基于基因表达式编程的多目标自动聚类算法

徐丽丽[1] 许春秀[1] 张 静[1] 齐 峰[2]

[1](山东劳动职业技术学院信息工程系 山东 济南 250022)

[2](山东师范大学商学院 山东 济南 250014)

摘 要 聚类是将物理或抽象对象的集合分成由类似的对象组成的多个类(簇)的过程。同一个簇中的对象彼此相似,而不同簇中的对象差异较大。以基因表达式编程算法为基础,结合新设计的广义聚类代数算子和目标优化函数,提出一种基于基因表达式编程的多目标自动聚类算法(MAGEP-Cluster)。该算法不仅可以自动确定最优聚类的数目,还可以同时基于簇内数据紧凑性和簇间数据连通性两个指标实现数据的有效划分。在三个人工数据集和五个 UCI 数据集上的实验结果表明,与 GEP-Cluster、MOCK 和 VAMOSA 等算法相比,MAGEP-Cluster 具备更好的聚类性能。

关键词 自动聚类 多目标 基因表达式编程 簇间连通性

中图分类号 TP181 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2022.03.040

MULTI-OBJECTIVE AUTOMATIC CLUSTERING ALGORITHM BASED ON GENE EXPRESSION PROGRAMMING

Xu Lili[1] Xu Chunxiu[1] Zhang Jing[1] Qi Feng[2]

[1](*Department of Information Engineering, Shandong Labor Vocational and Technical College, Jinan 250022, Shandong, China*)

[2](*Business College, Shandong Normal University, Jinan 250014, Shandong, China*)

Abstract Clustering is the process of dividing a collection of physical or abstract objects into clusters of similar objects. Objects in the same cluster are similar to each other, while objects in different clusters are quite different. On the basis of gene expression programming algorithm, this paper combines the newly designed generalized clustering algebra operator and objective optimization function and proposes a multi-objective automatic clustering algorithm based on gene expression programming (MAGEP-Cluster). It automatically determined the optimal number of clusters, and reasonably divided all data sets according to the compactness of data in clusters and the connectivity of data between clusters. Experimental results on three artificial datasets and five UCI datasets show that, MAGEP-cluster has better clustering performance than GEP-cluster, MOCK and VAMOSA.

Keywords Automatic clustering Multi-objective Gene expression programming Connectivity of data between clusters

0 引 言

聚类是将一组未标记的对象分配到不同簇中,使得来自同一簇内的对象之间的相似性最大,而来自不同簇间的对象之间的相似性最小[1]。聚类算法目前已被广泛应用于许多领域,包括机器学习、模式识别、图像分析、信息检索、生物信息学、数据压缩和计算机图形学。

传统的单目标聚类算法具有简单、执行速度快且效率高等优点,但同时也存在着算法容易陷入局部最优,需要预先指定最终的聚类数目以及算法适应性低等局限性。与单目标聚类算法仅采用单个聚类标准来对对象进行划分不同,多目标聚类算法可通过同时优

收稿日期:2020-02-16。国家自然科学基金项目(61502283);中国职协研究课题(2019069)。**徐丽丽**,讲师,主研领域:数据挖掘,大数据,知识图谱。**许春秀**,硕士。**张静**,硕士。**齐峰**,副教授。

化多个聚类评价标准来获得聚类结果,这样可以增加算法对大部分聚类结果的鲁棒性[2-3]。其中,进化算法可以对聚类解空间进行全局搜索从而克服传统聚类算法容易陷入局部最优的缺点,因此,近年来基于进化计算的聚类算法逐渐成为科研人员关注的焦点[4-6]。除此之外,传统聚类算法需要预先设定聚类的数目,但是在实际应用中对于待聚类的数据通常缺乏与类别数目有关的先验知识,而基于进化计算的聚类算法借助进化机制实现聚类解空间的随机搜索,无须预先指定聚类数目。本文以基因表达式编程算法(Gene Expression Programming,GEP)为基础,结合设计的广义聚类代数算子和目标优化函数,提出了一种基于基因表达式编程的多目标自动聚类算法(MAGEP-Cluster)。MAGEP-Cluster 算法不仅可以自动确定最优聚类的数目,还可以同时基于簇内数据紧凑性和簇间数据连通性两个衡量指标实现数据的有效划分。

1 相关工作

作为统计学、模式识别、机器学习、数据挖掘和生物信息学等几个领域的基础,聚类的目的是为了确定一组未标记数据的固有分组,其中每个组中的对象在某些相似性标准下是不可区分的。聚类将数据集划分为组(簇),其中组(簇)中的数据元素彼此有很高的相似性,而组(簇)间元素彼此间的相似性很低。

自动多目标聚类的目的就是为了解决需要预先确定最终聚类数目或分区数目的问题。MOCK(具有自动 k-确定的多目标聚类)[7]是一种基于 PESA-II[8]的多目标聚类算法,它被用于优化两个互补的目标函数:总体偏差(测量簇内数据的紧凑性)和连通性(考虑相邻数据项是否放置在相同的簇中)。以往的研究表明,在各种基准数据集中,MOCK 的表现优于传统的单目标聚类算法。该算法适用于超球形或分离的簇,并且可以提供更好的聚类结果,但是 MOCK 在重叠簇上的聚类结果并不能令人满意。文献[9]中提出了一种基于对称性的多目标聚类算法(VAMOSA),该方法以基于模拟退火的多目标优化方法作为底层优化策略,借助聚类中心字符串化的编码规则,并在聚类中使用文献[10]中新提出的点对称距离来代替常规的欧几里德距离。实验结果表明,VAMOSA 算法整体性能优于 MOCK,但是,同样该算法对重叠数据集的聚类没有很好的鲁棒性。基于基因表达式编程算法[11],文献[12-14]提出了一类名为 GEP-Cluster 聚类算法,该类算法都借鉴了遗传进化过程中的某些生物特性,例如小生境等,相关实验结果表明该类算法的性能要好于基于原始 GEP 聚类算法的表现。

文献[15]针对层次聚类算法不足,提出了一种基于 GEP 计算模型的层次聚类算法(GEPHCA),寻找适应度最高的聚类中心。相关仿真实验表明,该算法不仅能够实现自动聚类,而且具有自适应迭代、速度较快、稳定高效等优点。文献[16]通过结合基因表达式编程和已有的串行聚类 DBSCAN 算法提出了一种 GEP-DBSCAN 协作过滤聚类算法来寻找最近邻居,改进基于密度的协作过滤方法,实验表明了算法的有效性以及提高了时间效率。文献[17]针对模糊 C-均值聚类图像分割方法存在的对初始值敏感及抗噪性能差的问题,提出一种结合基因表达式编程与空间模糊聚类的图像分割方法。仿真实验表明,该算法在聚类划分系数、聚类划分熵和峰值信噪比等评价指标上总体性能优于其他比较算法。文献[18]对传统的基于基因表达式编程的聚类算法进行改进,提出了一种新的聚类合并准则解决原算法后期过合并的问题,同时改进算法编码方式并在进化过程中引入多目标来解决聚类问题。仿真结果表明,新算法的性能优于对比算法。文献[19]提出了一种改进的融合基因表达式编程和 k 均值算法的自动聚类算法,在 GIS 物流选址优化问题中实验结果表明,所提算法具有收敛速度快和聚类精度高的优势。

本文以 GEP-Cluster 聚类算法为研究对象,通过改进聚类代数算子、引入新的目标优化函数和设计聚类中心合并规则,提出了一种新的基于基因表达式编程(GEP)自动多目标聚类算法,称为 MAGEP-Cluster。该算法不仅可以在没有任何先验知识的情况下自动确定聚类的数目,而且与其他聚类算法相比性能更优。

2 MAGEP-Cluster 算法

聚类可以被视为一个多目标优化问题。MAGEP-Cluster 算法不仅可以自动确定聚类的数目,还可以基于簇内数据的紧凑性和簇间数据的连通性两个标准实现所有数据的有效划分。本节将对 MAGEP-Cluster 算法的核心部分进行详细阐述。

2.1 广义聚类代数算子

聚类算子是实现 MAGEP-Cluster 算法的关键,同时也是 GEP 染色体的核心构成元素,其设计的合理性直接决定了自动多目标聚类算法的效能。结合文献[20-21]中关于聚类代数算子的相关研究成果,这里

提出了两种分别用于分割和聚合的广义聚类代数算子,将其命名为$\cup_n$和$\cap_n$,如图 1 所示。

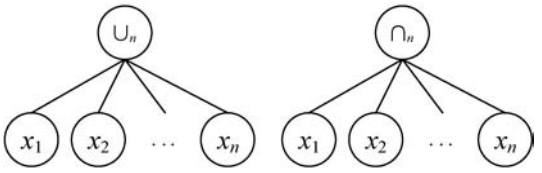


图 1　广义聚类分割算子$\cup_n$和广义聚合算子$\cap_n$

假设O_x, O_y, O_z分别是簇C_x、C_y和C_z的d维度中心点,其中,$O_x = (x_1, x_2, \cdots, x_d)$,$O_y = (y_1, y_2, \cdots, y_d)$,$O_z = (z_1, z_2, \cdots, z_d)$。

当$n=3$时,广义聚类分割算子$\cup_n$和广义聚合算子$\cap_n$的计算规则如下:

(1) $\cup_3(O_x, O_y, O_z) = \{O_x, O_y, O_z\}$

(2) $\cap_3(O_x, O_y, O_z) = \{O_{xyz}\}$

式中:O_{xyz}表示三个聚类中心O_x、O_y、O_z的平均值。在情况(1)中,等号右边为广义分割算子的运算结果,其含义表示以O_x、O_y、O_z为中心的三个独立的簇C_x、C_y和C_z,即它们仍然保持分割状态;在情况(2)中,三个子簇C_x、C_y和C_z被合并成一个新的簇C_{xyz},它的簇中心为$O_{xyz} = \left(\frac{x_1+y_1+z_1}{3}, \frac{x_2+y_2+z_2}{3}, \cdots, \frac{x_d+y_d+z_d}{3}\right)$。

2.2　GEP 染色体编码

MAGEP-Cluster 算法采用了与 GEP 相同的染色体编码方式,每条染色体由头部和尾部构成。头部可以包含函数节点或者终端节点,而尾部只能包含终端节点。MAGEP-Cluster 算法中,函数节点集合为$F = \{\cup_2, \cap_2, \cup_3, \cap_3, \cdots, \cup_n, \cap_n\}$,终端节点集合$T = \{x_1, x_2, \cdots, x_m\}$,$i \in [1, m]$,其中$x_i$表示数据集中第$i$个对象,$m$代表数据集中对象的个数。

根据 GEP 染色体编码的特点,这里使用F和T的元素组成 GEP 染色体的头部,使用T中的元素构建 GEP 染色体的尾部。GEP 染色体在初始化时,若基因位位于染色体头部,则随机从集合$F \cup T$中随机选取一个元素作为该基因位的取值;若基因位位于染色体尾部,则随机从集合T中随机选取一个元素作为该基因位的取值。

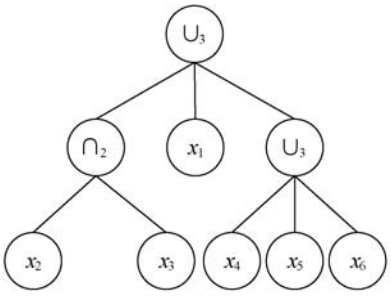
另外,特别需要注意的是,GEP 染色体中的实数编码信息表示某个类别的聚类中心,而不是某单个数据对象,所以在染色体进化过程中允许有重复的终端节点。

假设h表示 GEP 染色体头部的长度,t表示 GEP 染色体尾部的长度。由于聚类算子的最大参数量是h,那么$t = h(n-1)+1$。从聚类表达树中可以看出,在最极端的情况下,一条 GEP 染色体最多可以表达$h(n+1)$个聚类中心。

图 2 展示了 MAGEP-Cluster 中 GEP 染色体三种表达形式。图 2(a)展示了一条 GEP 染色体($h=4$)的编码串形式,图 2(b)展示了 GEP 染色体对应的 GEP 聚类表达树,图 2(c)展示了 GEP 聚类表达树对应的 GEP 聚类操作算子表达式。

$\cup_3 \cap_2 x_1 \cup_3 x_2 x_3 x_4 x_5 x_6$

(a) GEP 染色体编码串



(b) GEP 聚类表达树

$\cup_3(\cap_2(x_2, x_3), x_1, \cup_3(x_4, x_5, x_6))$

(c) GEP 聚类代数算子表达式

图 2　MAGEP-Cluster 中 GEP 染色体三种表现形式

2.3　GEP 聚类表达树解码

由图 2 可知,GEP 染色体编码串到 GEP 聚类表达树的转换是通过从左到右和从上到下的广度优先遍历操作来实现的。而 GEP 聚类表达树的解码即获得对应聚类代数算子表达式是通过从左至右和从上到下的深度优先遍历操作来实现的,并且在解码过程可根据聚类代数算子的定义对 GEP 聚类表达树中包含的聚类中心进行聚合和分割。

对 GEP 染色体对应的 GEP 聚类表达树解码的目的是为了了解染色体信息中携带的聚类中心信息。在实际解码时,可采取递归方式遍历聚类表达树来完成解码过程,其中主要包含聚合、分割和集中数据等操作。

解码过程的主要步骤如下:(a) 从聚类表达树的根节点开始处理。如果聚类表达树的根节点是$\cup_n$,则将以该节点的子节点为根节点的子树从左至右依次放置到n个不同的集合中,实现多个不同簇间的分割;如果聚类表达树的根节点是$\cap_n$,则从左至右依次遍历其所有子树,将所有子树中的数据放入同一簇中,计算所有子树根节点的平均值,将该平均值作为聚类中心点。注意,真正的聚类中心是由距离聚类中心点最近的数据对象来替代表示的。(b) 如前所述,递归地解码所有的子树。

GEP 聚类表达树解码算法的具体流程,如算法 1

所示。

算法 1 聚类表达树解码(寻找聚类中心)

输入:聚类表达树

输出:聚类中心

(1) 如果聚类表达树的根节点非空,则跳转到根节点;

(2) 判断根节点类型:

(3) 如果根节点为 $\cup_n$,则依次将该节点的所有子树放置到集合 I 中;

(4) 遍历集合 I 中的所有子树并依次进行递归解码;

(5) 如果根节点为 $\cap_n$,则依次遍历解码其所有子树;

(6) 聚类中心点←计算子树中的所有数据点的平均值;

(7) 寻找距离聚类中心点最近的数据点作为聚类中心;

(8) 返回所有的聚类中心;

(9) 算法结束。

2.4 GEP 染色体遗传操作

与经典 GEP 算法类似,MAGEP-Cluster 采用了三种遗传操作:选择、交叉和变异。

(1) 选择操作:主要根据 GEP 染色体适应度值借助双人锦标赛方法来实现,同时精英选择策略也被使用,即每代种群最优个体自动保留到新种群中。

(2) 交叉操作:主要采用了单点交叉和双点交叉两种方式。大量实验结果表明,交叉概率设为 0.87 时 MAGEP-Cluster 呈现出更好的性能。由于交叉点是随机选取的,一旦产生的染色体后代是非法的,算法规定本次交叉操作将取消。

(3) 变异操作:是进化过程中跳出局部极值点的有效方式。大量实验结果表明,变异概率设为 0.024 时 MAGEP-Cluster 呈现出更好的性能。实际操作时,随机选择染色基因位进行变异,根据基因位所处位置分为两种变异操作:如果基因位在头部,则在集合 $F \cup T$ 中随机选择一个元素代替当前基因位的元素;如果基因位在尾部,则在集合 T 中随机选择一个元素代替当前基因位中的元素。如果变异后的染色体是非法,则重新随机选择基因位执行变异操作,直到产生合法染色体后代为止。

2.5 聚类中心合并算法

大量实验结果表明,虽然 MAGEP-Cluster 在解决多目标聚类问题方面效果很好,但有时最终获得的聚类中心不一定是最优的,例如会出现聚类中心过多的现象。因此,在通过 MAGEP-Cluster 获得数据的聚类中心后,有必要设计一种聚类中心合并算法。

通过研究相邻簇中数据点的相互关系,本文提出了一种聚类中心自动合并算法,算法流程如算法 2 所示。

算法 2 聚类中心自动合并算法

输入:MAGEP-Cluster 聚类中心列表 L

输出:合并后的聚类中心列表 L

(1) 从 L 中随机选择一个聚类中心 C_i,根据剩余聚类中心到 C_i 的距离对 L 进行排序,得到一个新的排序聚类中心列表 $L_s = \{C_i, C_j, \cdots\}$,其中 C_j 是距离 C_i 最近的簇,以次类推;

(2) 从 C_i 中选择一个最接近 C_j 的点 x_i;

(3) 从 C_j 中选择一个最接近 C_i 的点 x_j;

(4) 计算 x_i 和 x_j 间的欧氏距离 D_{ij};

(5) 计算 C_i 的平均距离 D_i;

(6) 计算 C_j 的平均距离 D_j;

(7) 如果 $D_{ij} \leqslant D_i$ 或者 $D_{ij} \leqslant D_j$,则将 C_i 和 C_j 合并为 C_{ij},同时替换 L 中的 C_i 和 C_j;

(8) 重复上述过程,直到 L 中没有聚类中心可以合并;

(9) 输出合并后的聚类中心列表 L;

(10) 算法结束。

2.6 聚类目标函数

多目标聚类算法性能在很大程度上取决于聚类目标函数的选择,这些目标函数应尽可能满足簇内紧凑,簇间稀疏的总体要求。本文考虑选取两个互补的目标函数:簇内紧凑性函数和簇间连接性函数作为多目标聚类算法中 GEP 染色体适应度值。

簇内紧凑性函数用来度量簇内所有数据点的总对称距离,即计算数据对象与其对应的聚类中心之间的总对阵距离。簇内紧凑性函数定义为:

$$Comp(C) = \sum_{i=1}^{k}\sum_{j=1}^{n}\left(1 - \exp\left(-\frac{\|x_{ij} - \mu_i\|^2}{\beta_i}\right)\right) \quad (1)$$

式中:k 表示簇总数;n 代表第 i 个簇中所有数据点的个数;μ_i 表示第 i 个簇的中心;x_{ij} 表示第 i 个簇的第 j 个数据点,$\beta_i = \dfrac{\sum_{j=1}^{n}\|x_{ij} - \mu_i\|^2}{n}$。

簇间连接性函数用来评估相邻数据点被放置在同一个簇中的重要程度,定义为:

$$Conn(C) = \sum_{i=1}^{k}\frac{1}{n}\sum_{h=1}^{|C_i|}\left(\sum_{j=1}^{L} x_{h,nn_{hj}}\right) \quad (2)$$

式中:nn_{hj} 表示数据对象 h 的第 j 个最近邻数据对象,n 代表待聚类所有数据点的个数,$|C_i|$ 表示簇 C_i 中的数据点的个数,L 代表为簇间连接性做出贡献的邻居数量,$x_{h,nn_{hj}} = \begin{cases}\dfrac{1}{j} & \nexists\ C_i : x_h \in C_i \wedge x_{nn_{hj}} \in C_i \\ 0 & \text{其他}\end{cases}$。

聚类目标函数包括簇内紧凑性函数和簇间连接性函数的一个重要原因是它们能够平衡彼此增加或减少簇总数的趋势,从而不必预先指定聚类簇个数 k。与紧凑性相关的目标值必然随着簇个数的增加而提高,而连通性恰好相反,两者之间的相互作用对于保持簇个数的动态变化至关重要。

2.7　MAGEP-Cluster 算法流程

本节将对所提出的算法 MAGEP-Cluster 算法流程进行详细描述,具体的算法流程如算法 3 所示。MAGEP-Cluster 算法采用了与经典多目标遗传算法:NSGA-II[22] 类似的进化流程,其中非支配排序和拥挤度计算规则与 NSGA-II 完全相同。

算法 3　MAGEP-Cluster

输入:数据集 D,染色体头部长度 h,种群大小 N,最大进化代数 G_{max},最大簇个数 K_{max},贡献簇间连通性的邻居个数 L,变异概率 P_m,选择概率 P_s,交叉概率 P_c。

输出:数据集 D 的聚类中心列表。

(1) 根据数据集 D 的特点及实验要求,设置参数 h、N、G_{max}、K_{max}、L、P_m、P_c、L_{max}的初始值。

(2) 根据 GEP 染色体编码规则,创建初始种群 $P_t(t=0)$,种群规模为 N。

(3) 根据 GEP 染色体解码规则,结合两个聚类目标函数和算法 1,对种群 P_t进行非支配排序和拥挤度计算。

(4) 依概率 P_c 和 P_m 对种群 P_t进行单点交叉操作、双点交叉操作和变异操作,生成一个新的规模为 N 的种群 Q_t。

(5) 构建中间种群 $R_t = P_t \cup Q_t$。

(6) 根据 GEP 染色体解码规则,结合两个聚类目标函数和算法 1,对种群 R_t 进行非支配排序和拥挤度计算,借助双人锦标赛选择法和精英保留策略,从中间种群 R_t 中选择 N 个染色体生成新种群 $P\_(t+1)$,令 $t = t+1$。

(7) 如果 $t \leqslant G_{max}$,则返回第(4)步,否则转向第(8)步。

(8) 取当前种群的最优 GEP 染色体执行算法 2,返回最优聚类中心列表。如果满足实验要求,则转第(9)步,否则转第(2)步。

(9) 算法结束。

3　实验与结果分析

3.1　数据集

为了验证 MAGEP-Cluster 算法的有效性,这里选择了 3 个人工数据集和 6 个 UCI 标准数据集来比较 MAGEP-Cluster 和其他三个聚类算法的性能。

虽然人工数据集看起来很简单,但都包含一些特殊特征,例如 Data_5_2 中部分数据高度重叠,Data_4_3 中存在部分噪声点(仅在两个不同的簇之间),这些特殊特征会对聚类目标优化函数产生干扰,影响聚类算法的最终效果。因此,选取人工数据集目的就是为了验证算法能否在处理这些具有特殊特征数据集时具有良好的鲁棒性。

表 1 中给出了数据集的简要描述,主要涉及总样本数据点数(n),分类簇个数(k)和样本数据属性个数(d)三个方面。

表 1　人工数据集和 UCI 数据集

人工数据集	n	K	D
Data_3_2	76	3	2
Data_5_2	250	5	2
Data_4_3	400	4	3
Iris	150	3	4
BreastCancer	569	2	9
Newthyroid	215	3	5
LungCance	32	3	56
Wine	178	3	13
LiverDisorder	345	2	6

3.2　性能评价

为了评价算法性能,除了选取聚类数目作为衡量标准外,还引入了簇精度(ClusterAccuracy,CA)作为评价 4 种算法性能的指标,CA 主要用来评估聚类结果中正确分类的数据点(由算法获得的聚类标签)所占比例,其计算式为:

$$CA = \sum_{i=1}^{K} \frac{\sum_{j=1}^{|C_i|} \delta(s_j, r_j)}{|C_i|} \tag{3}$$

式中:r_j,s_j分别表示数据 $x_j \in C_i$所对应的聚类标签和真实标签;$|C_i|$是第 i 个簇中的数据点个数;δ 表示指示函数,定义如下:

$$\delta(x,y) = \begin{cases} 1 & x = y \\ 0 & \text{其他} \end{cases} \tag{4}$$

CA 本质上表示的是通过聚类获得的标签与实际标签相同的数据点的数量。显然,CA 越大,聚类效果越好。

3.3　结果分析

为了更好验证 MAGEP-Cluster 算法的性能,这里分别选取了另外三种算法:GEP-Cluster、Mock 和 VAMOSA 作为比较对象。由于每种算法的初始种群是随机生成的,为了保证比较的公平性,对于每个数据集,每个算法独立运行 200 次,以使聚类结果具有可比性,然后分别计算每种算法得到的最优簇数(K)和簇精度(CA)的平均值。另外,由于 MAGEP-Cluster 与 GEP-Cluster 均使用类似的 GEP 染色体编码和解码规则,在实验时,规定两个算法中 GEP 染色体的头部和尾部相同,唯一区别在于两个算法使用的函数集不同,MAGEP-Cluster 采用了广义聚类代数算子即算子的操作数大于等于 2,而 GEP-Cluster 采用的是二元聚类代数算子即

算子操作数只能为2。

表2 和表3 分别给出了四种算法在人工数据集和UCI 数据集上的 K 和 CA 比较结果。

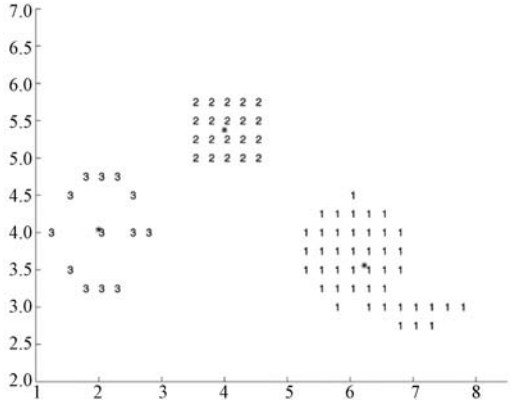
表2 人工数据集最优簇数(K)和簇精度(CA)

UCI 数据集	分类数	MAGEP-Cluster		GEP-Cluster		MOCK		VAMOSA	
		K	CA	K	CA	K	CA	K	CA
Data_3_2	3	3.04	0.95	3.48	0.91	3.91	0.86	3.52	0.90
Data_5_2	5	4.99	0.98	4.67	0.89	3.86	0.77	4.45	0.87
Data_4_3	4	4.02	0.98	4.23	0.94	4.49	0.89	4.36	0.92

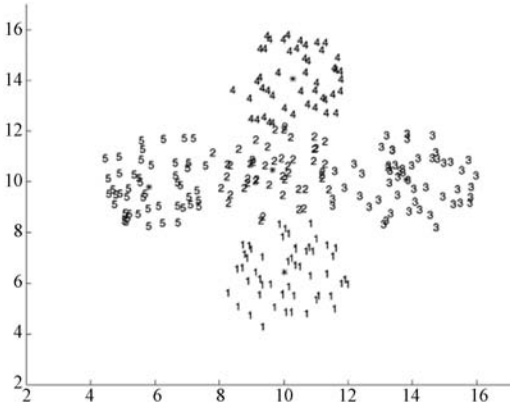
表3 UCI 数据集最优簇数(K)和簇精度(CA)

人工数据集	分类数	MAGEP-Cluster		GEP-Cluster		MOCK		VAMOSA	
		K	CA	K	CA	K	CA	K	CA
Iris	3	2.86	0.93	2.80	0.90	2.51	0.82	2.73	0.88
BreastCancer	2	2.08	0.99	2.15	0.95	2.42	0.89	2.36	0.93
Newthyroid	3	3.11	0.94	3.20	0.90	3.47	0.88	3.39	0.90
LungCance	3	2.95	0.92	2.84	0.88	2.75	0.81	2.79	0.87
Wine	3	3.18	0.95	3.22	0.92	3.30	0.86	3.21	0.90
LiverDisorder	2	1.93	0.96	1.86	0.92	1.72	0.79	1.79	0.82

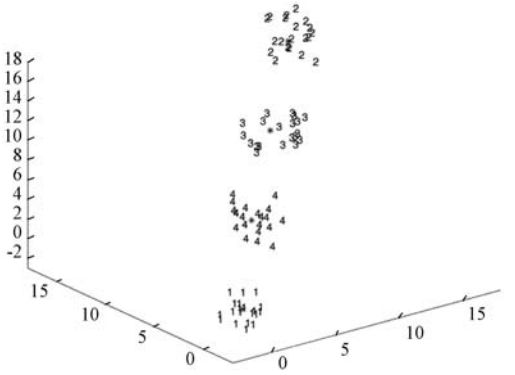
MAGEP-Cluster 在三个人工数据集上的聚类结果如图3 所示。



(a) MAGEP-Cluster 在 Data_3_2 上的聚类结果



(b) MAGEP-Cluster 在 Data_5_2 上的聚类结果



(c) MAGEP-Cluster 在 Data_4_3 上的聚类结果

图3 人工数据集聚类结果

如算法2 和算法3 所示,通过本文提出的 MAGEP-Cluster 算法获得的聚类数的平均值最接近每个数据集中的实际聚类数,与 GEP-Cluster、MOCK 和 VAMOSA 相比,MAGEP-Cluster 算法在聚类方面具明显的优势。此外,从算法2 和算法3 中还可以看出,对于所有的数据集来说,所提出的 MAGEP-Cluster 算法产生的聚类精度(CA)的平均值高于 GEP-Cluster、MOCK 和 VAMOSA 的平均值,表明 MAGEP-Cluster 可以为不同的数据集找到更好的聚类分区。

4 结 语

本文通过改进聚类代数算子、引入新的目标优化函数和设计聚类中心合并规则,提出了一种新的基于基因表达式编程(GEP)自动多目标聚类算法,称为 MAGEP-Cluster。所提出的 MAGEP-Cluster 可以通过优化簇内数据紧凑性和簇间数据连接性两个目标函数来自动计算簇的最佳数量并提高了聚类算法的准确性。实验部分分别在3 个人工数据集和5 个 UCI 数据集,比较了 MAGEP-Cluster 与 GEP-Cluster、MOCK 和 VAMOSA 的聚类性能。实验结果表明,MAGEP-Cluster 算法能够有效地计算最优聚类数目,并在一定程度上提高了最终聚类结果的准确性。MAGEP-Cluster 与 GEP-Cluster 相比之下,除了使用的函数集不同外,其他部分相似性很高,但在使用时前者却表现出了更好的鲁棒性,其中一个很重要的原因在于 MAGEP-Cluster 采用了广义聚类算子,但该算子对 MAGEP-Cluster 算法具体的影响机制尚不明确,因此,广义聚类算子的影响机制将成为未来研究工作的重点。

参 考 文 献

[1] Ni Y, Du X, Xie D, et al. A multi-objective cluster algorithm based on GEP[C]//Proceedings of the 2014 International Conference on Cloud Computing and Big Data. IEEE,

2014: 33 – 38.
[2] Jose-Garcia A, Gomez-Flores W. Automatic clustering using nature-inspired metaheuristics: A survey[J]. Applied Soft Computing, 2016, 41: 192 – 213.
[3] Handl J, Knowles J. Evidence accumulation in multiobjective data clustering[C]//International Conference on Evolutionary Multi-Criterion Optimization. Springer, 2013: 543 – 557.
[4] Babu G P, Murty M N. Clustering with evolution strategies [J]. Pattern Recognition, 1994, 27(2): 321 – 329.
[5] Murthy C A, Chowdhury N. In search of optimal clusters using genetic algorithms[J]. Pattern Recognition Letters, 1996, 17(8): 825 – 832.
[6] Bandyopadhyay S, Maulik U. An evolutionary technique based on K-means algorithm for optimal clustering in RN [J]. Information Sciences, 2002, 146(1/4): 221 – 237.
[7] Handl J, Knowles J. An evolutionary approach to multiobjective clustering[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(1): 56 – 76.
[8] Corne D W, Jerram N R, Knowles J D, et al. PESA-II: Region-based selection in evolutionary multiobjective optimization[C]//Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. ACM, 2001: 283 – 290.
[9] Saha S, Bandyopadhyay S. A symmetry based multiobjective clustering technique for automatic evolution of clusters[J]. Pattern Recognition, 2010, 43(3): 738 – 751.
[10] Bandyopadhyay S, Saha S. GAPS: A clustering method using a new point symmetry-based distance measure[J]. Pattern recognition, 2007, 40(12): 3430 – 3451.
[11] Ferreira C. Gene expression programming: A new adaptive algorithm for solving problems[EB]. arXiv: cs/0102027, 2001.
[12] 陈瑜,唐常杰,叶尚. 基于基因表达式编程的自动聚类方法[J]. 四川大学学报(工程科学版),2007,39(6):107 – 112.
[13] 姜代红,张三友. 基于基因表达式编程的 K 均值自动聚类算法[J]. 计算机仿真,2010,27(12):216 – 220.
[14] Lin Y, Hong P. Niche gene expression programming based on clustering model[C]//Proceedings of the Workshop on Intelligent Information Technology Application. ACM, 2007: 10 – 13.
[15] 姜代红,尹洪胜,张三友. 采用基因表达式编程的自适应层次聚类方法[J]. 华侨大学学报(自然科学版),2018,39(3):435 – 438.
[16] 蔡宏果,元昌安. 一种基于基因表达式编程的串行聚类算法并行化研究[J]. 中南民族大学学报(自然科学版),2017,36(4):112 – 115.
[17] 李婷婷,江朝晖,饶元,等. 结合基因表达式编程与空间模糊聚类的图像分割[J]. 中国图象图形学报,2017,22(5):575 – 583.
[18] 陈超. 基因表达式编程优化算法及其在聚类分析中的应用[D]. 西安:西安电子科技大学,2013.
[19] 姜代红,张三友. 基于基因表达式编程的 K 均值自动聚类算法[J]. 计算机仿真,2010,27(12):216 – 220.
[20] Ni Y, Du X, Xie D, et al. A multi-objective cluster algorithm based on GEP[C]//International Conference on Cloud Computing & Big Data. IEEE, 2014.
[21] Chen Y, Tang C, Zhu J, et al. Clustering without prior knowledge based on gene expression programming[C]//International Conference on Natural Computation. IEEE, 2007.
[22] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182 – 197.

(上接第 246 页)

[9] Srichandan S, Kumar T A, Bibhudatta S. Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm[J]. Future Computing and Informatics Journal, 2018, 3(2): 210 – 230.
[10] Abdullahi M, Ngadi M A, Dishing S I, et al. An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment[J]. Journal of Network and Computer Applications, 2019, 133: 60 – 74.
[11] 王虎,雷建军,万润泽. 基于改进的粒子群优化的云计算资源调度模型[J]. 华中师范大学学报(自然科学版),2018,52(6):788 – 791.
[12] Liu W, Shi C, Yu H, et al. Task scheduling of an improved cuckoo search algorithm in cloud computing[J]. International Journal of Performability Engineering, 2019, 15(7): 1965 – 1975.
[13] 张鑫帅,刘俊,罗世彬. 基于改进多目标布谷鸟搜索算法的翼型气动优化设计[J]. 航空学报,2019,40(6):49 – 62.
[14] Huang C L, Jiang Y Z, Yin Y, et al. Multi objective scheduling in cloud computing using MOSSO[C]//2018 IEEE Congress on Evolutionary Computation (CEC), 2018.
[15] Ebadifard F, Babamir S M. A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment[J]. Concurrency and Computation: Practice and Experience, 2018, 30(12): 1 – 16.
[16] Reddy G N, Kumar S P. Modified ant colony optimization algorithm for task scheduling in cloud computing systems [M]//Smart Intelligent Computing and Applications. Springer, 2019: 357 – 365.
[17] Vila S, Guirado F, Lerida J L, et al. Energy-saving scheduling on IaaS HPC cloud environments based on a multi-objective genetic algorithm[J]. The Journal of Supercomputing, 2019, 75(3): 1483 – 1495.