基于一种改进的遗传算法的空间聚类分析

钱光超,贾瑞玉,张 然,李龙澍
(安徽大学 计算机科学与技术学院,安徽 合肥 230039)

摘 要:空间数据挖掘是数据挖掘的一个研究分支。空间聚类分析是空间数据挖掘的一个重要的研究领域。传统的K-均值方法用于聚类具有收敛速度快、算法实现简单等特点,但容易陷入局部最优,并对初始解敏感。遗传算法是一种全局搜索算法,但是收敛速度较慢。提出一种改进的遗传算法进行聚类,该算法通过全局搜索与局部搜索相结合,取得较好效果。实验表明:文中提出的算法在聚类分析中搜索到全局最优解(或近似全局最优解)的能力要优于经典的K-均值聚类算法,且局部收敛速度和全局收敛性能较好。

关键词:空间数据挖掘;空间聚类;K-均值算法;遗传算法
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2007)12-0071-05

Spatial Clustering Analysis Based on an Improved Genetic Algorithm

QIAN Guang-chao, JIA Rui-yu, ZHANG Ran, LI Long-shu
(School of Computer Science and Technology, Anhui University, Hefei 230039, China)

Abstract: Spatial data mining (SDM) is a branch of data mining (DM). Spatial clustering is an important field in SDM. Although the traditional K-means algorithm has good convergence rate and can be realized easily, it can easily be trapped in a local optimum, and it is sensitive in initial setting. Theoretically, genetic algorithm (GA) is a global optimization method, but with low convergence rate. Proposes an improved GA for spatial clustering. By integrating with global searching and local searching, it gets a better result . Experiments show that the ability of catch the global best solution (or approximate global best solution) of this algorithm in clustering analysis is better than classical K-means algorithm. And this new algorithm has better local convergence rate and global convergence performance.

Key words: spatial data mining; spatial clustering; K-means algorithm; genetic algorithm

0 引 言

空间数据挖掘是从空间数据库中识别或提取隐含特征和未知关系,是数据挖掘研究的子领域。其目的是:

1)抽取有意义的空间模式和特征;

2)获取空间数据和非空间数据之间的内在联系;

3)在更高概念层次上以简洁方式展示数据规律性;

4)利用所获"知识"调整空间数据库结构,使之更适合表达数据语义,提高数据库整体性能。

空间聚类是空间数据挖掘的重要组成部分,是按照某种相似性度量准则,将空间数据集分组成为由类似对象组成的多个类或簇的过程。类中对象彼此间具有较高相似性,类间对象具有较大差异性。

主要的聚类算法包括:层次方法和划分方法[1]。其中基于划分的聚类方法具有算法简单、收敛速度快的特点,但是该方法经常以局部最优结束聚类过程,而没有考虑数据分布的全局特性。遗传算法[2]是由美国密执安大学的Holland教授等人创立的,它仿效了遗传学中生物从低级到高级的进化过程,已被成功地应用于各种优化问题,而且遗传算法是以概率1收敛到全局最优点也已得到证明[3]。空间数据的聚类问题在一定条件下可以归结为一个带约束的最优化问题,因此遗传算法作为一种鲁棒性很强的全局优化算法,可以用于解决聚类问题[4]。由于空间聚类问题的复杂性以及聚类规模大的特点,如直接运用遗传算法去解决空间聚类问题,效率很低,收敛速度慢。K-均值算法[1]是一种基于划分的聚类算法,该算法采用了启发式方法,用每类的平均值来表示该类,降低了计算的复杂性,实现简单,收敛速度快,使得处理大规模数据集成为可能。但是该算法存在以下两个缺陷:

(1)对初值敏感,即随机产生的初始值不同聚类的

收稿日期:2007-03-11
基金项目:安徽省教育科研项目(2005kj056)
作者简介:钱光超(1982-),男,安徽无为人,硕士研究生,研究方向为智能软件;贾瑞玉,副教授,研究方向为计算机图形学、数据挖掘、人工智能。

效果差别很大;

(2)该算法是基于目标函数的方法,通常采用梯度法求极值,由于梯度法的搜索方向是沿着能量下降的方向,不可避免会常常陷入到局部最优。

文中综合遗传算法的全局收敛性和 k-均值方法的收敛速度快的优点,并针对聚类问题的具体特点,采用了一种改进的遗传算法,提高了遗传算法的搜索效率,同时也改善了聚类结果。

1 聚类问题描述

在 N 维的欧氏空间 R^n 中的聚类过程就是把一个给定的拥有 n 个点的集合基于某种相似度或者相异度标准划分成 K 个簇[5]。

设模式样本集为 $\boldsymbol{X} = \{\boldsymbol{X}_i, i = 1,2,\cdots,n\}$,其中 $\boldsymbol{X}_i$ 为 d 维模式向量,聚类问题就是要找到一个划分 $C = \{c_1, c_2, \cdots, c_k\}$,满足:

$$X = \bigcup_{i=1}^{k} c_i$$

$$c_i \neq \emptyset \quad (i = 1,2,\cdots,k)$$

$$c_i \cap c_j = \emptyset, (i,j = 1,2,\cdots,k; i \neq j)$$

并使得总的类间离散度之和:

$$J = \sum_{i=1}^{k} \sum_{X_l \in c_i} d(X_l, m_i) \tag{1}$$

达到最小,其中 m_i 是第 i 个聚类的中心。$d(X_l, m_i)$ 为 X_l 到对应聚类中心 m_i 的距离。聚类准则函数 J 即为各类样本到其对应聚类中心的距离之和。在空间聚类分析中 $d(X_l, m_i)$ 为欧氏空间距离,即:

$$d(X_l, m_i) = \| X_l - m_i \|$$

2 遗传聚类算法

综合遗传算法高效全局优化搜索与 K-均值方法局部搜索快速的优点,提出了一种改进的遗传空间聚类算法。并结合空间数据所特有的特点采用了一种基于最近邻基因匹配的交叉算子,使得交叉过程能不断产生有意义的新个体,保证种群的多样性。对进化产生的新个体,即对选择交叉变异之后的个体插入了一次 K-均值聚类计算,这样可以加快收敛速度。同时文中还在选择以及变异的操作中作了一些灵活变动,实验证明效果较好。算法大致步骤描述如下:

输入:

变异概率 P_m

交叉概率 P_c

种群规模 populationsize

最大进化代数 Max_ Gen

输出:最后的聚类结果的编码 s^*

{初始化种群 P;

for(int $i = 1; i <= $ Max_ Gen; i ++)

{

计算种群 P 中每个染色体的适应度;

P' = 选择操作后得到的种群;

交叉操作;

For(int $j = 1; j <= $ populationsize; j ++)

{对种群中每个染色体根据变异概率进行变异

For(int $j = 1; j <= $ populationsize; j ++)

{对经过交叉、变异后的种群进行一次 K-均值聚类计算}

}

找到种群 P 中最优的染色体,将其编码字符串赋值给 s^*

}

算法的详细实现介绍如下:

(1) 编码。

针对聚类问题的具体特点,遗传算法通常有以下两种编码方式:

① 将可行解空间直观地编成这样的一个染色体 $sw, sw = s_1 s_2 \cdots s_n$($n$ 是模式数目),染色体长度是 n,每个基因 s_i 取值{1,…,k}(k 是空间聚类最后结果簇的数目),每个基因代表一个模式,其值代表了其所对应的聚类的簇的标号,这是一种基于划分的整数编码。

② 采用一种基于聚类中心的实数编码方式,每个染色体是由一系列实数组成,这些实数代表了相应的聚类中心。如对于一个 N 维的空间,染色体的长度就是 $N * K$(K 是结果簇的数目),第一个前 N 个位置的实数代表了这 N 维空间的第一个聚类中心,后面的 N 个位置的实数代表了第二个聚类中心,依次下去。染色体中的实数对都是一个独立的基因。

由于空间聚类对象数目较大,如采用第一种编码方式则染色体长度较长,而聚类的最终结果簇的数目 k 往往远远小于聚类问题的样本数目 n,因此采用第二种基于聚类中心的实数编码方式更为有效。一个染色体由 k 个聚类中心组成:$S = s_1 s_2 \cdots s_k$,s_i 就是第 i 个聚类中心,在空间聚类中 s_i 就是第 i 个聚类的中心的坐标,是一个二维向量。

(2) 种群初始化。

初始种群随机产生。即随机选择 k 个对象作为初始的聚类中心,作为初始个体的染色体编码串,共生成 populationsize 个初始个体,作为第一代种群。

(3) 适应度的计算。

适应度的计算包含两个步骤:

① 根据种群中染色体所表示的聚类中心将空间对象分配到相应的簇中,形成聚类划分。这步根据下面的最近邻法则来完成,即将每个点 $X_i, i = 1,2,\cdots,n$ 分配给簇 c_j(c_j 的中心是 m_j),满足:

$$\| X_i - m_j \| < \| X_i - m_p \|, p = 1,2,\cdots,k \text{ 且 } p \neq j \quad (2)$$

该划分工作完成后就形成了新的聚类中心,新的聚类中心是根据对应的聚类中的点计算这些点的重心得到的,以此来替换原先的聚类中心。

② 根据形成的聚类划分和聚类中心以及公式(1)计算类间离散度之和 J。适应度函数定义为 $f = 1/(1 + J)$,这样较优的个体,其类间离散度较小,因此适应度也较高。

(4) 选择算子。

群体中 10% 的优秀个体直接进入下一代,剩下的部分采用锦标赛选择方式,将淘汰的个体用随机产生的个体代替。这样既能保证下一代的最佳个体绝对不比上一代的最佳个体差,同时由于锦标赛选择方式只适应适应度相对值作为选择标准避免了超级个体的影响,而随机产生的个体在一定程度上也避免了陷入局部最优的陷阱。

(5) 交叉算子。

选择出的个体随机地两两配对后 对于每一对个体以交叉概率进行交叉运算交叉操作是遗传算法的关键部分,而交叉算子的优劣,在很大程度上决定了算法的性能好坏。由于聚类中心编码串的无序性,两条染色体的等位基因之间,在空间位置上并不互相对应,因此如果直接进行单点或多点交叉,父代个体的优良性状不能有效地被传到下一代,而且还有可能产生非法染色体,为了更有效地产生新的有意义的个体,保持群体的多样性,定义以下的基于最近邻基因匹配的交叉运算:

设 $S1 = s_1^1 s_2^1 \cdots s_k^1$ 和 $S2 = s_1^2 s_2^2 \cdots s_k^2$,$s$ 为待交叉的两条染色体,对 $S1$ 每个 s_i^1,选择 $S2$ 中与 s_i^1 距离最近的 s_j^2,将 s_j^2 与 s_i^1 配对,并且在基因配对过程中,已经配对的基因,不再参加后续的配对。这样,$S1$ 与 $S2$ 的基因最终可以两两配对。再将 $S2$ 按照基因配对的顺序重新排列,得到 $S2^*$。对 $S1$ 和 $S2^*$,随机选择交叉点进行单点交叉,得到新个体 $S1'$ 和 $S2'$。

交叉操作的结果与双亲一起保留。至于它们是否被最终保留则由适应度函数决定。这样,种群就更加丰富。

(6) 变异算子。

变异是能够跳出局部最优的关键,变异算子对最终能否获得全局最优解影响重大。这里,变异运算按照基因位进行,每一个基因位上的聚类中心,以变异概率发生随机变异,发生变异的基因位被随机选择的聚类中心所代替。

(7)K – 均值优化。

对于新一代个体,按照以下的 K – 均值算法进行优化:

① 根据个体的聚类中心编码,按照最近邻法则(式(2)),确定对应该染色体的聚类划分;

② 按照聚类划分,计算新的聚类中心,取代原来的编码值。

由于 K – 均值算法具有较强的局部搜索能力,因此引入 K – 均值优化后的遗传算法的收敛速度可以大大提高。

(8) 非正常染色体的调整。

第(7) 步的操作在可以提高遗传算法的收敛速度的同时也可能会产生异常的染色体,即在对应该染色体的聚类划分中,可能会出现空的聚类。通过以下方法对异常的染色体进行调整:从空的聚类中心 c_i 最临近的一个非空聚类中心 c_j 中取出一个对象 X_l,X_l 是聚类 c_j 中离 c_j 中心最远的一个对象,将这个 X_l 放入 c_i 中,如此反复,直到没有空的聚类。

3　实验结果演示

实验针对两个不同的仿真数据集分别采用 K – 均值算法和文中提出的算法进行聚类分析,并对其结果进行比较。

两个仿真数据集如下图所示:图 1 是没有相互覆盖的空间数据点(1000 个点);图 2 是有相互覆盖的空间数据点(1000 个点)。

算法参数设置如下:

K – 均值算法中:$k = 6$。

文中的遗传算法:聚类数目为 6,种群规模 popula-

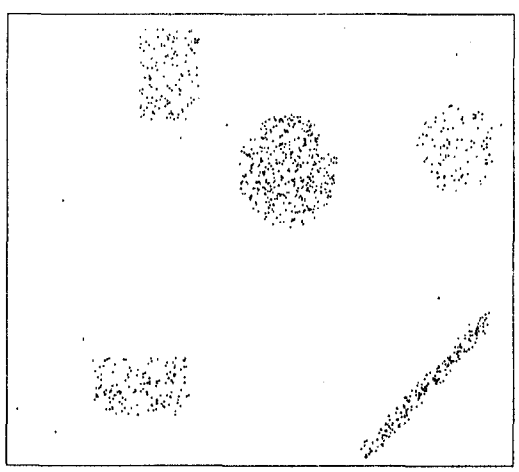

图 1　无相互覆盖的样本数据集

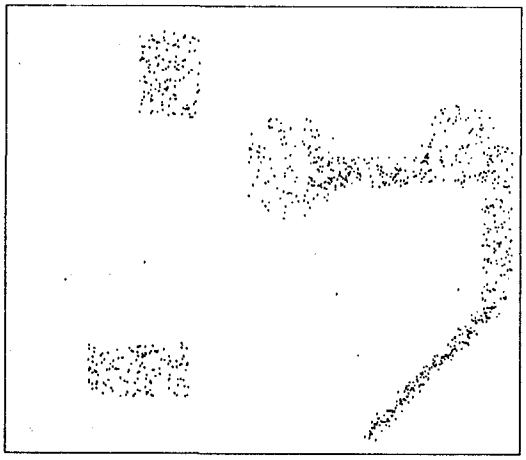

图 2 有相互覆盖的样本数据集

tionsize=60,交叉概率 $Pc=0.9$,变异概率 $Pm=0.01$(变异概率不宜过大,否则易引起震荡),最大代数为 50。每个实验分别进行 4 次。

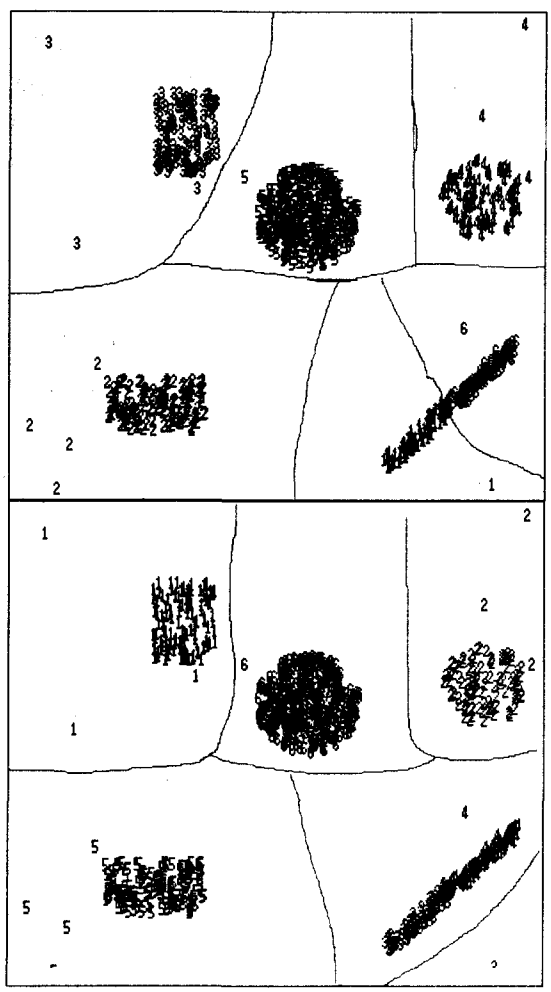

图 3 K-均值算法和改进的遗传算法针对图 1 的样本集的聚类结果

图 3,图 4 分别是运用 K-均值算法和改进的遗传算法对两个模拟数据进行聚类的实验比较示意图(其中 K-均值算法实验取的是 4 次实验中最好的结果,改进的遗传算法的几次实验效果差别很小)。很明显,文中提出的算法的效果要比单纯的 K-均值算法的效果好得多,尤其是在那种聚类形状较为复杂的时候。

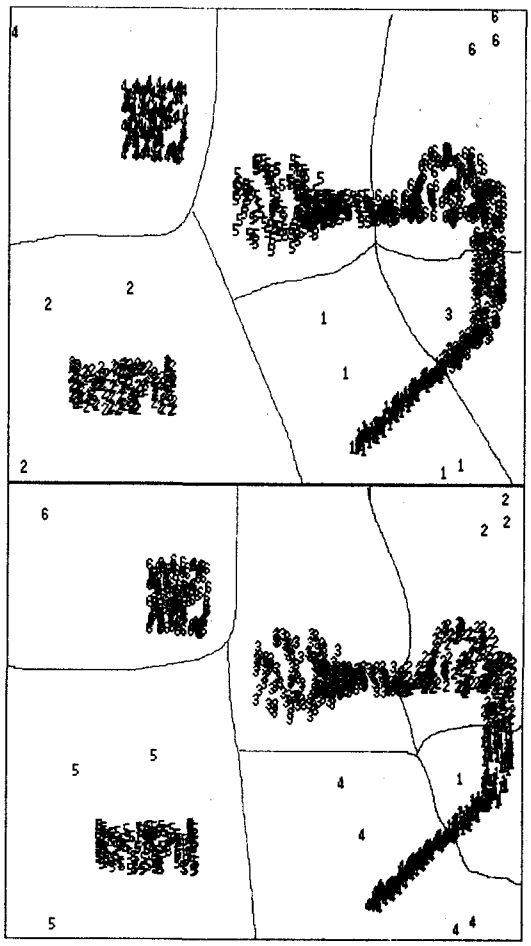

图 4 K-均值算法和改进的遗传算法针对图 2 的样本集的聚类结果

表 1 和表 2 是针对两个模拟数据的实验中 K-均值聚类和文中的遗传空间聚类算法的类间离散度和 J 的值,从中可知 K-均值算法的 J 值在 4 次试验中都不相同,说明 K-均值算法对初始值敏感,不同的初始值收敛于不同的局部极优点;而文中的遗传空间聚类算法每次都在基本相同的最优点处收敛。由此可得出结论:文中的遗传空间聚类算法与 K-均值算法相比,具有较强的全局收敛能力。文中的遗传空间聚类算法兼顾了局部收敛和全局收敛性能,在兼顾局部收敛速度的同时寻找到的空间聚类中心保持了良好的全局分布特性。

表 1 针对图 1 模拟数据的实验中两种聚类的类间离散度和 J 的值

	第 1 次	第 2 次	第 3 次	第 4 次
K-均值	32377.3174	30964.4839	32732.2336	29896.2899
改进的遗传聚类	29876.2899	29876.3012	29876.2892	29876.2876

表 2　针对图 2 模拟数据的实验中两种聚类的类间离散度和 J 的值

	第 1 次	第 2 次	第 3 次	第 4 次
K－均值	42082.3611	41204.2499	36016.7951	36026.7866
改进的遗传聚类	35939.3833	35938.5744	35939.3724	35938.2509

4　算法复杂度分析

本次实验中,K－均值算法消耗时间为 100ms,改进的遗传空间聚类算法消耗时间为 842ms(均采用 4 次实验消耗时间的平均值),说明改进的遗传空间聚类算法相对而言计算速度较慢,这是因为遗传算法是从全局最优的角度来进行空间聚类的。

在改进的遗传空间聚类算法中,估算每个染色体的类内离散度和的复杂度是 $O(nd)$(其中,d 是维数,在对空间数据进行聚类时就是 2),交叉操作的复杂度是 $O(n^2d)$,K－均值优化操作是 $O(nKd)$。由于变异概率较低,等位基因中只有很少一部分做变异操作。而且,K－均值优化操作是一个逐步减少的操作,并在达到一个局部最优的时候不改变染色体编码(除非该染色体编码被变异操作改变)。同样,如果一个染色体编码在上一代的进化中没有变化的话则在当前无需做 K－均值优化操作。事实上,K－均值优化操作只是在最初的进化阶段运行的比较多,在进化的最后阶段,只有染色体编码被变异操作改变时才运行 K－均值优化操作。因此,如果将染色体的变化情况记录下来的话则可以有效地减少计算复杂度。

5　结　论

提出了一个改进的遗传空间聚类算法,该算法结合了遗传算法全局搜索的优点和 K－均值方法局部收敛速度快的特点,克服了传统的 K－均值对初始选值敏感和易陷入局部最优的缺陷,提高了遗传算法的收敛速率,并取得较好的结果。但是该算法依然存在一些缺陷,例如在指定聚类结果数目的情况下,对孤立点数据和"噪音"的处理不是很科学,需要做进一步完善。

参考文献:

[1] Han Jiawei, Kamber M. Data Mining: Concepts and Techniques[M]. 北京:机械工业出版社,2001.

[2] 徐宗本,张讲社,郑亚林. 计算智能中的仿生学:理论与算法[M]. 北京:科学出版社,2003.

[3] Maulik U, Bandyopadhyay S. Genetic Algorithm－based clustering technique[J]. Pattern Recognition, 2000, 33(9): 1455－1465.

[4] 杨善林,倪志伟. 机器学习与智能决策支持系统[M]. 北京:科学出版社,2004.

[5] Krishna K, Murty M N. Genetic K－means Algorithm[J]. IEEE Transactions on System, Man and Cybernetics: Part B, 1999, 29(3): 433－439.

(上接第 70 页)

适应的变异方法或其他方法来解决;

2)佳点集方法能够保证子代的解集保留父代最大特性,但在对解的优化过程中并未找到现有的最优解(429.98)。需要进一步探索佳点集方法的机理与蚁群算法信息素更新方式之间的匹配问题,研究其本质上是否存在的相互影响。

4　结　语

提出了带佳点杂交算子的非均匀窗口蚁群算法。通过缩小蚂蚁的搜索空间有效地改进了蚁群算法中搜索时间过长的缺点。同时,采用佳点交叉算子对蚁群算法的解进行优化,改进解的质量。实验表明佳点交叉算子对解的优化有较好的作用,但对算法各部分所采用的方法的平衡问题还有待于进一步研究。

参考文献:

[1] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transaction on Systems, Man, and Cybernetics, Part B, 1996, 26(1): 28－41.

[2] Stutzlet, Hoos H H. MAX－MIN ant system[M]. Future Generation Computer Systems, 2000, 16(8): 889－914.

[3] 段海滨,王道波,朱家强,等. 蚁群算法理论及应用研究的进展[J]. 控制与决策, 2004, 19(12): 1321－1326.

[4] 段海滨,王道波. 一种快速全局优化的改进蚁群算法及仿真[J]. 信息与控制, 2004, 33(2): 241－244.

[5] 朱庆保,杨志军. 基于变异和动态信息素更新的蚁群优化算法[J]. 软件学报, 2004, 15(2): 185－192.

[6] 全惠云,文高进. 求解 TSP 的子空间遗传算法[J]. 数学理论与应用, 2002, 22(1): 36－39.

[7] 王正志,薄　涛. 进化计算[M]. 长沙:国防科技大学出版社, 2000: 287－296.

[8] 张　铃,张　钹. 佳点集遗传算法[J]. 计算机学报, 2001, 24(9): 1－6.

[9] 丁建立,陈增强,袁著祉. 遗传算法与蚂蚁算法的融合[J]. 计算机研究与发展, 2003, 40(9): 1351－1356.