一种基于差分演化的 K-medoids 聚类算法*

孟 颖[a], 罗 可[a], 刘建华[b], 石 爽[c]
(长沙理工大学 a. 计算机与通信工程学院; b. 电气与信息工程学院; c. 交通运输工程学院, 长沙 410114)

摘 要: 针对传统的 K-medoids 聚类算法具有对初始聚类中心敏感、全局搜索能力差、易陷入局部最优、收敛速度缓慢等缺点,提出一种基于差分演化的 K-medoids 聚类算法。差分演化是一类基于种群的启发式全局搜索技术,有很强的鲁棒性。将差分演化的全局优化能力用于 K-medoids 聚类算法,有效地克服了 K-medoids 聚类算法的缺点,缩短了收敛时间,改善了聚类质量。通过仿真验证了此算法的稳定性和鲁棒性。

关键词: 差分演化; 聚类质量; K-medoids 算法; 全局优化
中图分类号: TP331 **文献标志码**: A **文章编号**: 1001-3695(2012)05-1651-03
doi:10.3969/j.issn.1001-3695.2012.05.013

K-medoids clustering algorithm method based on differential evolution

MENG Ying[a], LUO Ke[a], LIU Jian-hua[b], SHI Shuang[c]
(*a. Institute of Computer & Communication Engineering, b. Institute of Electrical & Information Engineering, c. Institute of Traffic & Transportation Engineering, Changsha University of Science & Technology, Changsha* 410114, *China*)

Abstract: The traditional K-medoids clustering algorithm, because on the initial clustering center sensitive, the global search ability is poor, easily trapped into local optimal, slow convergent speed, and so on. Therefore, this paper proposed a kind of K-medoids clustering algorithm based on differential evolution. Differential evolution was a kind of heuristic global search technology population, had strong robustness. It combined with the global optimization ability of differential evolution using K-medoids clustering algorithm, effectively overcame K-medoids clustering algorithm, shortend convergence time, improved clustering quality. Finally, the simulation result shows that the algorithm is verified stability and robustness.
Key words: differential evolution(DE); cluster quality; K-medoids algorithm; global optimization

在现有的聚类算法中,K-medoids 算法是解决聚类分析问题的一种经典算法[1],广泛应用于数据挖掘和知识发现。但因 K-medoids 算法[2]在寻找聚类中心的过程中,对初始聚类中心点选择敏感,全局搜索能力差,易陷入局部最优,尤其用在海量数据集和大矢量空间中,这种算法的性能更差。这些缺陷极大地限制了其应用范围。

差分演化(DE)是一种快速而有效的演化算法,由 Storn 等人[3]提出,是基于群体智能理论的优化算法,它具有较强的全局搜索能力和鲁棒性、结果简单、容易操作、实用性强。因此,DE 作为一种高效的并行搜索算法,被广泛用于神经元网络、机械设计、机器人、信号处理、生物信息学等领域[4]。该算法在应用方面取得了很多突破性研究进展。文献[5]中将差分演化用于粗糙集中,得到了很好的效果。但是如何用 DE 算法来优化 K-mediods 聚类质量,值得进一步研究。

鉴于此,本文提出一种基于差分演化的 K-mediods 聚类算法,利用 DE 有较强的全局搜索能力和鲁棒性、求解效率高等特点,不仅能有效地克服 K-medoids 聚类算法的缺点,而且能提高算法全局搜索能力,缩短收敛时间,改善聚类质量。

1 预备知识

1.1 DE 算法

DE 是一种基于群体进化的算法,具有记忆个体最优解和种群内信息共享的特点,即通过种群内个体间的合作与竞争来实现对优化问题的求解,其本质是一种基于实数编码的具有保优思想的贪婪遗传算法。

1.1.1 DE 算法思想

为保证较大的搜索空间,DE 算法使用随机函数生成初始种群 $X^0=[x_1^0,x_2^0,\cdots,x_{N_p}^0]$。其中:$N_p$ 为种群规模,用于表示特征优化的个体 $x_i^0=[x_{i,1}^0,x_{i,2}^0,\cdots,x_{i,D}^0]$,$D$ 为优化可行解的维数。

DE 算法中变异和交叉操作的处理[6]如下:

a)变异操作。对每一个在 t 时刻的个体 x_i^t 进行变异操作,得到与其相对应的变异个体 V_i^{t+1},即

$$V_i^{t+1}=\boldsymbol{x}_{r_1}^t+K(\boldsymbol{x}_{r_2}^t-\boldsymbol{x}_{r_3}^t) \tag{1}$$

其中:$r_i,r_2,r_3\in\{1,2,\cdots,N_p\}$,$r_1\neq r_2\neq r_3$;$\boldsymbol{x}_{r_1}^t$ 为父代基向量;$\boldsymbol{x}_{r_2}^t-\boldsymbol{x}_{r_3}^t$ 为父代差分向量;K 为缩放比例因子,其取值一般在 0.4 ~ 0.6 之间,当 K 接近于 0.5 时,变异个体接近于父代向量。

b)交叉操作。对个体 x_i^t 和变异个体 V_i^{t+1} 进行交叉操作,得到优化实验个体 U_i^{t+1},即

收稿日期: 2011-10-13; **修回日期**: 2011-11-21 **基金项目**: 国家自然科学基金资助项目(11171095,10871031);湖南省自然科学衡阳联合基金资助项目(10JJ8008);湖南省科技计划项目(2011FJ3051);湖南省教育厅重点项目(10A015)

作者简介: 孟颖(1984-),女,河南开封人,硕士研究生,主要研究方向为数据挖掘、计算机网络等(kfmengying@126.com);罗可(1961-),男,湖南长沙人,教授,博士,主要研究方向为数据挖掘、计算机应用等;刘建华(1985-),男,湖南永州人,硕士研究生,主要研究方向为电力系统运行与控制;石爽(1983-),男,山东定陶人,硕士研究生,主要研究方向为交通运输规划与管理.

$$U_{i,j}^{t+1}=\begin{cases}V_{i,j}^{t+1} & \text{if}(\text{rand}(j)\leqslant P_{CR})\ \text{or}\ (j=\text{rnbr}(i))\\ x_{i,j}^{t} & \text{otherwise}\end{cases}\tag{2}$$

其中：rand(j) ~ U[0,1]之间的均匀分布随机数；P_{CR}为[0,1]之间的交叉概率；rnbr(i)为{1,2,…,D}之间的随机量。

经过变异交叉得到新种群的个体 x_i^{t+1}，即

$$x_i^{t+1}=\begin{cases}u_i^{t+1} & \text{if}(f(u_i^{t+1})<f(x_i^t))\\ x_i^t & \text{otherwise}\end{cases}\tag{3}$$

其中：f为目标函数。

DE 算法基本思想是：

a)初始化种群 X^0；

b)对当前种群进行变异，利用式(1)产生新的变异个体；

c)对新的变异个体进行交叉，利用式(2)实施交叉，产生新的实验个体；

d)利用式(3)对实验个体和目标函数进行比较，选择目标函数最低值的个体作为新种群的个体。

1.1.2 DE 算法步骤

DE 算法的搜索性能取决于算法全局探索和局部开发能力的平衡[7]，而这在很大程度上依赖于算法控制参数的选取，包括种群规模、缩放比例因子和交叉概率等。相对其他进化算法而言，DE 所需调节的参数较少。

a)确定 DE 算法控制参数和所采用的差分策略，其中，控制参数包括种群、变异、交叉、最大进化数、终止条件等。

b)随机产生初始种群 X^0，进化代数 n。

c)对 X^0进行评价，即计算 X^0中每个个体的目标函数值。

d)判断是否达到终止条件或 n 达到最大。若是，则进化终止，将此时的最优个体作为最优解输出；否则，继续。

e)进行变异和交叉操作，对边界条件进行处理，得到中间种群。

f)对临时种群进行评价，计算中间种群中每个个体的目标函数值。

g)进行选择操作，得到新种群。

h)进化代数 $n=n+1$，转步骤 d)。

DE 算法优点归纳如下：算法通用，不依赖于问题信息；算法原理简单，容易实现；群体搜索，具有记忆个体最优解的能力；协同搜索，具有利用个体局部信息和群体全局信息指导算法进一步搜索的能力；易与其他算法结合，构造出具有更优胜能力的算法。

1.2 K-medoids 算法简介

1.2.1 K-medoids 聚类算法思想

K-medoids 聚类算法的核心是中心点的选择[8]。假设聚类 C 原先的中心点是 O_{c_old}，现拟改为 O_{c_new}，根据数据对象属于和其距离最近的聚类原则，可能引起各数据对象所属聚类的情况发生调整。对于原先聚类 C 中的数据对象 p 可能有如下情况：

a)p 与 O_{c_new}的距离仍然小于其他聚类中心点的距离，因此 p 仍属于聚类 C，如图 1(a)所示，如用 O_{c_new}代替 O_{c_old}，数据对象 p 的代价为 $d(p,O_{c\_new})-d(p,O_{c\_old})$，$d$ 表示两点之间的距离。

b)p 与其他某一聚类 r 的中心点的距离最短，则 p 将改属于聚类 r，如图 1(b)所示，如用 O_{c_new}代替 O_{c_old}，数据对象 p 的代价为 $d(p,O_r)-d(p,O_{c\_old})$，其中 O_r 为聚类 r 的中心点，此时代价为正值。

类似地，原先聚类 C 外的任意数据对象 p 也可能有两种情况：

c)p 仍然与它原先所属的聚类的中心点距离最短，则 p 仍将属于原先的聚类，如图 1(c)所示，如用 O_{c_new}代替 O_{c_old}，数据对象 p 的代价不变。

d)在所有聚类的中心点中，p 与 O_{c_new}的距离最短，则 p 将改属于聚类 O_{c_new}，如图 1(d)所示，如用 O_{c_new}代替 O_{c_old}，数据对象 p 的代价为 $d(p,O_{c\_new})-d(p,O_r)$，其中 O_r 为聚类 r 的中心点，此时代价为负值。

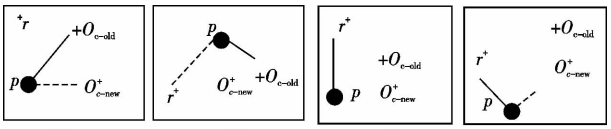

图1 聚类中心优化示意

假如 O_{c_new}使得总代价(所有数据对象代价之和)小于 0，则 O_{c_new}代替 O_{c_old}成为聚类的新的聚类中心；反之，则说明 O_{c_new}目前不适合作为聚类 C 的新聚类中心，重新试探其他点。

1.2.2 K-medoids 聚类算法步骤

输入：聚类个数 k，包含 n 个数据对象的数据库。

输出：k 个聚类使得所有对象与其最近中心点的相异度总和最小。

1)从 n 个数据对象随机选择 k 个对象作为初始聚类的中心点；

2)Repeat

3)指派每个剩余的对象给离它最近的中心点所代表的聚类；

4)随机选择一个非中心对象 O_{rundom}；

5)计算用 O_{rundom}代替 O_j 的总代价 S；

6)if $S<0$, then O_{random}代替 O_j，形成新的 k 个中心点的集合；

7)Until 不发生变化。

2 算法设计

2.1 算法基本思想

基于 DE 的 K-medoids 聚类算法思想是：从种群中随机选取可行解规模为 N_1 的群体，群体的最大规模为 N_2；然后，利用 DE 算法对种群进行变异交叉，确定种群间的最优变异个体 x_i^{t+1} 和最优实验个体 U_i^{t+1}，避免种群陷入局部最优，加强算法对聚类所在空间区域的局部搜索能力；再根据 K-medoids 聚类算法对这些最优实验个体 U_i^{t+1}进行新的聚类分析，确定每个个体所在的聚类以及它们之间的最优距离；从而得到目标函数最低值的个体作为新种群的个体，输出最优解。

2.2 算法步骤与流程

a)对种群进行初始化操作，确定种群个数、变异交叉次数、最大进化次数。

b)根据式(1)对种群进行变异，计算每个个体之间的差分向量，确定种群间的最优变异个体，并将此作为种群的新变异个体。

c)根据式(2)对新的变异个体进行交叉，确定种群间的最优实验个体。

d)根据 K-medoids 聚类算法对个体进行新的聚类分析，确定每个个体所在的聚类以及它们之间的最优距离。

e)对形成的新种群根据式(3)对实验个体和目标函数进行比较，选择目标函数最低值的个体作为新种群的个体。

f)判断是否达到终止条件(到达最优进化次数或者最优解)。若是，则进化终止，将此时的最优个体作为最优解输出；否则，转向 b)。

其流程如图 2 所示。

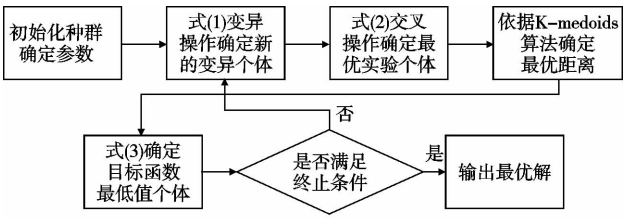

图2　算法流程

2.3　算法复杂度分析

2.3.1　空间复杂度

算法的空间复杂度:种群规模 N,变异交叉次数 M,最大进化数 max_N,优化可行解的维数 D。

UCI 中的数据集:聚类中心个数 k,样本的属性个数 p。由于规模上 N 远远比样本的属性个数 p 要大,max_N 与 D 均为常数,所以,算法总体的空间复杂度为 $T = O(N \times M \times \max\_N + D \times k \times N)$。

2.3.2　时间复杂度

本文仅考虑种群规模对时间复杂度的影响。

变异操作的时间复杂度为 $T_1 = O(N_1)$;最大规模的时间复杂度为 $T_2 = O(N_2)$;群体所需的时间复杂度为 $T_3 = O(N_1 + N_2)$;交叉更新群体的最坏时间复杂度为 $T_4 = O((N_1 + N_2)^2)$;保持最优群体的最坏时间复杂度为 $T_5 = O((N_1 + N_2)^3 + (N_1 + N_2)\lg(N_1 + N_2))$;算法迭代一次的最坏时间复杂度为 $T = \sum_{i=1}^{5} T_i$。

由于 T_5 与 T_4 要远远比 T_3、T_2、T_1 大得多,所以本文算法的总体时间复杂度可写为 $T_o = O((N_1 + N_2)^3 + (N_1 + N_2)^2 + (N_1 + N_2)\lg(N_1 + N_2))$。

3　仿真实验

1)仿真环境　软件为操作系统 Windows XP,编译软件 MATLAB 7.0.1;硬件为 Pentium® Dual-Core CPU T4200@2.00 GHz,内存 2 GB。

2)采用的数据集　本文分别对 UCI 中 4 维 Iris 数据集、高维的数据集和多样本高维的数据集进行性能测试。

3)参数设置　可行解集合的规模 $N_1 = 100$,N_1 指代种群大小,随机生成个体的数目 $N = 200$,参数 $P_{CR} = 0.5$。为了排除随机影响,采用迭代次数 1 000 作为算法终止条件,每个算法重复运行 10 次,对结果进行统计分析。

下面分别用 K-medoids 算法、PSO 聚类[9]、PSO-Kmeans 算法[10]、K-means 算法[11]和本文 DE-Kmedoids 算法对上述数据集进行聚类分析,比较各自的稳定性与收敛时间,如表 1 ~ 3 所示。

表 1　Iris 数据聚类结果比较

算法	最大值	最小值	均值	中间值	时间/s
K-means	79.331	8.070	33.102	28.180	1.012
PSO	61.097	22.479	39.852	40.422	5.847
PSO-Kmeans	24.437	7.535	12.967	12.952	51.831
K-medoids	21.335	6.984	11.697	11.594	6.457
DE-Kmedoids	18.765	5.742	10.258	10.150	4.326

从实验结果可以看出,本文改进的算法与其他算法相比,能够有效地逃出局部最优而找到全局最优解,增强了全局搜索能力。本文算法的均值、中间值及收敛时间均优于其他算法,显示出 DE-Kmedoids 聚类算法的稳定性和鲁棒性。针对收敛时间而言,由于算法本身的精度所在,K-means 算法的收敛时间小于 DE-Kmedoids 聚类算法,但 DE-Kmedoids 聚类算法的稳定性要比 K-means 算法优越很多。

表 2　高维数据聚类结果比较

算法	最大值	最小值	均值	中间值	时间/s
K-means	279.315	61.070	140.102	146.078	13.313
PSO	188.263	98.389	132.561	132.025	135.237
PSO-Kmeans	80.869	59.705	71.139	71.618	1 037.400
K-medoids	69.460	45.180	62.452	62.313	179.235
DE-Kmedoids	60.480	39.761	49.491	49.362	140.357

表 3　多样本高维数据聚类结果比较

算法	最大值	最小值	均值	中间值	时间/s
K-means	1291.471	216.780	701.124	638.218	20.137
PSO	991.471	434.554	664.134	674.449	213.893
PSO-Kmeans	377.833	234.656	271.899	266.039	3 076.100
K-medoids	289.367	196.341	243.647	233.679	421.106
DE-Kmedoids	260.457	164.753	213.489	210.608	369.543

4　结束语

本文借助 DE 算法的基本思想,提出了一种基于 DE 的 K-medoids 聚类算法。该方法在对 K-medoids 聚类算法研究的基础上,克服了 K-medoids 聚类算法对初始聚类中心敏感、全局搜索能力差、易陷入局部最优和收敛时间缓慢的缺点。通过对 Iris数据集和高维数据集的仿真实验,将算法与 K-means、PSO、PSO-Kmeans 和 K-medoids 算法进行比较,验证了本文 DE-Kmedoids 算法的稳定性和鲁棒性,并改善了聚类质量。但是,如何降低该算法的时间复杂度,将是下一步研究的重点。

参考文献:

[1] SCHOLKOPF B, MIKA S, BURGESC J C, *et al*. Input space versus feature space in kernel-based methods[J]. IEEE Tran on Neural Networks, 1999, 10(5): 1000-1017.

[2] GUO Hai-xiang, ZHU Ke-jun, GAO Si-wei, *et al*. An improved genetic K-means algorithm for optimal clustering[C]//Proc of the 6th IEEE International Conference on Data Mining Workshops. Washington DC: IEEE Computer Society, 2006: 793-797.

[3] STORN R, PRICE K. Minimizng the real functions of the ICEC'96 contest by differential evolution[C]//Proc of IEEE International Conference on Evolutionary Computation. Nagoya: IEEE, 1996: 842-844.

[4] STORN R, PRICE K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces[R]. Berkeley: University of California, 2006: 643-689.

[5] 高意,颜宏文. 基于差分演化算法的粗糙集属性约简[J]. 计算机应用, 2010, 30(9): 2329-2331.

[6] PEI Zhen-kui, YU Hui, ZHAO Yan-Li. Image restoration based on different evolution algorithm[J]. Journal of PLA University of Science and Technology: Natural Science Edition, 2010, 11(5): 489-492.

[7] LIU jun-hong, LAMPINEN J. A fuzzy adaptive differential evolution algorithm[J]. Soft Computing, 2005, 9(6): 448-462.

[8] 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): 674-688.

[9] 陶新民,徐晶,杨立标,等. 一种改进的粒子群和 K-均值混合聚类算法[J]. 电子与信息学报, 2010, 32(1): 92-97.

[10] 苏锦旗,薛惠锋,詹海亮. 基于划分的 K-均值初始聚类中心优化算法[J]. 微电子学与计算机, 2009, 26(1): 8-11.

[11] 任景彪,尹绍宏. 一种有效的 K-means 聚类初始中心选取方法[J]. 计算机与现代化, 2010(7): 84-86, 92.