.213 3E−08**	**1**

由表 7 可知,IHHO-KMedoids 算法的实验结果均达到最优,其适应度的最好值、平均值和最差值都相同且为 44.664 8,标准差为 $1.213\ 3\times10^{-8}$。虽然 IHHO-KMedoids 与 HHO-KMedoids 的最好值比较接近,但这 2 种算法平均值的差异较大,分别为 44.664 8 和 50.982 6,同时 IHHO-KMedoids 的标准差达到最小,具有最好的稳定性。表 8 给出了上述 4 种算法在给定文本数据集上的评价指标结果。

表 8 4 种算法在文本数据集上的聚类评价指标结果

Tab. 8 Clustering evulation index results by 4 algorithms on text datasets

算法	F-Measure	ARI	NMI
K-Means	0.404 9	0.161 4	0.442 3
K-Medoids	0.455 0	0.292 3	0.436 6
HHO-KMedoids	0.428 7	0.198 9	0.485 3
IHHO-KMedoids	**0.517 6**	**0.355 6**	**0.548 3**

从表 8 可以看出 IHHO-KMedoids 算法聚类的 F-Measure、ARI 和 NMI 值分别达到 0.517 6、0.355 6 和 0.548 3,其聚类性能均优于其他对比算法,其中 K-Means 的 F-Measure 和 ARI 值最差,而 K-Medoids 的 NMI 值最差,表明了所提算法在该文本数据集上具有最好的聚类性能。总体上讲,IHHO-KMedoids 利用 IHHO 全局搜索能力好和 K-Medoids 不受离群点影响的优势,使得算法聚类效果好,稳定性高。

4 结 语

本文提出了一种基于 IHHO 与 K-Medoids 的混合聚类算法。在 IHHO 算法中,带有 Logistic 混沌扰动的控制参数策略平衡了算法的全局探索和局部开发能力。集成变异策略提高了算法的全局搜索能力,从而增强了算法的优化性能。翻筋斗觅食策略不仅增强了种群多样性,也避免了算法陷入局部最优、提高了收敛速度。IHHO 不仅保持了 HHO 较强的局部开发能力和易于实现的特点,而且利用各改进策略的优势,有效平衡了算法的探索和开发能力,具有较强的全局搜索能力。CEC2014 测试函数上的结果表明,IHHO 与 5 种其他群智能算法和 4 种改进 HHO 相比具有显著优势。将 IHHO 与 K-Medoids 进行结合并用于求解聚类问题,利用 IHHO 算法好的全局搜索性能解决了 K-Medoids 容易陷入局部最优的不足,而凭借 K-Medoids 效率高、鲁棒性强的特点降低了 K-Means 对离群点的敏感性。UCI 数据集和中文文本数据集上的实验结

果验证了本文所提 IHHO-KMedoids 的可行性。下一步将研究如何将其他优化策略与 HHO 结合来更进一步提高算法的寻优性能,并将改进 HHO 与其他聚类方法进行融合以解决一些实际的优化问题。

参考文献:

[1] YU Donghua, LIU Guojun, GUO Maozu, et al. An improved K-medoids algorithm based on step increasing and optimizing medoids[J]. Expert Systems with Applications, 2018, 92: 464-473.

[2] 赵烨, 黄泽君. 蚁群 K-Medoids 融合的聚类算法[J]. 电子测量与仪器学报, 2012, 26(9): 800-804.
ZHAO Ye, HUANG Zejun. Clustering algorithm based on fusion of ant colony algorithm and K-Medoids[J]. Journal of Electronic Measurement and Instrument, 2012, 26(9): 800-804.

[3] KHATAMI A, MIRGHASEMI S, KHOSRAVI A, et al. A new PSO-based approach to fire flame detection using K-Medoids clustering[J]. Expert Systems with Applications, 2017, 68: 69-80.

[4] MOKHTARZADEH M, TAVAKKOLI-MOGHADDAM R, TRIKI C, et al. A hybrid of clustering and meta-heuristic algorithms to solve a *p*-mobile hub location-allocation problem with the depreciation cost of hub facilities[J]. Engineering Applications of Artificial Intelligence, 2021, 98: 104121.

[5] HEIDARI A A, MIRJALLILI S, FARIS H, et al. Harris hawks optimization: algorithm and applications [J]. Future Generation Computer Systems, 2019, 97: 849-872.

[6] 王光义, 袁方. 级联混沌及其动力学特性研究[J]. 物理学报, 2013, 62(2): 020506.
WANG Guangyi, YUAN Fang. Cascade chaos and its dynamic characteristics[J]. Acta Physica Sinica, 2013, 62(2): 020506.

[7] ZHANG Hongliang, WANG Zhiyan, CHEN Weibin, et al. Ensemble mutation-driven salp swarm algorithm with restart mechanism: framework and fundamental analysis[J]. Expert Systems with Applications, 2021, 165: 113897.

[8] ZHAO Weiguo, ZHANG Zhenxing, WANG Liying. Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications[J]. Engineering Applications of Artificial Intelligence, 2020, 87: 103300.

[9] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//Proceedings of the 6th International Symposium on Micro Machine and Human Science. IEEE, 1995: 39-43.

[10] YANG Xinshe. A new metaheuristic bat-inspired algorithm[M]//Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Springer, 2010: 65-74.

[11] MIRJALLILI S, GANDOMI A H, MIRJALLILI S Z, et al. Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems[J]. Advances in Engineering Software, 2017, 114: 163-191.

[12] MIRJALLILI S, MIRJALLILI S M, HATAMLOU A. Multi-Verse Optimizer: a nature-inspired algorithm for global optimization[J]. Neural Computing and Applications, 2016, 27(2): 495-513.

[13] MIRJALLILI S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems, 2015, 89(1): 228-249.

[14] LI Chenyang, LI Jun, CHEN Huiling, et al. Memetic Harris Hawks Optimization: developments and perspectives on project scheduling and QoS-aware web service composition[J]. Expert Systems with Applications, 2021, 171: 114529.

[15] LIU Yun, CHONG Guoshuang, HEIDARI A A, et al. Horizontal and vertical crossover of Harris hawk optimizer with Nelder-Mead simplex for parameter estimation of photovoltaic models[J]. Energy Conversion and Management, 2020, 223: 113211.

[16] FAN Qian, CHEN Zhenjian, XIA Zhanghua. A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems [J]. Soft Computing, 2020, 24: 14825-14843.

[17] SONG Shiming, WANG Pengjun, HEIDARI A A, et al. Dimension decided Harris hawks optimization with Gaussian mutation: balance analysis and diversity patterns [J]. Knowledge-Based Systems, 2021, 215: 106425.

[18] 余冬华, 郭茂祖, 刘扬, 等. 基于距离不等式的 K-medoids 聚类算法[J]. 软件学报, 2017, 28(12): 3115-3128.
YU Donghua, GUO Maozu, LIU Yang, et al. K-medoids clustering algorithm based on distance inequality [J]. Journal of Software, 2017, 28(12): 3115-3128.

[19] BEL K N S, SAM I S. Black hole Entropic Fuzzy Clustering-based image indexing and Tversky index-feature matching for image retrieval in cloud computing environment [J]. Information Sciences, 2021, 560: 1-19.

[20] LOSSIO-VENTURA J A, GONZALES S, MORZAN J, et al. Evaluation of clustering and topic modeling methods over health-related tweets and emails[J]. Artificial Intelligence in Medicine, 2021, 117: 1-18.

[21] 胡强, 沈嘉吉, 荆广辉, 等. 基于描述语境特征词与改进 GSDMM 模型的服务聚类方法[J]. 通信学报, 2021, 42(8): 176-187.
HU Qiang, SHEN Jiaji, JING Guanghui, et al. Service clustering method based on description context feature words and improved GSDMM model[J]. Journal on Communications, 2021, 42(8): 176-187.

[22] 宗成庆, 夏睿, 张家俊. 文本数据挖掘[M]. 北京: 清华大学出版社, 2019.

(责任编辑 王绪迪)