一种基于改进果蝇优化的 K-mediods 聚类算法

王全民,杨 晶,张帅帅
(北京工业大学 信息学部,北京 100124)

摘 要: 经典的果蝇优化算法存在收敛精度不高、易陷入早熟收敛和局部最优的缺点,对此,提出一种新的果蝇优化算法。该算法在初始位置选取时利用混沌思想使初始值均匀分布在解空间中,算法后期收敛时,使用禁忌搜索跳出局部最优,避免早熟收敛。针对 K-mediods 聚类算法易陷入局部最优的缺点,将改进的果蝇优化算法与 K-mediods 聚类算法融合形成一种新的 K-mediods 算法,利用改进果蝇优化算法的全局寻优特点优化 K-mediods,使得算法可达到更好的聚类效果。在对比性实验中,采用标准优化测试函数验证改进的果蝇算法性能,结果表明改进的果蝇优化算法在寻优速度和精度上效果更优。在人工数据集与 UCI 数据集上对新的 K-mediods 算法与其他算法聚类效果进行比较,结果表明新的 K-mediods 算法在聚类准确率和效率上均有所提高,同时适用于高维数据的聚类。

关键词: 聚类; 果蝇优化算法; 混沌映射; FOA; 禁忌搜索; K-mediods
中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1673-629X(2018)12-0017-06
doi: 10.3969/j.issn.1673-629X.2018.12.004

A New K-mediods Clustering Algorithm Based on Improved Fruit Fly Optimization Algorithm

WANG Quan-min, YANG Jing, ZHANG Shuai-shuai
(School of Information, Beijing University of Technology, Beijing 100124, China)

Abstract: The classical fruit fly optimization algorithm has the disadvantages of low convergence accuracy, falling into premature convergence and local optimum easily. Therefore, we propose an improved fruit fly optimization algorithm. When the initial location is selected, the initial value is uniformly distributed in the solution space by using logistic mapping. When the algorithm converges at the later stage, Tabu search is used to jump out of the local optimum and avoid premature convergence. The improved fruit fly optimization algorithm is combined with K-mediods clustering algorithm with strong local search ability to improve the K-mediods algorithm's low clustering accuracy and easy falling into the local optimal solution, so as to achieve better clustering results. The improved fruit fly algorithm is tested by standard optimization test function through the targeted experiment. The results show that the improved fruit fly algorithm is superior to the classical FOA algorithm in the optimization precision and the optimization speed. Comparing the clustering results of the new K-mediods algorithm with other algorithms on the artificial data set and the UCI dataset, the results show that the new K-mediods algorithm improves both the accuracy and the efficiency of the clustering, which is suitable for the clustering of high-dimensional data.

Key words: clustering; fruit fly optimization algorithm; logistic mapping; FOA; Tabu search; K-mediods

0 引 言

K-mediods 算法是一种基于划分的聚类算法,改进了 K-means 算法对噪声和孤立点数据敏感的缺点,收敛速度较快且局部搜索能力较强[1]。但该算法聚类效率和精度较低、全局搜索能力较差等缺点仍需要改进[2]。文献[3]针对 K-medoids 算法对初始聚类中心敏感的缺陷,提出了有效的解决方案,提高了算法的性能。文献[4]提出了一种基于改进人工蜂群的 K-mediods 算法,由于结合粒计算和最大最小距离初始化蜂群降低了对噪声的敏感性,聚类效果较稳定。文献[5]提出了基于核的自适应聚类,该算法降低了 K-medoids 对初始值的敏感性。

针对 K-mediods 算法易陷入局部最优的缺点,提出了一种基于改进果蝇优化算法的 K-mediods 聚类

收稿日期: 2018-01-30 **修回日期:** 2018-05-30 **网络出版时间:** 2018-09-21
基金项目: 国家自然科学基金(61272500)
作者简介: 王全民(1963-),男,博士,副教授,硕导,CCF 高级会员(E200005398S),研究方向为网络与信息安全; 杨 晶(1992-),女,硕士研究生,研究方向为分布式系统和网络与信息安全。
网络出版地址: http://kns.cnki.net/kcms/detail/61.1450.TP.20180920.1536.040.html

?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

算法。将混沌映射思想应用在果蝇群体的初始位置生成,保证初始位置能够在全局更均匀的分布,算法后期根据群体适应度方差判断果蝇优化算法是否处于局部收敛状态,若是,利用禁忌搜索思想在最优值邻域内寻求更优解。将融入了混沌映射与禁忌搜索的果蝇优化算法与传统的K-mediods算法相结合,对每次聚类迭代中的簇中心进行寻优,以提高K-mediods算法的全局搜索能力和聚类精度。

1 基于Logistic映射的混沌技术

混沌映射的基本思想是:将优化变量(一个在[0,1]区间波动的变量)通过混沌映射规则映射到混沌变量空间的取值区间内,利用混沌变量的遍历性和规律性寻优搜索,最后将获得的优化解线性转化到优化空间[6]。通常其系统方程为:

$$x(t+1)=\mu x(t)(1-x(t)) \tag{1}$$

其中,μ为控制参数;t为迭代次数,$x(t)\in[0,1]$。当$\mu=4$时,系统出现混沌状态,即不重复地随机访问系统解空间。

式2为混沌变量Cx_i的一种变换算式。

$$Cx(t+1)_i=4Cx(t)_i(1-Cx(t)_i),\quad i=1,2,\cdots,N \tag{2}$$

其中,$Cx(t)_i$为第i个混沌变量经历i次混沌映射之后得到的变量值。当$Cx_i\in[0,1]$,$\mu=4$且$Cx_i\neq\{0.25,0.5,0.75\}$时,系统出现混沌状态。优化变量$x_i\in[x_{min},x_{max}]$,式3、式4为$x_i$与$Cx_i\in[0,1]$进行往返映射的过程。

$$Cx_i=(x_i-x_{min})/(x_{max}-x_{min}) \tag{3}$$

$$x_i'=x_{min}+Cx_i(x_{max}-x_{min}) \tag{4}$$

2 禁忌搜索算法

禁忌(Tabu search,TS)算法是一种启发式搜索算法,首先确定初始解,然后按照一定方向移动进行搜索试探。为了避免陷入局部最优的状况,在搜索过程中,禁忌搜索算法采用两个策略:禁忌准则和特赦准则[7]。禁忌搜索建立一个禁忌表,它有一定的长度,储存近期遍历过的最优解。根据邻域计算函数计算邻域内的最优值,查看禁忌表,以防近期遍历过的最优值重复遍历陷入局部最优,但若该解扩大了解空间的搜索范围,则特赦保留。

TS算法步骤如下:

(1)给定初始解x_0,计算$f(x_0)$,令当前点$x=x_0$,最优点$x_{best}=x_0$,最优值$f(x)_{best}=f(x_0)$。

(2)计算点x邻域内所有点的$f(x)$。

(3)寻找邻域内最优值best($f(x)$)的对应点x'。

(4)检查是否满足特赦准则,若是,则令$x=x'$,$x_{best}=x'$,$f(x)_{best}=f(x')$,执行步骤6;否则转入步骤5。

(5)判断禁忌准则:遍历禁忌表,若没有点x',则$x=x'$,执行转入步骤6,否则在邻域中删除x'。转入步骤3。

(6)更新禁忌表。判断是否满足终止条件,若满足,则终止计算,否则转入步骤2。

算法中禁忌准则是指通过建立“禁忌表”模拟记忆,设定合适的禁忌长度,将近期遍历过的解存放进去,将搜索到的最优解与之比较,以防搜索陷入循环,从而避免局部最优。特赦准则是指在禁忌搜索算法的迭代过程中,若被访问的解再次出现在禁忌长度内,但搜索范围可以得到很大的优化时,为了保障全局最优,可以接触对该最优值的禁止,进行重新选择。

3 果蝇优化算法(FOA)

果蝇优化算法是一种群智能全局优化搜索算法[8]。果蝇能够通过特殊的嗅觉和视觉搜集空气中的各种气味以锁定食物的位置,飞到食物位置附近后亦可使用敏锐的视觉发现食物和同伴聚集的位置,并且向该方向飞去[9]。

果蝇优化算法的一般步骤为:

(1)初始化参数:最大迭代次数maxgen,种群规模sizepop,果蝇群体位置X_axis、Y_axis。

(2)根据式5、式6,果蝇个体通过敏锐的嗅觉搜寻食物,并向随机方向移动。

$$X_i=X\_axis+Random\ Value \tag{5}$$

$$Y_i=Y\_axis+Random\ Value \tag{6}$$

其中,X_i和Y_i分别为果蝇个体飞向的横纵坐标。

(3)根据式7计算与原点的距离(Dist),根据式8计算得到味道浓度判定值S_i。

$$Dist_i=sqrt(X_i^2+Y_i^2) \tag{7}$$

$$S_i=1/Dist_i \tag{8}$$

(4)将味道浓度判定值S_i代入味道浓度判定函数即适应度函数,计算该果蝇个体位置处的味道浓度$Smell_i$。

$$Smell_i=Function(S_i) \tag{9}$$

(5)找出该果蝇群体中的最佳味道浓度值bestSmell。

$$[bestSmell\ bestIndex]=max(Smell) \tag{10}$$

(6)保留最优位置坐标和最优味道浓度值,果蝇群体利用视觉向该位置方向飞去。

$$\begin{cases}Smellbest=bestSmell\\X\_axis=X(bestIndex)\\Y\_axis=Y(bestIndex)\end{cases} \tag{11}$$

(7)进入迭代寻优,重复执行步骤2~5,判断味道

?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

浓度是否优于最佳味道浓度值,若是则执行步骤 6,直至当前迭代次数达到最大迭代数 maxgen 或已达到精度要求,则停止迭代。

4　基于混沌映射与禁忌搜索的果蝇混合优化算法(LGTS-FOA)

混沌映射思想保证在解空间范围不重复地访问所有状态,因此利用这一系统构造果蝇群体初始位置可以很好地使果蝇群体较为均匀地分布在解空间。而禁忌搜索思想对于初值有很高的要求,因此考虑在 FOA 算法的后期引入该算法。将后期收敛解作为 TS 算法的初始解,采用果蝇群体的适应度方差 σ^2 来判断后期收敛情况[10],当算法进入后期局部收敛状态时,利用禁忌搜索算法跳出局部最优。

4.1　LGTS-FOA 算法

LGTS-FOA 算法步骤如下:

(1) 设置种群规模 sizepop、最大迭代次数 MCN 和适应度方差阈值 δ。

(2) 利用 logistics 多次映射的结果作为果蝇种群的初始位置。

(3) 执行 FOA 算法中的步骤 2 ~ 5。

(4) 利用式 13 得到当前迭代中种群的适应度方差。

$$\mathrm{Smell}_{\mathrm{avg}} = \sum_{i=1}^{\mathrm{sizepop}} \mathrm{Smell}_i / \mathrm{sizepop} \tag{12}$$

$$\sigma^2 = \sum_{i=1}^{\mathrm{sizepop}} (\mathrm{Smell}_i - \mathrm{Smell}_{\mathrm{avg}})^2 \tag{13}$$

(5) 若 $\sigma^2 < \delta$,说明算法已进入后期收敛状态,则执行禁忌搜索算法的步骤 2 ~ 6。否则执行步骤 6。

(6) 如果当前迭代得出的 bestSmell 优于 Smellbest,保存最佳浓度和最优位置坐标。判断是否达到最大迭代次数,若是结束算法,否则,转入步骤 3。

4.2　有效性测试

文中采用 4 个经典标准函数进行有效性测试:

(1) Sphere 函数:

$$f_1(x) = \sum_{i=1}^{D} x_i^2$$

(2) Rosenbrock 函数:

$$f_2(x) = \sum_{i=1}^{D} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

(3) Griewank 函数:

$$f_3(x) = \frac{1}{4\ 000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

(4) Rastrigin 函数:

$$f_4(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos 2\pi x_i + 10)$$

实验参数设置: 群体规模 sizepop = 30,最大迭代次数 maxgen = 1 000,适应度方差 δ = 0.000 01。用 4 个经典标准函数分别对改进的 LGTS-FOA 算法和经典 FOA 算法进行测试,在相同的实验条件下,各运行 50 次,比较二者的最优值(best)、优化均值(median)和标准差(std),结果见表 1。

表 1　FOA 与 LGTS-FOA 算法性能比较

函数	FOA			LGTS-FOA		
	最优值	优化均值	标准差	最优值	优化均值	标准差
f_1	4.526 4e-6	8.266 0e-4	0.001 2	1.659 9e-36	1.376 5e-35	1.125 9e-38
f_2	28.652 6	28.264 8	0.109 0	27.132 1	28.623 5	0.056 2
f_3	9.362 2e-8	2.860 1e-6	4.895 9e-6	1.028 5e-13	1.036 1e-13	1.465 2e-35
f_4	3.135 2e-5	3.569 2e-5	2.612 6e-6	1.456 5e-14	1.524 2e-14	6.146 5e-16

从表 1 可以得出,在同等条件下,对于单峰函数(f_1、f_2)和多峰函数(f_3、f_4)来说,LGTS-FOA 算法在均值和标准差上较经典 FOA 算法都不同程度的改善,优化均值精度提高了 8 ~ 9 个数量级,其中 f_1 函数甚至提高 30 个数量级,说明 LGTS-FOA 在寻优精度上有显著提高; 标准差的降低,说明 LGTS-FOA 比 FOA 的稳定性要强; 计算多峰函数 Griewank 函数和 Rastrigin 函数最优值上 LGTS-FOA 算法的寻优精度数量级提高了 10 个左右,说明其在高维多峰函数上的效果显著。

观察 4 个经典标准函数迭代 1 000 次的果蝇群体适应度进化过程。为方便显示,纵坐标选取以 10 为底适应度的对数。在 4 个标准函数上经典 FOA 算法表现为迭代前期收敛速度较快,之后陷入局部最优,从而导致寻优精度不够高,而 LGTS-FOA 在保证了收敛速度的情况下,在陷入局部最优时能够迅速跳出,继续寻优,精度数量级有大幅提高。在 Rosenbrock 函数的优化过程中虽然两者精度都不高,但是 FOA 迭代初期就陷入局部最优,寻优速度减缓; 而 LGTS-FOA 在迭代过程中,寻优速度较快,LGTS-FOA 性能总体优于 FOA。在 Griewank 函数、Rastrigin 函数优化过程中,由于初始位置采用混沌映射生成,因此适应度的起始值要优于 FOA,有利于帮助 LGTS-FOA 更快速精确的寻优。综上所述,LGTS-FOA 的适应度收敛速度和精度均优于 FOA。

?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

5 K-mediods 聚类算法

K-mediods 聚类算法的出现是对经典 K-means 算法的一种改进，避免了 K-means 算法对噪声和孤立点的敏感程度[11]。典型的 K-mediods 算法步骤如下[2]：

(1) 随机选取 k 个样本作为初始聚类中心点集。

(2) 依次计算数据集中所有样本点到中心点的距离(欧几里德距离)，将样本点放入距离最近的中心点代表的簇中。

(3) 在各簇中，计算与簇内各样本点距离的绝对误差最小的点，替代旧中心点。

(4) 如果聚类中心点集保持不变，算法终止；否则，更新中心点，重复执行步骤 2～4。

6 基于改进果蝇优化算法的 K-mediods 聚类算法

6.1 适应度函数

采用类内距离衡量聚类的内聚程度[12]，公式如下：

$$D_i = \sum_{j=1}^{k} \sum_{p \in C_j} | p - z_{ij} |^2 \tag{14}$$

其中，z_{ij} 表示第 i 个蜜蜂所表示的第 j 个聚类 C_j 的中心点；p 表示类 C_j 中的非聚类中心点。

文中将类内距离的倒数作为适应度值。当类内距离最小时聚类效果最优，此时适应度值最大，表示如下：

$$\mathrm{fit}_i = 1/D_i \tag{15}$$

6.2 新聚类中心

每次迭代果蝇群体寻找出的最优解并不能保证落到某个样本坐标上，因此需要确定新的聚类中心 v_{ij}，表示距离聚类中心最近的样本[13]，如式(16)。

$$v_{ij} = \{ x_t \mid \min_{t=1}^{n} | x_t - z_{ij} | \} \tag{16}$$

其中，$i = 1,2,\cdots,\mathrm{SN}$，$j = 1,2,\cdots,k$；$n$ 表示数据集个数；x_t 表示果蝇个体；z_{ij} 表示第 i 只果蝇的第 j 维(聚类中心)。

6.3 算法步骤

6.3.1 果蝇编码

初始化种群每一个体对应一组聚类中心向量 $\boldsymbol{Z}_i = (z_{i1}, z_{i2}, \cdots, z_{ik})$，其中 z_{ik} 表示第 i 个果蝇表示的第 k 簇的中心向量。

6.3.2 具体步骤

(1) 初始化聚类个数 k，群体规模 sizepop，迭代数 maxgen，味道浓度方差阈值 δ。

(2) 利用 Logistic 映射产生的混沌序列作为初始果蝇群体的位置编码，利用式 16 将分别距离序列最近的 k 个样本作为初始聚类中心点。

(3) 根据欧几里得距离计算，将剩余样本加入到距离最近的聚类中心所代表的簇中。

(4) 执行 LGTS-FOA 算法步骤 3～6，适应度函数如式 15 所示。

(5) 将此时最佳适应度位置编码利用式 16 得出新的聚类中心。

(6) 对数据集进行一次 K-mediods 聚类，得到新的聚类中心，更新果蝇位置。

(7) 若达到最大次数 Maxgen 或者聚类中心收敛，则停止算法；否则转入步骤 3，$M = M + 1$。

7 实验结果与分析

为了验证文中算法的有效性和可行性，分别采用 K-mediods、文献[14-15]提出的算法及文中算法，分别在随机生成的人工数据集、Iris、Wine 和 Seeds 数据集上进行测试。在 10 次实验中，参数设置分别为：蜂群个数 sizepop = 30，最大循环次数 Maxgen = 100，δ = 0.000 01。

7.1 人工数据集实验

随机生成的人工数据集属性维度为 2，样本个数为 300，类个数为 3。人工数据集的样本分布如图 1 所示，算法在人工数据集上的聚类结果分别为图 1～5 所示。

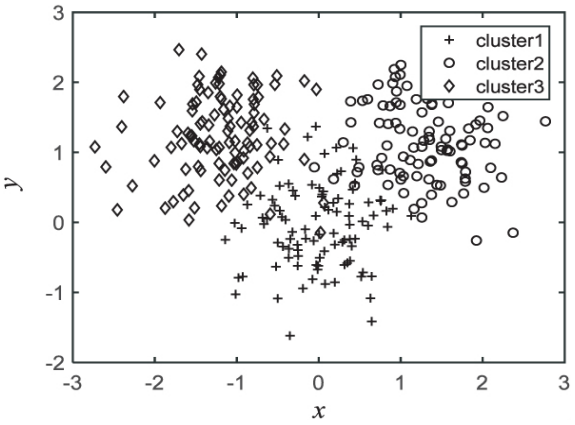

图 1 人工数据集分布

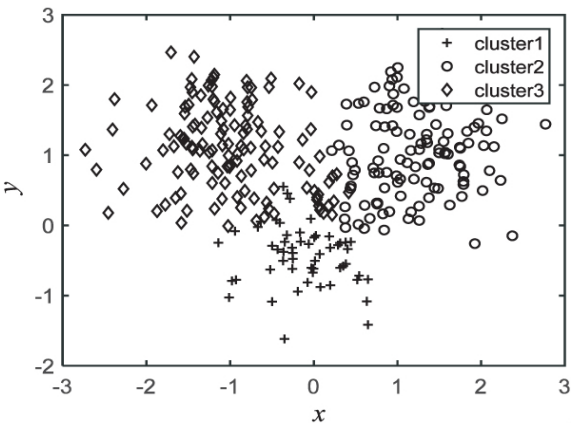

图 2 K-mediods 聚类结果

?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

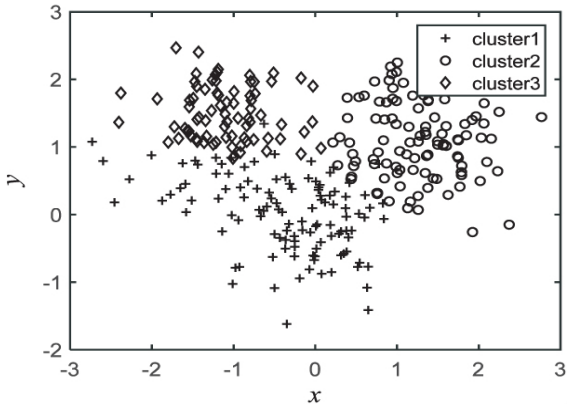

图 3　文献 [14] 算法的聚类结果

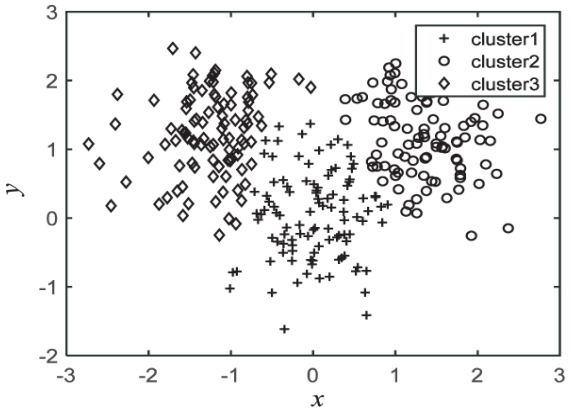

图 4　文献 [15] 算法的聚类结果

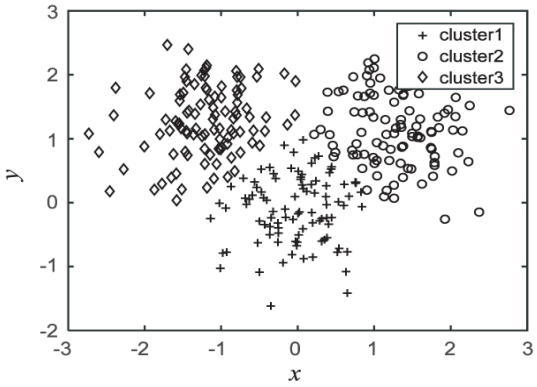

图 5　文中算法的聚类结果

实验结果表明,K-mediods、文献 [14-15] 的算法及文中算法在人工数据集上的聚类准确率分别为 79.13%、82.56%、89.41%、93.79%。相比其他几种算法,文中算法有较好的聚类效果,不但聚类准确率较高而且聚类效果稳定。通过比较图 2 ~ 5 可知,文中算法对类边界的处理最为精确。其他算法在边界处理上出现较大偏差导致聚类准确度不够高,与原始数据样本分布有较大偏差。

7.2　UCI 标准数据集实验

采用 UCI 标准数据集中的 Iris、Wine 和 Seeds。各类算法在上述几种标准数据集中得到的平均正确率以及迭代次数如表 2 所示。

表 2　不同算法对 UCI 数据集聚类结果的比较

算法	Iris		Wine		Seeds	
	平均正确率/%	平均迭代次数	平均正确率/%	平均迭代次数	平均正确率/%	平均迭代次数
K-mediods	76.52	5.63	54.26	10.23	50.63	9.45
文献 [14]	89.12	4.12	70.56	4.52	64.16	7.28
文献 [15]	94.25	2.65	73.12	4.06	71.21	4.51
文中算法	96.58	3.46	78.41	3.26	73.32	4.21

通过比较表 2 发现,文中算法在迭代次数上与其他几种算法相比有所减少,说明了采用映射思想初始化果蝇群体时,能够将初始值较为均匀地分散在解空间,避免了初始聚类中心分布不理想对 K-mediods 算法的影响,改善了 K-mediods 算法对初始值敏感的缺点,从而减少了迭代次数; 使用禁忌思想能够避免陷入局部最优解,提高聚类正确率; 在 Iris 数据集上表现尤为明显,聚类最高正确率达到 97%,平均准确率比 K-mediods 提高 20%。相比 Iris,文中算法在 Wine 和 Seeds 数据集上的平均准确率相对较低,但其平均正确率和平均迭代次数仍然优于其他几种算法,同时实验数据也表明基于 LGTS-FOA 的 K-mediods 算法同样适用于高维数据集的聚类研究。可见,文中算法在聚类准确率、效率以及稳定性等方面有明显的改进。

综上所述,对不同的数据集,在相同的实验条件下,相比其他几种算法,文中将改进的果蝇算法和 K-mediods 相结合,改善了传统 K-mediods 算法易陷入局部最优的缺点,缓解了其对初始值的敏感,在聚类准确度和效率上均有所提高,对于高维数据集同样具有较好的适应性。

8　结束语

提出一种基于改进果蝇优化算法的 K-mediods 聚类算法,将融合了混沌映射与禁忌搜索思想的果蝇优化算法与 K-mediods 算法相结合,使用类内距离的倒数作为适应度函数,在人工数据集和 UCI 数据集上进行实验,结果表明文中算法既能保证聚类准确度又能有效提高效率,且对于高维数据集具有较好的适应性。

?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

参考文献:

[1] XIE Juanying, JIANG Shuai. A simple and fast algorithm for global k-means clustering [C] //Second international workshop on education technology and computer science. Wuhan, China: IEEE, 2010: 36-40.

[2] ALKOFFASH M S. Automatic arabic text clustering using k-means and k-mediods [J]. International Journal of Computer Applications, 2012, 51(2): 5-8.

[3] 潘 楚,罗 可. 基于改进粒计算的 K-medoids 聚类算法 [J]. 计算机应用, 2014, 34(7): 1997-2000.

[4] 孙立新,张栩之,邓先瑞,等. 自适应果蝇算法优化模糊均值聚类算法图像分割 [J]. 控制工程, 2016, 23(4): 494-499.

[5] 孙 胜,王元珍. 基于核的自适应 k-medoid 聚类 [J]. 计算机工程与设计, 2009, 30(3): 674-675.

[6] 程 慧,刘成忠. 基于混沌映射的混合果蝇优化算法 [J]. 计算机工程, 2013, 39(5): 218-221.

[7] 印 溪,许 斌,亓 晋. 一种基于禁忌策略的混合优化算法 [J]. 计算机技术与发展, 2017, 27(2): 46-50.

[8] PAIL W T. A new fruit fly optimization algorithm: taking the financial distress model as an example [J]. Knowledge-Based Systems, 2012, 26: 69-74.

[9] DU Tingsong, KE Xianting, LIAO Jiagen, et al. DSLC-FOA: an improved fruit fly optimization algorithm application to structural engineering design optimization problems [J]. Applied Mathematical Modelling, 2017, 55: 314-339.

[10] 韩俊英,刘成忠. 自适应混沌果蝇优化算法 [J]. 计算机应用, 2013, 33(5): 1313-1316.

[11] VERMA J, RICHHARIYA V. A review: salient feature extraction using k-mediods clustering technique [J]. Asian Journal of Computer Science & Information Technology, 2013, 2(3): 11-19.

[12] 王宏智,高学东,赵 杨. 一种群体智能聚类算法研究 [J]. 中国管理信息化, 2013, 16(2): 74-75.

[13] 李 莲,罗 可,周博翔. 一种改进人工蜂群的 K-medoids 聚类算法 [J]. 计算机工程与应用, 2013, 49(16): 146-150.

[14] 马 菁,谢娟英. 基于粒计算的 K-medoids 聚类算法 [J]. 计算机应用, 2012, 32(7): 1973-1977.

[15] 姚丽娟,罗 可,孟 颖. 一种基于粒子群的聚类算法 [J]. 计算机工程与应用, 2012, 48(13): 150-153.

(上接第 16 页)

[4] PACINI E, MATEOS C, GARINO C G. Distributed job scheduling based on swarm intelligence: a survey ☆ [J]. Computers & Electrical Engineering, 2014, 40(1): 252-269.

[5] LI Kun, XU Gaochao, ZHAO Guangyu, et al. Cloud task scheduling based on load balancing ant colony optimization [C] //Sixth annual china grid conference. [s. l.]: IEEE, 2011: 3-9.

[6] 左利云,左利锋. 云计算中基于预先分类的调度优化算法 [J]. 计算机工程与设计, 2012, 33(4): 1357-1361.

[7] 张春艳,刘清林,孟 珂. 基于蚁群优化算法的云计算任务分配 [J]. 计算机应用, 2012, 32(5): 1418-1420.

[8] 查英华,杨静丽. 改进蚁群算法在云计算任务调度中的应用 [J]. 计算机工程与设计, 2013, 34(5): 1716-1719.

[9] 王 芳,李美安,段卫军. 基于动态自适应蚁群算法的云计算任务调度 [J]. 计算机应用, 2013, 33(11): 3160-3162.

[10] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.

[11] TAWFEEK M A, EL-SISI A, KESHK A E, et al. Cloud task scheduling based on ant colony optimization [C] //8th international conference on computer engineering & systems. Cairo, Egypt: IEEE, 2013: 64-69.

[12] 袁亚博,刘 羿,吴 斌. 改进蚁群算法求解最短路径问题 [J]. 计算机工程与应用, 2016, 52(6): 8-12.

[13] 黄 俊,王庆凤,刘志勤,等. 基于资源状态蚁群算法的云计算任务分配 [J]. 计算机工程与设计, 2014, 35(9): 3305-3309.

[14] CALHEIROS R N, RANJAN R, BELOGLAZOV A, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms [J]. Software Practice & Experience, 2011, 41(1): 23-50.

?1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net