基于 JADE 的自动聚类算法①

唐 亚, 王振友

(广东工业大学 应用数学学院, 广州 510520)

摘 要: JADE 算法是传统差分进化算法(DE)的一种改进算法, 其收敛速度更快、优化性能更好, 拥有一套完整的自适应参数调整机制有效加强了算法的全局搜索优化能力. 本文将自适应差分进化算法(JADE)用于聚类, 提出了一个新的基于 JADE 的自动聚类算法(AC-JADE). 首先, 本文采用双交叉策略, 在传统的两点式交叉操作之后, 针对 DE 用于自动聚类时的特定的编码方式, 添加了一种基于个体间聚类中心随机交换交叉策略; 其次, 针对聚类中心选取方法的随机性导致的聚类中心有可能偏离数据集或者聚类中心过于集中的缺陷做出了相关改进, 通过先对聚类中心进行筛选在进行聚类, 有效避免了因算法本身的随机性导致的错误聚类划分. 通过对 UCI 的 4 个数据集的仿真实验比较, 该种双交叉操作的聚类算法明显好于同类算法.

关键词: 自动聚类; 差分进化; 双交叉策略

Automatic Clustering Based on JADE Algorithm

TANG Ya, WANG Zhen-You

(College of Applied Mathematics, Guangdong University of Technology, Guangzhou 510520, China)

Abstract: JADE algorithm is an improved algorithm of basic differential evolution algorithm (DE) with better convergence speed and optimization performance, whose self-adaptive parameter adjustment mechanism improves its global optimization ability. In this paper, we use self-adaptive differential evolution algorithm (JADE) for clustering and propose a new automatic clustering algorithm based on JADE，named as AC-JADE. Firstly, it takes double crossover strategy for clustering. Specifying to the encoding mode of DE used for clustering, it adds a new crossover strategy after the conventional two point crossover operation. This new crossover strategy acts directly on two clustering centers derived from parent vector and trial vector separately. Secondly, it makes improvements on the drawbacks that the selected clustering centers may deviate from the data set or they are too close results from the randomness of mode for choosing clustering center. Sifting clustering centers before choosing some of them for clustering results has a better effect. The experimental results carried on 4 UCI datasets verifies effectiveness of the proposed algorithm.

Key words: automatic clustering; differential evolution; double crossover strategy

1 引言

聚类是数据挖掘领域中重要的技术之一, 已被成功地运用到了许多领域, 例如, 模式识别、图像处理等[1-2]. J.B.MacQueen 在 1967 年提出的 K-means 算法[3], 是一种被广泛应用于科学研究和工业应用中的经典聚类算法. 该算法对大型数据集的处理效率较高, 特别是当样本分布呈现类内团聚状时, 可以达到很好的聚类结果. 欠缺的是, K-means 需要事先确定聚类个数, 而事实上这个聚类个数是很难确定的.

Swagatam Das, Ajith Abraham 和 Amit Konar 于 2008 年提出了基于差分进化的自动聚类算法(ACDE)[4], 该算法将 DE 用于聚类, 并用其特定的编码表示方法突破了传统需要人工确定聚类数的限制, 实现了自动聚类的效果. ACDE 所使用的是传统的差分进化算法(DE), 而传统的 DE 不管在收敛速度还是搜索能力上都有所欠缺, 且容易陷入局部最优. 近年

① 收稿时间:2016-01-11;收到修改稿时间:2016-03-14 [doi:10.15888/j.cnki.csa.005381]

来, 基于DE的改进已经取得了很多可观的成果, 并被成功地用于各种领域. Jingqiao Zhang 和 Arthur C. Sanderson 于 2009 年提出 JADE 算法[5], 该算法提出了一种新的变异策略以及一种自适应参数调整方法, 其有效的优化能力已经被大量的实验证明了, 特别是对于处理高维数据, JADE 的有效性都要远远高于原始的 DE. 因此, 本文将 JADE 算法用于聚类, 替代原来的 DE 算法.

Weiguo Sheng, Shengyong Chen 等人于 2014 年提出了一种新的交叉策略, 并将其用于基因文化算法进行聚类[6]. 这种策略与不同于传统的两点式交叉策略, 它随机将长短不一的两个个体中的两个完整的聚类中心进行交换. 对于聚类所需要的特定的编码方式, 这种新的交叉策略能将分类效果好的完整聚类中心保存到下一代, 且其仿真结果说明该种新的交叉策略起到了不错的效果. 但是, 文献[6]仅将个体间的聚类中心进行交换而忽略了传统两点式交叉策略能在不断迭代的过程中对聚类中心的单个基因进行优化的重要性. 基于此, 本文在保留传统的两点式交叉策略的基础上加入了一种新的交叉策略, 采用双交叉策略进行优化.

另外, 文献[4]中聚类中心是否会被选取进行聚类划分取决于聚类中心所对应的随机数是否大于其设定的阈值, 这种随机性就可能会造成选取到偏离数据集的聚类中心或者所选取得到的聚类中心过于集中[7]. 因此, 在选取聚类中心进行聚类划分前应该对聚类中心进行筛选, 以保证聚类划分的有效性和准确性.

2 JADE算法

JADE 算法由 Jingqiao Zhang 和 Arthur C. Sanderson 于 2009 年提出. 与传统 DE 相同, 在初始化之后, JADE 进入变异、交叉、选择三个步骤的不断循环, 直至满足条件才退出. 而与传统的 DE 所不同的是 JADE 提出了一种新的变异策略以及一种自适应参数调整方式.

2.1 初始化

算法首先在可执行区域的最大范围内随机生成 NP 个 D 维个体: $\{X_{i,0}=(x_{1,i,0}, x_{2,i,0},..., x_{D,i,0})|i = 1, ... , NP\}$, 例如, 第 i 个个体的第 j 维可由式(1)生成:

$$x_{j,i,0} = x_{j,\min} + rand(0,1)\cdot\left(x_{j,\max} - x_{j,\min}\right) \tag{1}$$

2.2 变异

初始化之后, NP 个个体进行变异操作, JADE 所采用的是一种新的变异策略“DE/current-to-*p*best”. 令集合A由那些被淘汰的个体组成的集合, P为当代种群组成的集合, 那么“DE/current-to-*p*best”变异策略可以用下式表达:

$$V_{i,g} = X_{i,g} + F_i\cdot\left(X^p_{best,g} - X_{i,g}\right) + F_i\cdot\left(X_{r1,g} - \tilde{X}_{r2,g}\right) \tag{2}$$

这里 $X^p_{best,g}$ 表示从当代种群中适应度值靠前的 $100p\%$的个体中随机选择的, 其中 $p\in(0,1]$, 另外$X_{i,g}$, $X_{r1,g}$是从集合 P 中随机选择的, $\tilde{X}_{r2,g}$从 P∪A中随机选择.

2.3 交叉

在变异操作之后, 紧接着的是交叉操作, 而二项式交叉操作是最常用到的交叉操作之一. 实验向量 $U_{i,g}=(u_{1,i,g}, u_{2,i,g}, ... , u_{D,i,g})$可以通过下列式子得到:

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & if \quad rand(0,1)\le CR_i \quad or \quad j = j_{rand} \\ x_{j,i,g}, & otherwise \end{cases} \tag{3}$$

其中, j_{rand}是从整数集$\{1, 2, ... , NP\}$中随机取得的整数, 交叉概率 CR 一个事先确定的 0 到 1 之间的数.

2.4 选择

选择操作是将试验向量 $U_{i,g}=(u_{1,i,g}, u_{2,i,g}, ... , u_{D,i,g})$和目标向量$\{X_{i,0} =(x_{1,i,0}, x_{2,i,0},..., x_{D,i,0})|i = 1, ... , NP\}$的适应度值大小进行比较, 两者之间适应度值较好的一个将会被成功地保留下来进入下一代, 成为下一代得父代.

$$X_{i,g+1} = \begin{cases} U_{i,g}, & if \quad f(U_{i,g}) < f(X_{i,g}) \\ X_{i,g}, & otherwise \end{cases} \tag{4}$$

2.5 自适应参数调节机制

交叉概率 CR_i 和放缩因子 F_i 由下列 两个式子产生:

$$CR_i = randn_i(\mu_{CR}, 0.1) \tag{5}$$

$$F_i = randn_i(\mu_F, 0.1) \tag{6}$$

这里 $randn(\cdot)$ 表示正态分布函数, μ_{CR} 和 μ_F 在初始化时被设置为 0.5 然后根据下列两个式子不断更新:

$$\mu_{CR} = (1-c)\cdot\mu_{CR} + c\cdot mean_A(S_{CR}) \tag{7}$$

$$\mu_F = (1-c)\cdot\mu_F + c\cdot mean_L(S_F) \tag{8}$$

这里 c 是 0 到 1 之间的一个常数, $mean_A(\cdot)$ 表示常见的平均值, 而 $mean_L(\cdot)$表示 Lehmer 平均值.

$$mean_L(S_F) = \frac{\sum_{F\in S_F} F^2}{\sum_{F\in S_F} F} \tag{9}$$

3 ACDE算法

ACDE 算法将 K-means 算法融入 DE 算法, 利用 DE 算法的随机搜索优化能力, 寻找最佳的聚类结果,

其核心问题在于个体的编码表示以及适应度函数的选取.

3.1 个体的编码表示方法

ACDE 算法将多个聚类中心表示到一个解上, 并且每个聚类中心配对一个 0 到 1 之间的随机值, 随机值的大小决定着其对应的聚类中心是否会被选取到用来聚类. 具体编码表示如下所示.

$$v=(a_1,a_2,a_3,\cdots,a_{k\max},m_1,m_2,m_3,\cdots,m_{k\max})$$

其中, a_i(i=1,2,3,…,$kmax$)表示第 i 个随机值, m_i (i=1,2,3,...,$kmax$)表示第 i 个聚类中心, m_i= ($m_{i,1}$, $m_{i,2}$, $m_{i,3}$,...,$m_{i,D}$), D 为数据维数, $kmax$ 表示作者事先设定的最大聚类中心. 若 a_i>0.5, 即第 i 个聚类中心处于活跃状态, 则聚类中心 m_i 就会被选为聚类中心.

3.2 适应度函数

ACDE 采用了两种适应度函数(*DB* 指标和 *CS* 指标)来衡量其算法的有效性, 本文选用了 *DB* 指标作为 JADE 中的适应度函数, 以下是 *DB* 的具体表示:

$$S_{i,q}=\left[\frac{1}{N_i}\sum_{X\in C_i}\|X-m_i\|_2^q\right]^{1/q} \tag{10}$$

$$d_{ij,t}=\left\{\sum_{p=1}^{d}\left|m_{i,p}-m_{j,p}\right|^t\right\}^{1/t}=\|m_i-m_j\|_t \tag{11}$$

$$R_{i,qt}=\max_{j\in K,j\neq i}\left\{\frac{S_{i,q}+S_{j,q}}{d_{ij,t}}\right\} \tag{12}$$

$$DB(K)=\frac{1}{K}\sum_{i=1}^{K}R_{i,qt} \tag{13}$$

其中, m_i 表示第 i 个聚类中心, N_i 表示划分到第 i 个类 C_i 的对象个数, t≥1, q 为正整数.

从表达意义上来说, 式(10)计算的是第 i 个聚类的散度, 衡量了第 i 个类里元素的相似性; 而式(11)计算了第 i 个类和第 j 个类之间的距离, 衡量的是两个类之间的相似性. 因此, 将 *DB* 作为适应度函数进行优化, 既能保证各个类的元素高相似性, 也能达到使任意两个类间的低相似性的目的, 从而保证了聚类划分的有效性.

4 AC-JADE算法

本文将 JADE 算法用于聚类, 提出了一种新的基于 JADE 的自动聚类算法 AC-JADE, 其完整的伪代码如算法 1 所示. 该算法在原始的两点式交叉策略之后添加了一种新的基于完整聚类中心之间的交叉策略, 并针对有编码方式所带来的随机性导致的所选取到的聚类中心可能偏移数据集或者聚类中心过于集中这一缺陷做出了相关改进, 以下是具体的改进方法.

4.1 基于完整聚类中心的交叉操作

Weiguo Sheng, Shengyong Chen 等人于 2014 年提出了一种新的交叉策略, 并将其用于基因文化算法进行聚类. 不同于传统的两点式交叉策略, 该种策略以随机的方式交换个体间的两个完整的聚类中心, 能有效地将聚类效果好的聚类中心保存到下一代, 其仿真结果证明该种新的交叉策略相对于这种特定的编码方法起到了不错的效果. 但是, 仅随机将两个聚类中心进行交换忽略了传统两点式交叉策略能在不断迭代的过程中对聚类中心的单个基因进行优化的重要性. 因此, 采用双交叉方式, 在两点式交叉策略之后添加这种基于完整聚类中心的交叉策略, 能够将两种交叉策略的优点都有效保留下来以达到最佳的聚类效果.

另外, 文献[6]将同代个体间的聚类中心进行交换, 虽能加强将这一代中好的聚类中心保存下来的可能, 但可能其父代的某个个体中有更佳的聚类中心, 因此, 本文将这种新的交叉策略作用于个体与其对应的父代个体之间, 以将父代个体中较好的个体保存下来. 具体做法如下:

随机生成两个随机整数 k_1 和 k_2, 将 u_i 的第 k_1 中心替换为 x_i 的第 k_2 类中心, 且其对应的随机数也进行变换, 而父代 x_i 不作改变. 具体做法如下:

若父代 x_i 和其传统交叉过后的解 u_i 如下表示:

x_i: <0.1 0.5 0.2 0.7 0.1|0.2 0.9 0.8 0.7 0.3|0.8 0.9 0.5 0.6 0.2|0.5 0.1 0.6 0.2 0.5|(0.21 0.49 0.87 0.92)>

u_i: <0.4 0.2 0.8 0.4 0.5|0.9 0.7 0.3 0.5 0.1|0.1 0.8 0.6 0.9 0.2|0.4 0.5 0.3 0.4 0.1|(0.32 0.11 0.98 0.56)>

若随机生成两个随机整数分别为 3 和 1, 则我们将 u_i 的第 1 聚类中心替换为 x_i 的第 3 个聚类中心, 且其对应的随机数也进行变换, 而父代 x_i 不作改变. 交叉后, u_i 的编码表示为:

u_i: <0.8 0.9 0.5 0.6 0.2|0.9 0.7 0.3 0.5 0.1|0.1 0.8 0.6 0.9 0.2|0.4 0.5 0.3 0.4 0.1|(0.87 0.11 0.98 0.56)>

4.2 聚类中心的筛选

ACDE 将每个聚类中心配对一个 0 到 1 之间的随机值, 而聚类中心是否会被选取进行聚类划分取决于聚类中心其所对应的随机数是否大于其设定的阈值, 这种随机性就可能会造成选取到偏离数据集的聚类中心或者所选取得到的聚类中心过于集中. 因此, 在选

取聚类中心进行聚类划分前应该对聚类中心进行筛选, 以保证聚类划分的有效性和准确性.

4.2.1 关于聚类中心偏离数据集的改进

根据最邻近原则统计出属于每个聚类中心的样本数 N_i, 设定阈值 θ_1,若 $\theta_1<N_i$, 则将其对应的随机数 a_i 重置为 0 到 0.5 之间的一个随机数, 并将该聚类中心重置为数据集的均值; 反之, 则将 a_i 重置为 0.5 到 1 之间的一个随机数, 使其处于活跃状态.

显然, 阈值 θ_1 的大小决定了聚类中心集中于数据集密集区域的程度, θ_1 越大, 剩下的聚类中心越少, 也越集中于数据集的密集区域, 反之, θ_1 越小, 剩下的聚类中心越多, 聚类中心集中于数据集的密集区域的效果越不明显.

4.2.2 关于聚类中心过于集中的改进

以上操作虽然在一定程度上解决了聚类中心偏离数据集的问题, 但也可能同时加重了聚类中心过于集中这一问题, 因此, 紧接着以上操作我们做出了以下改进:

首先计算出任意两个聚类中心之间的欧氏距离, 用 d_{ij} 表示第 i 个聚类中心 m_i 与第 j 个聚类中心 m_j 之间的距离, 如下图所示.

$$\begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_{k\max} \end{array}\begin{array}{c} \begin{array}{ccccc} m_1 & m_2 & m_3 & \cdots & m_{k\max} \end{array} \\ \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1,k\max} \\ d_{21} & 0 & d_{23} & \cdots & d_{2,k\max} \\ d_{31} & d_{32} & 0 & \cdots & d_{3,k\max} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{k\max,1} & d_{k\max,2} & d_{k\max,3} & \cdots & 0 \end{bmatrix} \end{array}$$

对第 i 个聚类中心 $m_i(i=1,2,3,\ldots,k_{max})$, 计算其离第 $i+1$ 到第 k_{max} 个聚类中心的距离, 设定阈值 θ_2, 找出 $d_{ij}<\theta_2(j=i+1,i+2,\ldots,k_{max})$的聚类中心, 统计其个数 n_i, 若 $n_i>0$, 则令 m_i 为满足 $d_{ij}<\theta_2$ 条件的聚类中心的均值, 且将满足 $d_{ij}<\theta_2(j=i+1,i+2,\ldots,k_{max})$条件的所有聚类中心重置为数据集的上下界的范围内的随机值.

对于阈值 θ_2 的设置做了以下处理: 对于聚类中心 $m_i=(m_{i1},m_{i2},...,m_{iD})$和聚类中心 $m_j=(m_{j1},m_{j2},...,m_{jD})$,若其每个属性之间的距离小于 1, 即若对于每个 k, 都有 $d(m_{ik},m_{jk})=|m_{ik}-m_{jk}|<1(k=1,2,...,D)$, 我们就认为两个聚类中心是集中的, 也就是说当 $d(m_i,m_j)<D$ 时, 两个聚类中心是非常接近的, 因此我们令阈值 $\theta_2=D$.

另外, 由于聚类中心的选取取决于其对应的随机值是否大于 0.5, 而随机值大于 0.5 的聚类中心个数小于 2 个的情况是可能发生的, 因此当这种情况出现时, 本文采用文献[4]的处理方式, 随机选取某个或者两个小于 0.5 随机值将其重置为 0.5 到 1 之间的随机值, 以保证能选取到的聚类中心至少有两个.

算法 1　AC-JADE 伪代码

1) 初始化: 根据(1)式生成种群, 初始化相关参数;
2) 根据(2)式进行变异操作;
3) 双交叉操作:
　3.1) 根据(3)式进行两点式交叉操作;
　3.2) 根据 4.1 节所示进行基于完整聚类中心的交叉操作;
4) 根据 4.2 节筛选聚类中心;
5) 按照 K-means 方法进行聚类, 根据 (13)式的聚类评价函数 DB 对当前的聚类结果进行评价, 然后进行根据(4)式子选择操作;
6) 根据(7)-(9)更新参数 μ_{CR} 和 μ_F
7) 如果达到最大迭代次数则停止迭代并输出聚类结果; 反之, 返回步骤 2 继续迭代.

5　仿真结果

5.1　数据来源

为了验证算法的有效性, 本文选用了 UCI 中的 4 个最为常用的数据集进行实验, 数据集的名称、属性个数以及所包含的数据对象个数如表 1 所示.

表 1　数据集

数据集名称	属性个数	数据对象数	种类
Iris	4	150	3
Wine	13	178	3
Breast Cancer	9	699	2
Glass	9	215	6

5.2　参数设置

为了保证 JADE 的最佳的优化搜索能力, 同时为了保证对比的有效性, 本文保留了 JADE 及 ACDE 原有的参数设置, 这里, 我们设置种群大小 NP=100, 最大迭代次数 I_{max}=200.

对于阈值参数 θ_1 的选取, 本文采用文献[7]的做法, 令 θ_1=3, 另外, 对于阈值参数 θ_2, 令 $\theta_2=D$, 以下是独立运行 20 次的结果.

表 2 *DB* 值及聚类数均值(*mean*)和方差(*std*)

数据集名称	ACDE				AC-JADE			
	DB		聚类数		*DB*		聚类数	
	mean	*std*	*mean*	*std*	*mean*	*std*	*mean*	*std*
Iris	0.4338	0.0068	3.05	0.0739	0.4149	0.0008	2.97	0.0533
Wine	0.4842	0.0004	3.23	0.1327	0.4834	0.0002	3.06	0.0172
Breast Cancer	1.1399	0.1388	2.31	0.0573	0.8461	0.1510	2.15	0.0489
Glass	0.6006	0.0738	6.17	0.0869	0.5911	0.0621	5.93	0.0379

表 3 误分率(*mean*)和方差(*std*)

数据集名称	ACDE		AC-JADE	
	mean	*std*	*mean*	*std*
Iris	5.72	0.27	5.23	0.19
Wine	41.15	0.02	40.99	0.01
BreastCancer	26.75	0.25	25.15	0.22
Glass	8.86	0.43	8.67	0.12

5.3 结果评价

本文采用 *DB* 指标作为适应度函数进行优化选择, 一方面取 20 次最终的 *DB* 值的均值和方差作为评价标准, 另一方面取 20 次实验的聚类数的均值和方差来评价算法的聚类性能. 从表 2 可以看出, 在 4 种数据集上, AC-JADE 的聚类结果的 *DB* 值的均值和方差都比 ACDE 要小, 而最终的聚类数 AC-JADE 都比 ACDE 要接近数据集的实际种类. 因此, 不论是算法的聚类效果还是其稳定性, AC-JADE 都比 ACDE 要好. 同时, 表 3 中 AC-JADE 的误分率比 ACDE 的误分率都要小, 这也说明了 AC-JADE 算法比 ACDE 算法有更高的聚类划分精度.

6 结论

本文将较传统 DE 优化性能更为优越的 JADE 算法用于聚类, 采用双交叉策略, 在传统的两点式交叉操作之后, 针对 DE 用于自动聚类时的特定的编码方式, 添加了一种基于个体间聚类中心随机交换交叉策略. 另外, 针对由聚类中心选取方法的随机性导致的聚类中心有可能偏离数据集或者聚类中心过于集中的缺陷做出了相关改进, 通过先对聚类中心进行筛选在进行聚类, 有效避免了因算法本身的随机性导致的错误聚类划分. 通过 20 次独立运行实验结果对比发现, 不论是算法的聚类效果还是其稳定性都得到了提高.

参考文献

1 Jain-Dubes. Algorithms for clustering data. Prentice Hall, 1988.

2 Celenk M. A color clustering technique for image segmentation. Computer Vision, Graphics, and Image Processing, 1990, 52(2): 145–170.

3 MacQueen J. Some methods for classification and analysis of multivariate observations. Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. 1967, 1(14): 281–297.

4 Das S, Abraham A, Konar A. Automatic clustering using an improved differential evolution algorithm. IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans, 2008, 38(1): 218–237.

5 Zhang J, Sanderson AC. JADE: Adaptive differential evolution with optional external archive. IEEE Trans. on Evolutionary Computation, 2009, 13(5): 945–958.

6 Sheng W, Chen S, Fairhurst M, et al. Multilocal search and adaptive niching based memetic algorithm with a consensus criterion for data clustering. IEEE Trans. on Evolutionary Computation, 2014, 18(5): 721–741.

7 潘章明.一种改进的差分进化自动聚类算法.计算机仿真, 2010,(11):69–72.