改进混合蛙跳算法和 K-Means 的新型聚类算法①

卞艺杰, 吴 慧, 邹银马, 马瑞敏

(河海大学 商学院, 南京 211100)

摘 要: 研究针对现有聚类算法存在着精度较低, 易陷于局部最优等问题, 提出一种改进的混合蛙跳算法和 K-Means 相结合的新型聚类算法 ISFLA-K, 该算法使用对立学习的思想产生初始种群, 根据蛙自身具有认知能力和学习能力的特性对混合蛙跳算法的蛙跳规则进行改进, 即形成 ISFLA, 最后使用 ISFLA 优化 K-Means 聚类算法, 提高求解精度. 实验结果表明, ISFLA-K 具有很好的聚类性能, 求解精度高.

关键词: 混合蛙跳算法; K-Means 算法; ISFLA-K

New Clustering Algorithm Based on Improved Shuffled Frog Leaping Algorithm and K-Means Algorithm

BIAN Yi-Jie,WU Hui, ZOU Yin-Ma, MA Rui-Min

(Business School, Hohai University, Nanjing 211100, China)

Abstract: Existing clustering algorithms have the problems of low precision and easy to fall into local optimum. The paper proposes a new algorithm—ISFLA-K, which combined with an improved shuffled frog leaping algorithm and K-Means clustering algorithm. The algorithm uses the idea of an independent study to generate the initial population. According to the frogs' characteristics of cognitive and learning ability, it improve the rules of shuffled frog leaping algorithm leapfrog. The paper uses ISFLA to optimize K-Means clustering algorithm, which improved solution accuracy. The experimental results can prove the validity and superiority of the proposed algorithm.

Key words: shuffled frog leaping algorithm; K-means; ISFLA-K

随着 K-Means 算法作为一种经典的聚类算法, 广泛应用于数据挖掘和知识发现领域, 但是 K-Means 算法的聚类精度往往受到初始聚类中心选取的影响, 进而易陷入局部最优. 专家学者把 ACO、PSO 等群智能优化技术引入到 K-Means 算法中, 实验结果[1,2,3]表明, 这些进化算法在一定程度上可以解决陷入局部极值的问题, 但在此过程中也有可能产生退化现象, 导致 K-Means 算法迭代次数过多, 聚类准确性不高. 混合蛙跳算法(SFLA)作为一种全新的生物进化算法, 它不但具有粒子群算法较强的局部寻优能力, 还具有遗传算法的全局寻优能力. SFLA 概念比较简单, 调整参数少, 计算速度快, 易于实现. 因此, 本文提出一种改进的混合蛙跳算法和 K-Means 相结合的聚类算法 ISFLA-K, 运用改进后的 SFLA 算法的优点去弥补 K-Means 算法对初始聚类中心敏感常陷于局部极值的不足, 从而提高聚类算法的精度值.

1 K-Means聚类算法介绍

K-Means 算法是把 u 个数据对象分为 k 个簇, 使各聚类中的数据点到该聚类中心的距离平方和最小, 使得各簇内的对象具有较高的相似性, 而各簇间相似性较低. 算法首先任意选择 k 个点作为 k 个初始聚类的中心或均值, 其次计算剩余的数据点到各个聚类中心的距离, 按最近距离原则将数据点分配到离它最近的聚类中心所在的类. 然后使用平均值方法再对调整后的新类计算新的聚类中心(所有对象的均值), 如

① 收稿时间:2013-11-12;收到修改稿时间:2013-12-06

果相邻两次的聚类中心没有任何变化或聚类准则函数 E 已经收敛，说明数据对象调整结束. 该算法采用的是迭代更新的方法，在每一次迭代过程中均考察每个数据的划分是否正确，若不正确，就要调整. 在全部数据调整完后，再修改聚类中心，进入下一次迭代. 如果在一次迭代算法中，所有数据对象被正确划分，则不会有调整，聚类中心也不会再产生变化，这标志着 E 已经收敛，此时算法结束，输出结果为最后产生的 k 个聚类中心点和以其为中心的聚类划分. 衡量该算法是否正确的依据是随着算法的迭代次数增多 E 的值在不断减小，最终收敛至一个固定值.

K-Means 算法描述如下：输入为含有 u 个数据元素的数据集合，聚类个数 k;输出为 k 个聚类. Step1：从数据集合中随机选取 k 个初始聚类中心. Step2：计算所有数据元素到各个聚类中心的距离，并将这些数据元素置于距离最近的聚类中. Step3：一次分配完成后，重新计算 k 个聚类中心. Step4：与前一次的聚类中心进行对比，如果聚类中心位置改变，则转向 Step2，否则转向 Step5. Step5：输出聚类结果.

虽然 K-Means 算法对于大数据集可伸缩性较好，同时直观、易理解、聚类效率高，但是该算法存在着很大的局限性，它需要根据感觉和经验给出初始的簇的个数 k，对噪声和孤立点敏感，初始聚类中心随机生成，对聚类结果影响较大，使得算法常陷入局部最优.

2　混合蛙跳算法介绍

混合蛙跳算法作为群体智能优化算法中一种高效并行全局的搜索方法，已在函数优化、工业系统优化与控制、图像处理等方面得到广泛的应用. 混合蛙跳算法中，青蛙群体由多只青蛙组成，被分为多个子种群(族群)，每个族群包含一定数量的青蛙. 不同族群有不同的内部思想，在族群内每只青蛙个体也都有自己的思想，并影响着其它青蛙个体，且根据元进化策略而执行局部搜索，在各个族群内进行思想交流，局部搜索使信息在局部个体间进行传递. 当进化达到定义的局部搜索迭代次数以后，局部思想在各个族群间进行思想交流而实现了族群间的混合运算，执行混合策略使得局部间的思想在整个群体中得到交换. 重复执行这一过程直到满足所设置的收敛条件为止.

SFLA 按照蛙群分类进行信息传递，它有效地将全局信息交互与局部进化搜索相结合，来完成对问题的求解过程，其中蛙被定义为问题的一个解.

SFLA 的基本过程[4]：随机初始化产生 N 只蛙组成初始种群 $P=\{X_1, X_2, \ldots X_N\}$，第 K 只蛙表示问题的解为 $X_k=\{X_{k1}, X_{k2}, \ldots X_{kS}\}$，其中 S 表示解空间的维数，首先将群体内的蛙按适应度值大小进行降序排列，同时记录蛙群中具有最优适应度的蛙为 X_g，之后将整个蛙群分为 m 个子群，每个子群中包含 n 只蛙，m 和 n 满足 $N=m\times n$. 将蛙按公式 3-1 平均分配到 m 个子群中，即第 1 只蛙分入第 1 子群，第 2 只蛙分入第 2 子群，第 m 只蛙分入第 m 个子群，第 m+1 只蛙重新分入第 1 子群中，依次类推，直至所有蛙分配完毕. 设 F^i 为第 i 个子群的蛙的集合，其分配过程的表达式如下：

$$F^i=\{X_{i+m(l-1)}\in p \mid 1<l<n\} \quad 1<i<m \qquad (1)$$

对于每一个子群，X_b、X_w 分别代表最好适应度的蛙和最差适应度的蛙，接着对蛙的每一个子群执行局部搜索，即对子群中的 X_w 进行迭代更新操作，更新规则为：

$$D=r\times(X_b-X_w) \qquad (2)$$

$$X_w=X_w+D, \quad D\le D_{max} \qquad (3)$$

其中，r 表示 0 到 1 之间的一个随机数，D 表示蛙的移动步长，D_{max} 表示蛙的最大移动步长. 在经过更新后，如果得到的蛙 X_w 优于原来的蛙 X_w，则取代原子群中的蛙. 否则，用 X_g 取代 X_b，按公式 2 和公式 3 重新执行局部搜索规则. 若仍然没有改进，那么随机产生一个新蛙替换原来的 X_w，不断执行以上操作，当局部搜索达到预设的迭代次数 L_{max} 时，将所有蛙重新进行混合排序和划分子群，进行下一轮局部搜索，如此反复，直到满足终止条件，即代表最好解的蛙不再改变位置或者达到混合迭代次数 G_{max}. 这种全局信息交换和局部深度搜索的平衡策略使得算法可以跳出局部极值点，向全局最优方向发展.

3　改进混合蛙跳算法的方案

不少学者对 SFLA 的函数优化和更新等方面的研究较多[5-7]，但对于初始化蛙的群体研究并不多，一般都是随机生成，具有很大的盲目性，性能不高. 对于生物种群中的蛙而言，它也有一定的认知能力和学习能力，它不仅可以向个体领域最优值学习，即向自己的历史最好值学习，还可以向群内其它的蛙学习. 本文主要从初始化蛙群和改进蛙跳更新规则两个方面对混合蛙跳算法进行改进.

3.1 初始化蛙群

对于进化算法而言，高质量的初始解可以使得算法快速找到最优解，节约搜索时间，提高算法的精度. 所以针对混合蛙跳算法的初始解总是随机产生的，本文引入对立学习的策略提高初始化蛙群的质量. 对立学习策略的基本思想是：在寻找最优解时，既要考虑当前点的解，又要考虑其对立点的解.

对立点定义如下：假设 $O(e_1,e_{2,},...,e_z)$ 是 Z 维空间中的一个点，其中，a_i 和 b_i 分别为 e_i 的下界和上界. 则对立点为 $O'(e'_1,e'_2,...,e'_z)$，其中:

$$e'_1 = a_i + b_i - e_1, i = 1,2,..., z \quad (4)$$

将它引入到混合蛙跳算法的实现过程为：随机产生初始蛙群，利用公式 4 计算蛙的相对群体，然后根据蛙的适应度值大小从初始蛙群和蛙的相对群体中选择 N 只蛙作为新的初始蛙群.

3.2 改进蛙跳更新规则

在传统的 SFLA 中，蛙是根据子群内最优的个体或者是整个群体内最优的个体来调整位置的，但是在进化的过程中，蛙自身具有认知能力和学习能力，因此本文提出的改进的蛙的更新规则主要基于蛙的以下特点:

(1)蛙可以延续上次更新的步长，即记住前一次的更新步长值，同时向自己的历史最好值学习.

(2)蛙不仅会向子群内的最优蛙学习，还应该与子群内的其它蛙交流信息.

基于以上特点，本文提出一种新的蛙跳更新规则:

$$Dis(t+1) = W(R_1 His(P_w) + R_2 Dis(t)) + R_3(P_w(t) - P_1(t)) + R_4(P_b - P_w)$$

条件 $Dis \le \mathrm{Dis}M$ (5)

$$\mathrm{P}_w(t+1) = P_w(t) + Dis(t+1) \quad (6)$$

$$\mathrm{W} = \mathrm{W}_{\max} - \frac{t(W_{\max} - W_{\min})}{T_{mem}} \quad (7)$$

$\mathrm{R}_1, \mathrm{R}_2, \mathrm{R}_3, \mathrm{R}_4$ 表示 0~1 之间的随机数; P_w 表示迭代过程中适应度值最差的蛙，P_b 是子群内的最好个体，P_1 表示在同一子群内其它随机产生的蛙，即同一子群内与 P_w 交流信息的蛙;Dis 是 P_w 的移动步长;t 是当前迭代次数, $His(P_w)$ 是 P_w 的历史最优值， DisM 是蛙移动步长的上限值;w 为调节搜索过程的权重因子，实验表明，当 w 取值较大时，有助于全局最优解的搜索，当 w 较小时，有助于局部最优解的搜索，如公式 7 所示，算法首先设置较大的 w，然后逐渐减少，$W_{\max}$ 和 $W_{\min}$ 为初始权重值和最终权重值，分别取值为 0.9 和 0.4，这也是目前很多相关研究当中所设定的值; T_{mem} 为当前子群的最大迭代值. 相比公式 2 和公式 3，公式 5~公式 7 这种改进的蛙跳更新规则不仅包含了对过去经验的记忆，还具备了向其它蛙学习的能力，因此具有更强的寻优能力.

4 ISFLA-K聚类算法设计

将混合蛙跳算法应用到 K-Means 算法中，需要解决两个问题：一是如何对每只蛙进行构造或编码;二是如何设计适应度函数去衡量每只蛙的聚类效果.

4.1 蛙的构造方案

K-Means 聚类算法的核心是寻找最优聚类中心，而混合蛙跳算法中每只蛙就是一个候选解，代表一组聚类中心的集合. 即每只蛙由 q 个聚类中心组成，因而第 S 只蛙的编码可以构造如下: $\mathrm{X}_s = \{\mathrm{X}_{s1,}\mathrm{X}_{s2,},....\mathrm{X}_{sq,}\}$

4.2 蛙的适应度函数设计

SFLA 使用群体中蛙的适应度值来进行搜索和寻找最优解的，所以蛙的适应度函数的选择尤为重要，本文定义其目标函数为 I，公式如下:

$$I = \frac{\sum_{i=1}^{k}\left(\frac{1}{\partial}\sum_{x_j \in C_i}\|X_j - Z_i\|\right)}{k} \quad (8)$$

式中，C_i 为第 i 个聚类，X_j 为对应聚类中的数据元素，Z_j 为聚类中心. ∂ 为每个聚类内的数据元素的数量. I 越小表示聚类划分质量越好，因此为了取得较好的聚类效果，构造适应度函数如下:

$$f = \frac{1}{I} \quad (9)$$

4.3 ISFLA-K 聚类算法描述

结合改进的混合蛙跳算法和 K-Means 算法，本文提出的 ISFLA-K 算法流程描述如下:

Step1: 初始化参数：蛙的总群体数 N_{sum}，子群个数 n，每个子群内的蛙数 m，满足 $N = \mathrm{m} \times n$，全局搜索次数 T_{glo}，每个子群的搜索迭代次数 T_{mem}，初始权重值 $W_{\max}$ 和最终权重值 $W_{\min}$，k 个聚类中心.

Step2: 初始化群体：利用公式 4 的对立策略初始化蛙的群体，产生 N 只蛙.

Step3: 对产生的每只蛙执行 K-Means 操作，将蛙的编码作为初始的 K 个聚类，按照 K-Means 算法的最近邻法则更新聚类中心，直到满足算法要求，具体聚类过程前面已讲述.

Step4: 划分子群: 利用公式 8 和 9 计算每只蛙的适应度函数值, 按适应度大小对 N 只蛙进行降序排列并找到全局最优解 P_g. 将蛙分为 n 个子群, 每个子群里包含 m 只蛙, 确定各个子群的最优解 P_b 和最差解 P_w.

Step5: 执行子群局部搜索: 按公式 5、6 和 7 更新每一次迭代得到的解, 如果得到的解 $P_w(t+1)$ 优于前一次的解 $P_w(t)$, 那么 $P_w(t+1)$ 取代 $P_w(t)$, 否则用 P_g 代替 P_b, 按公式 5 、6 和 7 重新进行更新, 如果仍然没有改进, 则随机生成一只蛙, 并计算其适应度值, 当达到迭代次数 T_{mem} 后局部搜索完成.

Step6: 执行全局搜索: 当子群全部进化完成后, 混合所有蛙, 重新进行子群划分, 即转向 Step4. 当达到预先设定的最大迭代次数 T_{glo} 时, 迭代结束, 输出最优解.

5 ISFLA-K算法测试与分析

5.1 实验环境与参数选择

实验采用 UCI 数据库[8]中的 Lenses, Iris 和 Wine 三组数据集进行验证. 三组数据集的相关介绍如表 1 所示. 实验基于 K-Means 聚类算法、基于混合蛙跳的 K-Means 聚类算法(SFLA-K)和 ISFLA-K 对三组数据集进行聚类结果的对比. 试验的硬件平台配置为 Intel Pentium R 2.1GHz 处理器, 4G 内存, 500G 硬盘;操作系统为 Windows7 旗舰版;测试所用算法在 MATLAB 环境下实现.

表 1　Lenses, Iris 和 Wine 三组数据集介绍

数据集	数据来源	数据维数	数据点个数	聚类个数
Lenses	透镜数据集	4	24	3
Iris	莺尾类植物数据集	4	150	3
Wine	酒类识别数据集	13	178	3

在 SFLA 中有 5 个基本参数: 蛙的数量、子群个数、步长值、子群内迭代次数和混合迭代次数. 为了保证算法执行的效率和寻找最优解的准确率, 本实验经过多次反复试验, 同时参考文献[9]中建议的参数设置, 最终取蛙的总数为 150, 子群数为 5, 每个子群中蛙的数量为 30, 子群内迭代次数为 50, 混合迭代最大次数为 600, 当最大步长值取 0.75 时, 实际求解结果最为理想, 初始权重和最终权重分别设置为 0.9 和 0.4.

5.2 实验设计与结果分析

在聚类分析中, 聚类效果往往是由类内距离和类间距离决定, 要求有较小的类内距离, 较大的类间距离[9]. 实验的评价指标由适应度值, 聚类内距离和聚类间距离组成, 最理想的聚类结果是有较大的适应度值, 尽可能小的聚类内部距离, 尽可能大的聚类之间距离.

(1) 类内距离

定义 1. 假设数据对象分为 K_t 个聚类, 定义类内距离为每个聚类内的数据对象到对应聚类中心的距离之和.

$$\mathrm{int}raD\_1=\sum_{i=1}^{K_t}\sum_{p\in C_i}\|l-m_i\| \quad (10)$$

式中, $\mathrm{int}raD\_1$ 为类内距离, l 为空间内的数据对象, m_i 为聚类簇 C_i 的聚类中心, 即 C_i 所含数据对象的均值.

定义 2. 假设数据对象分为 K_t 个聚类, 定义每个聚类内的数据对象与同一聚类内所有其它数据对象之间的平均距离的最小值作为本聚类的类内距离, 而整个数据空间的类内距离则为所有聚类中类内距离的最大值.

$$\mathrm{int}raD\_2=\max_{1\le j\le K_t}\left\{\min_{1\le\alpha\le|C_j|}\left\{\frac{1}{|C_j|}\sum_{\beta=1,\alpha\ne\beta}^{|C_j|}\|m_\alpha-m_\beta\|\right\}\right\} \quad (11)$$

式中, $\mathrm{int}raD\_2$ 为类内距离, $|C_j|$ 为聚类簇 C_j 内的数据对象的个数, m_α, m_β 为 C_j 的对象.

(2) 类间距离

定义 3. 假设数据对象分为 K_t 个簇, 定义类间距离为所有簇的聚类中心(每个簇所含数据对象的均值)到全域中心(所有数据对象的均值)的距离之和的均值.

$$\mathrm{int}erD\_1=\frac{1}{K_t}\sum_{\eta=1}^{K_t}\|m_\eta-\overline{m}\| \quad (12)$$

$\mathrm{int}erD\_1$ 为类间距离, $\overline{m}$ 为全部数据对象的均值.

定义 4. 假设数据对象分为 K_t 个簇, 定义类间距离为不同聚类簇的最近两个数据对象之间的距离.

$$\mathrm{int}erD\_2=\min_{m_{u\in C_\tau},m_{v\in C_\delta},\tau\ne\delta}\|m_u-m_v\| \quad (13)$$

式中, $\mathrm{int}erD\_2$ 为类间距离, m_u, m_v 为不同聚类簇

C_τ, C_δ 中的数据对象.

为了测试本文所提出的 ISFLA-K 算法的聚类性能, 下面将分别针对三种算法利用上文所讲的适应度值、聚类内和聚类间距离的两种计算方式进行实验, 同时为了保证实验结果精确性, 三种算法在每组数据集中将分别独立运行 100 次, 得到表 2~表 4 各种算法对不同数据集的聚类性能评价值.

表 2 三种算法对 Lenses 数据集的聚类性能对比

性能指标 / 算法	平均适应度值	平均类内距离	平均类间距离	平均类内距离	平均类间距离
K-Means	—	1.4121	0.0972	0.0644	0.0735
SFLA-K	273.3497	1.2306	0.1026	0.0607	0.0946
ISFLA-K	388.2761	1.1365	0.1874	0.0510	0.1217

表 3 三种算法对 Iris 数据集的聚类性能对比

性能指标 / 算法	平均适应度值	平均类内距离	平均类间距离	平均类内距离	平均类间距离
K-Means	—	3.7153	0.4594	0.4114	0.4833
SFLA-K	101.4301	3.4241	0.4845	0.4016	0.5108
ISFLA-K	132.2748	3.3703	0.5770	0.3609	0.5934

表 4 三种算法对 Wine 数据集的聚类性能对比

性能指标 / 算法	平均适应度值	平均类内距离	平均类间距离	平均类内距离	平均类间距离
K-Means	—	4.6332	1.6142	1.1427	1.1456
SFLA-K	59.1341	4.4573	1.8763	0.9677	1.2608
ISFLA-K	67.0172	4.2809	2.0421	0.9120	1.3382

由表 2~表 4 可以看出, 对于三种不同的数据集, ISFLA-K 算法都具有较好的聚类性能, 它的适应度平均值比 SFLA-K 大, 同时应用提出的两种聚类内距离和聚类间距离的计算公式, 得出 ISFLA-K 算法的内部距离最小, 聚类间的距离最大. 因此, ISFLA-K 算法的聚类精度高, 优于 K-Means 和 SFLA-K 聚类算法.

本文提出的 ISFLA-K 算法相对于 SFLA-K 聚类算法来说, 首先引入对立学习的策略提高初始化蛙群的质量, 并对蛙跳更新规则进行了改进, 蛙不仅包含了对过去经验的记忆, 还具备了向其它蛙学习的能力, 因此具有更强的寻优能力, 比 SFLA-K 算法聚类精度高. 针对 K-Means 初始聚类中心敏感常陷于局部极值的不足, ISFLA-K 算法把改进的混合蛙跳算法引入到 K-Means 算法中弥补其不足, 所以聚类精度优于 K-Means 算法.

6 结语

本文提出一种改进混合蛙跳和 K-Means 相结合的聚类算法 ISFLA-K, 此算法首先使用对立学习的思想产生初始种群, 保持群体多样性和较优适应度值. 然后对混合蛙跳算法的蛙跳规则进行改进, 提高算法的寻优能力. 最后利用 ISFLA 优化 K-Means 聚类算法. 实验测试结果表明, ISFLA-K 具有很好的聚类性能, 求解精度高.

参考文献

1 刘靖明,韩立川.粒子群优化 K 均值的混合聚类算法研究. 中国管理科学,2004,12(z1):96–99.

2 Cui X, Potok TE. Document clustering analysis based onhybrid PSO+K-means algorithm. Journal of Computer Sciences, 2005(Special Issue): 27–33.

3 Antariksha B. A clonal selection based shuffled frogleaping algorithm. IEEE Int Advance Computing Conf. New York, IEEE Press, 2009: 125−130.

4 郑仕链,楼才义,杨小牛:基于改进混合蛙跳算法的认知无线电协作频谱感知.物理学报,2010,59(5):3611−3617.

5 Mashhadi KA, Alinia AM. Variousstrategies for partitioning of memeplexes in shuffled frogleaping algorithm. 14th Int CSI Computer Conf. NewYork, IEEE Press. 2009. 576–581.

6 李英海,周建中,杨俊杰.一种基于阈值选择策略的改进混合蛙跳算法.计算机工程与应用,2007,43(35):19−21.

7 韩凌波,王强,蒋正锋.一种改进的 k-means 初始聚类中心选取算法.计算机工程与应用,2010,46(17):150−152.

8 http://archive.ics.uci.edu/ml/.

9 于海涛,李梓,姚念民.K-means 聚类算法优化方法的研究.小型微型计算机系统,2012,(10):2273−2277.