Journal of System Simulation

Volume 28 | Issue 4 Article 6

7-2-2020

GEP Automatic Clustering Algorithm with Dynamic Penalty Factors

Chen Yan
College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China;

Kangshun Li
College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China;

Yang Lei
College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China;

GEP Automatic Clustering Algorithm with Dynamic Penalty Factors

Abstract

Abstract: Various problems such as sensitive selection of initial clustering center, easily falling into local optimal solution, and determining numbers of clusters, still exist in the traditional clustering algorithm. *A GEP automatic clustering algorithm with dynamic penalty factors was proposed. This algorithm combines penalty factors and GEP clustering algorithm, and doesn't rely on any priori knowledge of the data set. And a dynamic algorithm was proposed to generate the penalty factors according to the distribution characteristics of different data sets, which is a better solution for the impact of isolated points and noise points.* According to four dataset, penalty factors' effect was tested. Base on the result, a formula to generate penalty factors was proposed. The penalty factor calculated from the formula was used in clustering of the standard data set Iris. The experimental result shows that the efficiency and accuracy of the algorithm are good.

Keywords

dynamic penalty factor, GEP, clustering algorithm, machine learning

Recommended Citation

Chen Yan, Li Kangshun, Yang Lei. GEP Automatic Clustering Algorithm with Dynamic Penalty Factors[J]. Journal of System Simulation, 2016, 28(4): 806-814.

第 28 卷第 4 期 系统仿真学报© Vol. 28 No. 4
2016 年 4 月 Journal of System Simulation Apr., 2016

加入动态惩罚因子的 GEP 自动聚类算法

陈琰，李康顺，杨磊
（华南农业大学数学与信息学院，广东 广州 510642）

摘要：传统聚类算法存在对初始聚类中心敏感、易陷入局部最优解、及需要事先确定聚类个数等问题。*提出一种加入动态惩罚因子的 GEP(Gene Expression Programming)自动聚类算法，该算法将惩罚因子与 GEP 聚类算法结合，不需任何先验知识，可自动进行簇的划分。并进一步提出惩罚因子的动态生成算法，可根据不同数据集动态生成满足其分布特征的惩罚因子，更好地解决孤立点或噪声点的影响。*实验选择四组自构造数据集测试惩罚因子对聚类的影响，依据测试结果进行惩罚因子的建模，将该惩罚因子模型应用于标准数据集 Iris 上。实验结果表明：算法具有较高的效率和精确度。

关键词：动态惩罚因子；基因表达式程序设计；聚类算法；机器学习
中图分类号：TP311 文献标识码：A 文章编号：1004-731X (2016) 04-0806-09

GEP Automatic Clustering Algorithm with Dynamic Penalty Factors

Chen Yan, Li Kangshun, Yang Lei

(College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China)

Abstract: Various problems such as sensitive selection of initial clustering center, easily falling into local optimal solution, and determining numbers of clusters, still exist in the traditional clustering algorithm. *A GEP automatic clustering algorithm with dynamic penalty factors was proposed. This algorithm combines penalty factors and GEP clustering algorithm, and doesn't rely on any priori knowledge of the data set. And a dynamic algorithm was proposed to generate the penalty factors according to the distribution characteristics of different data sets, which is a better solution for the impact of isolated points and noise points.* According to four dataset, penalty factors' effect was tested. Base on the result, a formula to generate penalty factors was proposed. The penalty factor calculated from the formula was used in clustering of the standard data set Iris. The experimental result shows that the efficiency and accuracy of the algorithm are good.

Keywords: dynamic penalty factor; GEP; clustering algorithm; machine learning

引言

随着科学技术的日益发展，我们的生活中产生了越来越多的数据信息，如何利用这些数据信息，如何从这些数据信息中提取出我们想要的信息，是很多学者都在研究的问题，该领域被称为数据挖掘[1-2]。数据挖掘是一个跨越多个学科的研究领域，它包括了数据库技术[3]、机器语言学习[4]、模式识别[5]、神经网络[6]、人工智能[7]等多个学科。数据挖掘领域的一个重要研究方向是无监督学习，无监督学习中一种典型的方法是聚类[8-9]。聚类过程定义为：给定 n 维空间中的 M 个数据，根据数据点间的相似程度，把这 M 个点分成 k 个类，即相似

收稿日期：2014-11-26 修回日期：2015-05-29；
基金项目：2014 年度国家星火计划(2014GA780012)，广东省自然科学基金(2014A030313454)；
作者简介：陈琰(1977-)，女，回族，宁夏，博士，讲师，研究方向为演化计算，深度学习；李康顺(通讯作者 1962-)，男，江西，博士后，教授，研究方向为智能计算、遗传程序设计、基因表达式程序设计；杨磊(1978-)，男，河南，博士生，讲师，研究方向为演化计算。

的数据样本在同一类中，相异的数据样本在不同类中，目的是使得同一类内的数据相似度尽可能大，不同类间的数据相似度尽可能小。

目前聚类主要有以下几大类方法：基于划分的方法、基于层次的方法、基于密度的方法、基于网格的方法以及基于模型的方法等[10]。其中，K-Means (K-Means Clustering，KMC)算法是一种基于划分思想的聚类算法，该方法由于比较简单、高效，而且需要设置和调整的参数少等优点被应用到了很多领域。但也存在对初始聚类中心选择敏感、易陷入局部最优解、全局搜索能力较差等问题，尤其是 K-Means 算法需要事先根据先验知识确定聚类个数，在实际的应用中存在一定的局限性。另外，Murthy 和 Chowdhury 等在 1996 年提出了基于 GA(Genetic Algorithms)[11-12]的聚类算法[13]，该算法采用二进制编码，每个数据占个体染色体的一位。但该算法因染色体长度的限制，只适合数据集较小的聚类问题。为了解决该问题，Bandyopadhyay 和 Maulik 在 2002 年提出了使用聚类中心点来代替染色体中个体的改进 GA 聚类算法[14]，该算法使得染色体长度大大减小，因此能够处理比较大的数据集。但该算法同样要预先估计中心点的个数，因而也有其局限性。后来，陈瑜、唐常杰等提出了一个自动合并簇的基因表达式聚类算法。姜代红、张三友等从分析基因表达式编程和 K-Means 算法的特点出发，融合了 2 种算法优点，实现了在未知簇划分信息的情况下对数据集自动进行聚类分析。但是上述 2 种算法都属于划分类的聚类算法，因此对于孤立点的影响还不能很好的消除。

针对以上方面不足，本文提出了一种加入动态惩罚因子的 GEP(Gene Expression Programming)[15-16]自动聚类算法(DP-GEPCA，Dynamic Penalty Factors-GEP Automatic Clustering Algorithm)。该算法与已有聚类算法相比，具有以下方面特点：1) 将惩罚因子与 GEP 聚类算法结合，完全不需要数据集的任何先验知识，可以自动进行簇的划分；2) 提出了一种动态惩罚因子，在能够自动完成簇的划分前提下，还能够很好的解决孤立点或噪声点的影响；3) 采用了 GEP 算法，提高了聚类算法的寻优能力，对于大规模数据，具有搜索精度高等特点。

1 传统 GEP-Cluster 算法

传统 GEP-Cluster 算法是一种基于划分的聚类算法，算法采用了 GEP 算法思想，完成无监督的数据点聚类过程。对应聚类问题的特殊性，GEP 中染色体采用整数编码，由单基因构成。

由于 GEP 算法的优点，该算法可以解决传统聚类算法对初始聚类中心选择敏感、易陷入局部最优解、全局搜索能力较差等问题。但是该算法仍然需要预先给定聚类个数，否则聚类的结果将永远是实验所给定的最大聚类个数，并且该算法在解决孤立点或噪声点的影响等问题方面也存在不足。因此，本文提出了一种加入动态惩罚因子的 GEP 聚类算法，很好的解决了上述两方面问题。

2 加入动态惩罚因子的 GEP 聚类算法 DP-GEPCA

2.1 加入动态惩罚因子的基本思路

由于聚类算法是一种无监督算法，主要存在两方面困难：一个是在无任何先验知识的情况下自动进行簇的划分；二是解决孤立点或噪声点的影响问题。为了解决聚类算中聚类个数的最优自动产生，本文引入了惩罚因子的思想完成聚类算法中聚类个数的最优产生。但由于惩罚因子的设置与具体数据集的分布特征密切相关，实践中人为选择惩罚因子是极其困难的，尤其是在训练数据含孤立点或有噪声的情况下。因此，本文在加入惩罚因子的前提下进一步提出了惩罚因子的动态生成算法，不仅可以完成自动聚类过程，而且加入的惩罚因子也会依据待聚类数据集的分布特征动态产生，更好的解决孤立点或噪声点的影响。

第 28 卷第 4 期 系统仿真学报 Vol. 28 No. 4
2016 年 4 月 Journal of System Simulation Apr., 2016

惩罚因子的合理设置非常关键，不能过大也不能过小，要设置的恰到好处。本文对惩罚因子的定义是：惩罚因子的取值越小，表示越倾向于通过增加簇的个数来减小总体平方误差，增加簇的意愿就越强；惩罚因子的取值越大，表示越倾向于通过减少簇的个数来满足人们希望聚类个数尽可能少的期望，即增加簇的意愿就越弱。所谓恰到好处的惩罚因子，就是在没有预先设定聚类个数的前提下，最终聚类个数与经验值相符的惩罚因子，如能令经典聚类测试数据集 iris 自动聚成 3 类、glass 自动聚成 6 类的惩罚因子。

根据实际情况可知，该惩罚因子的大小具体与待聚类数据集的分布密切相关。为了能够动态产生该数据集最恰当的惩罚因子，本文基于聚类的 2 个准则，即最小化类内距及最大化类间距，选择了 4 个标识数据集离散程度及聚合程度的重要物理量：平均误差、方差、包含 3/4 数据点的半径及包含 1/4 数据点的半径作为生成惩罚因子的统计量来完成惩罚因子的动态产生，分别如下：

1) 平均误差：平均误差是各个数据点与中心之差的平均数。反映了整体数据点的偏移程度。

2) 方差：方差是各个数据点与中心之差的平方的平均数。在概率论和数理统计中，方差用来度量随机变量和其数学期望之间的偏离程度。

3) 包含 3/4 数据点的半径：以中心为球体中心，以最小的半径包含 3/4 的数据点。

4) 包含 1/4 数据点的半径：以中心为球体中心，以最小的半径包含 1/4 的数据点。

动态惩罚因子的产生包含以下步骤：

1) 数据点的归一化

由于数据点每个维度表示的信息有所不同，因此在进行统计量的计算之前，首先需要对原始聚类数据在每一个维度进行归一化处理，如下式所示：

$$x' = \frac{x}{x_{\max}} \tag{1}$$

具体做法是依次遍历每一维的数据点取值，计算出该维度中的每个数据点距离中心点的最大距离，以此为基准归一化该维，令所有数据点该维度的取值范围在–1~1。同理归一化数据点在所有维度的取值。

2) 统计量的计算

根据归一化之后的数据点集合完成平均误差、方差、包含 3/4 数据点的半径及包含 1/4 数据点的半径 4 个统计量的计算。

3) 动态惩罚因子的产生

基于以上 4 个统计量的值，采用 GEP 算法，拟合出惩罚因子与 4 个统计量之间的关系，完成不同数据集惩罚因子的动态构造。

2.2 加入动态惩罚因子的 GEP 聚类算法实现

2.2.1 算法流程

Step 1. 计算动态惩罚因子；

Step 2. 初始化种群；

Step 3. 按照个体所表达的聚类中心进行聚类；

Step 4. 计算个体的适应度；

Step 5. 保留最佳个体到下一代；

Step 6. 依照轮盘赌进行选择操作；

Step 7. 按交叉率对个体进行交叉算子运算；

Step 8. 按变异率对个体进行变异算子运算；

Step 9. 若达到最大迭代次数或适应度达到理想值时则执行 step 10，否则执行 step3；

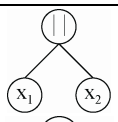
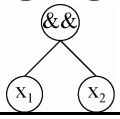
Step 10. 选择出最优个体；

Step 11. 输出聚类划分结果。

2.2.2 符号定义

表 1 所示为本文所实现的 DP-GEPCA 算法的符号定义。

表 1 DP-GEPCA 聚类算法符号定义

聚类运算符	含义	示例	说明
||	其左右子树中节点分属不同的两个簇，实现簇的分割		x_1，x_2 子树分别属于不同的簇
&&	其左右子树中节点属于同一个簇，实现簇的合并		x_1，x_2 子树的第一个中心属于同一个簇，定义 2 个中心的均值代表它们所在簇的中心，其余簇相互独立

2.2.3 编码方式

针对聚类问题特殊性，本文染色体采用整数编码，由单基因构成。每个基因由基因头部和基因尾部组成。其中，基因头部由聚类运算符&&、||和数据点序号组成，而基因尾部只能由数据点序号组成。例如，&&&&||&&|| $x_1x_2x_3x_4x_5x_6$ 是一合法染色体。其中，$x_i\,(i\in[0,n-1]$，n 为数据集中数据总个数，x_i 表示数据集中某一数据点)为当前系统中某个簇的中心。聚类过程初始种群由随机生成的一定数量染色体组成。

2.2.4 加入动态惩罚因子的适应度函数

经过解码染色体信息可以得到各个聚类簇的中心点坐标，然后对数据集中每个数据点，根据其与各个簇中心的欧氏距离，将其归入欧氏距离最小的簇。所有数据点完成操作后重新计算各个簇的中心坐标，并计算各个簇中所有数据点与中心坐标的平方误差总和 E。第 i 个簇的平方误差总和 e_i 定义如下：

$$e_i=\sum_{j=1}^{c_n}(c_j-m_i)^2 \tag{2}$$

式中：m_i 为第 i 簇的中心坐标；c_j 为第 i 簇中的数据点；c_n 为第 i 簇中的数据点的个数。e_i 的值越小，则生成的簇越紧凑和独立。

由于 GEP 一般按照适应度由大到小选择个体，通常在标准 GEP 算法中取如下适应度函数：

$$\text{fitness}=\frac{1}{1+\sum_{i=1}^{k}e_i} \tag{3}$$

式中：k 为簇的个数。

为实现聚类算法中最优聚类个数的自动产生，本文引入动态惩罚因子后的适应度函数修改为：

$$\text{finess}=\frac{1}{1+\sum_{i=1}^{k}c_ie_i} \tag{4}$$

式中：$C_i=(k-i)p$ (5)

p 为惩罚因子，产生过程如 2.1 小节所示。i 的取值是将(2)式产生的每个簇的平方误差总和进行从大到小的排序，得到新的平方误差总和序列 $E'\{e_1',e_2',\cdots,e_k'\}$，其中 $e_1'\geqslant e_2'\geqslant\cdots\geqslant e_k'$，由此完成 GEP 算法中适应度函数的构造。

2.2.5 遗传算子

1) 选择算子：

本文选择策略采用轮盘赌算法，根据适应值进行选择。适应值越好的个体被复制到下一代的可能性越大，在复制的过程中根据赌轮原则的结果确定被复制的次数，同时群体大小保持不变。

2) 变异算子设计：

变异操作可以作用在染色体的任意位置，并且规定如果变异发生在头部，可以变异为任意的函数或终结符；若发生在尾部，则只能变异为终结符，这种变异方法保证了产生的新染色体的结构的有效性。依据一定的变异概率，对染色体进行变异操作。

3) 插串算子设计：

GEP 算法通常采用 3 种插串操作：

i) IS 插串(Insertion Sequence Transposition)：转换片段是从染色体中随机选取的，并且，可以插入到基因头部除了起始位置的任何位置。

第 28 卷第 4 期 系统仿真学报 Vol. 28 No. 4
2016 年 4 月 Journal of System Simulation Apr., 2016

ii) RIS 插串(Root Insertion Sequence Transposition)与 IS 插串类似，但是它的转换片段只能插入到基因的起始位置。

iii) 基因插串(Gene Transposition)选取一个完整的基因，然后插入到染色体的起始位置，而被选中的基因在新染色体中被删除。

本文选择了 IS 插串和 RIS 插串两种插串操作，没有进行基因插串是因为本文的蛋白质是单基因编码的，不能进行基因插串操作。

4) 重组算子设计：

重组算子一般也有 3 种：单点重组、两点重组和基因重组。根据不同的重组算子，在 2 个父染色体中随机的选择重组串或基因，然后互相交换被选中的串或基因。本文在重组算子方面也只选择了单点重组和两点重组 2 种重组算子。

2.2.6 停止条件

当满足以下条件之一时，则停止演化，否则再次进行适应值计算及演化算子操作：

1) 达到最大迭代次数，一般设为 10 000；

2) 适应度达到理想值，种群最优个体的适应值达到预先设定的值如 0.999 999，或在一定演化次数(如 3 次)内没有变化。

3 实验结果与分析

本文共包含了 3 组实验：

实验 1：主要针对惩罚因子研究其对实验结果的影响，实验由构造的 4 个特殊数据集完成；

实验 2：主要完成最佳惩罚因子自动生成，实验数据仍采用与实验 1 相同的 4 个数据集；

实验 3：主要采用标准聚类算法测试数据集 Iris 进行算法的有效性验证，并与已有的 4 种经典聚类算法进行比较，进一步证明了本文算法的有效性。

实验参数设置如表 2 所示，其中终结符集合、函数符集合、基因头部长度、适应值函数 4 个参数分别根据实验 1，实验 2 和实验 3 的特点在各项试验中单独设置：

表 2 DP-GEPCA 算法通用参数设置

参数	值	参数	值
终结符集合	***	IS 插串概率	0.1
函数符集合	***	IS 插串长度	5
基因头部长度	***	RIS 插串概率	0.1
适应值函数	***	RIS 插串长度	5
最大演化次数	10 000	基因插串概率	0.1
种群规模	50	单点重组概率	0.3
选择算法	轮盘赌选择	两点重组概率	0.3
变异概率	0.044	基因重组概率	0.1

3.1 实验 1：惩罚因子对实验结果影响

输入：点集、不同的惩罚因子

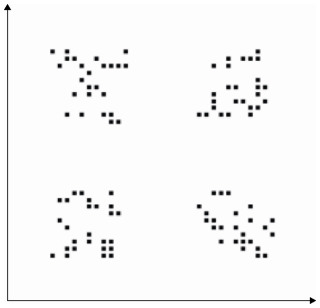
输出：聚类结果

与 K-Means 算法相比，本文 DP-GEPCA 不需要事先指定聚类的个数，由算法中的惩罚因子自动生成。显而易见，聚类的个数受惩罚因子影响。当惩罚因子等于 0 时，DP-GEPCA 聚类的个数接近于基因的尾部长度(即所允许的最大个数)，这是因为在无限制的情况下，DP-GEPCA 自动通过增加聚类的数目降低整体数据的平方误差，以达到适应值函数的最优化；当惩罚因子大于一定值时，DP-GEPCA 聚类的个数接近于 1，这是因为在惩罚因子过大的情况下，DP-GEPCA 自动通过减少聚类的数目降低整体数据的平方误差的惩罚。由此可见，惩罚因子不能过大也不能过小，要设置的恰到好处。而且该惩罚因子与具体的数据集有关。

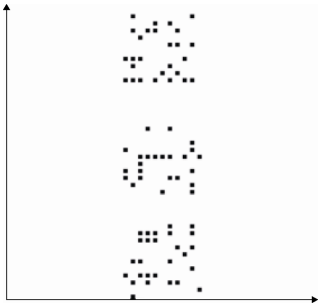
为了实现不同数据集中惩罚因子的自动生成，首先要研究惩罚因子对实验结果的影响。因此，实验 1 构造了 4 组特殊的数据集，研究其不同惩罚因子对 DP-GEPCA 聚类效果的影响。除了如表 2 中给定的 GEP 的相关实验参数外，终结符集合取 $\{0,1,\ldots,(n-1)\}$，函数符集合取 $\{-1(||),-2(\&\&)\}$，基因头部长度取 8，适应值函数用 2.2.4 节的计算方法。

3.1.1 实验数据

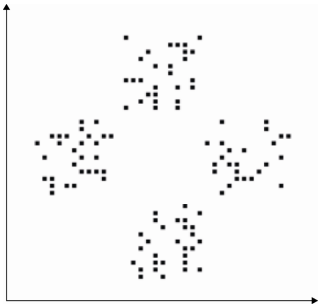
构造 4 组数据集如图 1 所示，惩罚因子分别取 0，0.1，0.2，……，1.0 各进行 100 次实验。



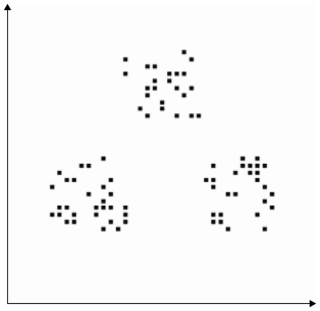
Dataset I



Dataset II



Dataset III



Dataset IV

图 1　4 组构造特殊数据集

3.1.2 实验结果

对上述 4 组数据集的聚类结果进行统计，统计结果如图 2 所示。其中横坐标代表惩罚因子，纵坐标代表最终聚类的个数。

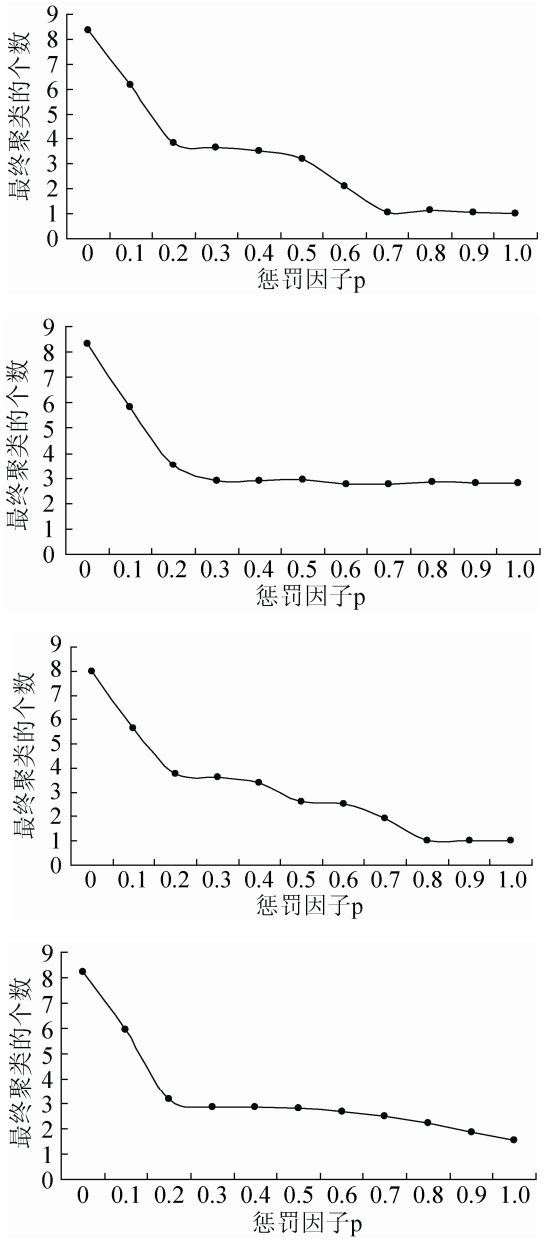


图 2　实验 1 结果图

3.1.3 实验分析

实验结果表明，对于以上 4 个数据集，根据图 1，最好的聚类个数显然分别是 4，3，4，3，因此，最适合这些数据集的惩罚因子区间为[0.2，0.4]，[0.3，1]，[0.2，0.3]，[0.2，0.7]。取区间的中点作为最佳惩罚因子，即 0.3，0.65，0.25，0.45。

3.2 实验 2：最佳惩罚因子自动生成方法

输入：数据集的特征统计量

输出：最佳惩罚因子

第 28 卷第 4 期 系统仿真学报 Vol. 28 No. 4
2016 年 4 月 Journal of System Simulation Apr., 2016

根据实验 1 可知，惩罚因子的取值与整体数据集的分布有关，因此实验 2 选取 2.1 小节中所述的反映数据集中数据点整体分布的 4 个重要的统计量：平均误差、方差、包含 3/4 数据点的半径及包含四分之一数据点的半径，并根据实验 1 的结果完成惩罚因子的自动生成。除了如表 2 中给定的 GEP 的相关实验参数外，终结符集合取 {0,1,…,3}，函数符集合取 {+,–,*,/,*s*,*c*,*q*,*e*,*l*}，基因头部长度取 10，适应值函数用 1/1+函数值。

3.2.1 实验数据

实验 2 继续采用图 1 所示实验数据，其中最佳惩罚因子值分别为：0.3，0.65，0.25，0.45，实验过程如下：

Step 1. 由于聚类关心的是数据集之间的相对位置，因此先将数据集进行归一化，以中心向量为原点，令每个数据在每一维度上的取值归一化到 –1~1 之间。

Step 2. 对归一化后的数据集，计算平均误差、方差、包含 3/4 数据点的半径及包含 1/4 数据点的半径等 4 个特征量，如表 3 的 *x* 列所示，*y* 列所表示的最佳惩罚因子值取自 3.1 节。

Step 3. 将统计量 *x* 作为输入，最佳惩罚因子 *y* 作为输出，采用 GEP 算法，构建最佳惩罚因子的计算表达式。

表 3 图 1 数据集的 4 个特征量

数据集	*x*				*y*
	平均误差	方差	包含 3/4 数据点的半径	包含 1/4 数据点的半径	最佳惩罚因子
Dataset I	14.357 112 49	10.683 041 43	16.977 645 18	12.082 570 86	0.3
Dataset II	8.365 397 461	48.112 919 09	14.399 684 98	0.074 851 705	0.65
Dataset III	12.095 850 35	9.193 179 015	14.993 642 71	9.424 270 811	0.25
Dataset IV	12.226 160 25	9.698 532 44	14.689 924 8	10.053 140 31	0.45

3.2.2 实验方法

由于在实验 1 中，得出的最佳惩罚因子精确到 0.05，因此实验 2 将由以上 4 个统计量构建出的输出值也精确到 0.05。适应值计算方法如下：

Step 1. 对基因进行解码，得到其对应的基因表达式

Step 2. 将 4 组数据的参数分别代入基因表达式，得到 4 个精确到 0.05 的输出值；

Step 3. 将 4 个输出值与对应的最佳惩罚因子作差，对差取平均值后求和得到总误差值；

Step 4. $\text{适应值} = \dfrac{1}{1+\text{总误差值}}$

3.2.3 实验结果

该实验在经过 100 次迭代后，拟合出的最优基因为：ss*ss*c/0302103311312。

其对应的表达式树见图 3 所示。

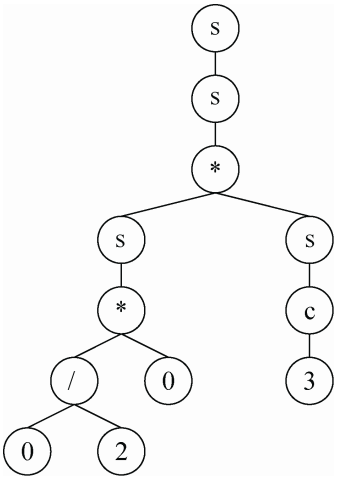


图 3 图 1 中 4 组数据构造的表达式树

其对应的表达式为：

$$\sin(\sin((\sin((x_1 / x_3) * x_1)) * \sin(\cos(x_4))))$$

该实验结果得到的表达式主要为了后续实验生成动态惩罚因子所用，通过该表达式及特定数据集，可以动态得到不同数据集的惩罚因子值。

3.3 实验 3：DP-GEPCA 算法有效性验证

为了进一步验证本文所提出的加入动态惩罚因子的 GEP 自动聚类算法的有效性，实验 3 采用标准聚类算法测试数据集 Iris 进行算法的有效性验证。除了如表 2 中给定的 GEP 的相关实验参数外，终结符集合取 0,1,…,149，函数符集合取 $\{-1(\|),-2(\&\&)\}$，基因头部长度取 8，适应值函数用 2.2.4 节的计算方法。

3.3.1 实验数据

本实验选取 UCI 标准数据集中聚类算法实验库中的 Iris 数据集进行算法的有效性验证。Iris 数据集是在数据挖掘、数据分类中非常常用的测试集、训练集。该数据集以鸢尾花的特征作为数据来源，共包含 150 个数据，分别为 setosa，versicolor，virginica 三类数据，每类 50 个数据，每个数据包含 4 个属性，每个数据包含 4 个独立的属性，即每个数据包含 4 维属性，分别代表植物的萼片、花瓣的长和宽。

3.3.2 实验方法

Step 1. 对初始数据集进行归一化；

Step 2. 计算归一化后的数据集的统计量，$x_1=0.37$，$x_2=0.03$，$x_3=0.54$，$x_4=0.18$；

Step 3. 代入实验 2 拟合出的表达式，得到惩罚因子 $p=0.20$；

Step4. 对初始数据集，基于惩罚因子 p 进行 100 次 DP-GEPCA 实验。

3.3.3 实验结果

通过实验，100 次结果中最好的基因为：–1，–1，39，132，62，5，102，27，128，34，139，39，132，62，5，102，27，其对应的表达式树如图 4 所示，其对应的表达式为：$(132\|62)\|39$。对应的聚类中心为：[6.615 789 473 684 21，2.998 245 614 035 09，5.468 421 052 631 58，1.966 666 666 666 67]，[5.793 023 255 813 95，2.704 651 162 790 7，4.160 465 116 279 07，1.290 697 674 418 6]，[5.006，3.418，1.464，0.244]。

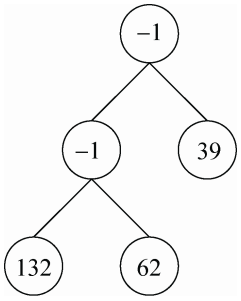


图 4 实验 3 结果表达式树

实验对 K-means 算法、PSO+K-means 算法、K-medians 算法、K-medoids 和本文的 DP-GEPCA 算法在经典数据集 Iris 上进行测试并作对比分析。各种算法的实验结果如表 4 所示。

表 4 Iris 数据集聚类结果性能对比表

	K-means	PSO+K-means	K-medians	K-medoids	DP-GEPCA
最好正确率	88.67	92	89.37	93.3	92.67
最差正确率	74.67	86.66	76	85.3	88.67
平均正确率	83.16	89.4	83.6	89.6	89.96
正确率标准差	26.32	4.5	22.95	8.75	0.74

由表 4 可以看出：K-means 算法的聚类效果依赖于初始点的选择，所以聚类的标准差相对较大，全局寻优能力较差；PSO+K-means 算法与 K-means 算法相比，算法的寻优能力增强，标准误差减小，但是由 PSO 易早熟，所以算法后期收敛速度缓慢，很难达到全局最优解；K-medoids 算法引入了线性调整策略可以较快地定位到最优解，但是全局搜索能力不明显，仍然存在易早熟的问题；本文的 DP-GEPCA 算法由于加入初始化过程、动态惩罚因子，使得全局搜索能力增强，并能够跳出局部最优解，聚类精准度都有提高，标准差最小。

综上所述可以看出，本文提出的加入动态惩罚因子的 GEP 自动聚类算法相比经典的聚类算法及传统的 GEP 聚类算法在效率和精确度上都有提高，克服了算法易陷入局部最优解和后期收敛速度缓慢的缺陷，提高了算法的鲁棒性和准确性。

第 28 卷第 4 期 系统仿真学报 Vol. 28 No. 4
2016 年 4 月 Journal of System Simulation Apr., 2016

4 结论

本文提出了一种加入动态惩罚因子的 GEP 自动聚类算法 DP-GEPCA。该算法首先采用 GEP 算法动态计算出针对不同数据集分布特点的动态惩罚因子，然后通过惩罚因子完成聚类个数与适应值函数之间的平衡控制，最终在得到好的聚类结果的同时，保证聚类个数的最优化，并能借助惩罚因子的值发现样本集中的离群点；同时，通过实验也进一步证明了本文算法较一些传统聚类算法在准确性上具有一定的有效性及先进性。

参考文献:

[1] Han Jiawei, Micheline Kamber. Data Mining: Concepts and Techniques [M]. San Francisco, USA: US Morgan Kaufmann Publishers, 2001.

[2] Abdelhamid N, Ayesh A, Thabtah F. Phishing Detection based Associative Classification Data Mining [J]. Expert Systems with Applications (S0957-4174), 2014, 41(10): 5948-5959.

[3] 谢东, 吴敏. 基于范围语义的非一致性数据库聚集查询 [J]. 中南大学学报(自然科学版), 2008, 39(4): 810-815.

[4] Ding Xiaojian, Chang Baofang. Active set strategy of optimized extreme learning machine [J]. Chinese Science Bulletin (S1001-6538), 2014, 59(10): 4152-4160.

[5] Zhao Xingming, Ngom Alioune, Hao Jinkao. Pattern recognition in bioinformatics [J]. Neurocomputing (S0302-9743), 2014, 10(1): 1-2.

[6] Feng Junping, Hang Weixing, Zhao Liang, et al. Endurance Mileage Estimation of the State of Charge for Electric Vehiclesm Based on the Fuzzy Inference of Artificial Neural Network [J]. International Journal of Advancements in Computing Technology (IJACT) (S2005-8039), 2013, 5(1): 540-548.

[7] Rahman Husna-Abdul, Harun Sulaiman-Wadi, Arof Hamzah, et al. Classification of Reflected Signals from Cavitated Tooth Surfaces using an Artificial Intelligence Technique Incorporating a Fiber Optic Displacement Sensor [J]. Journal of Biomedical Optics (S1083-3668), 2014, 19(5): 929-938.

[8] 赵小强, 李雄伟. 基于改进马氏距离的模糊 C 聚类研究 [J]. 中南大学学报(自然科学版), 2013, 44(增 2): 195-198.

[9] 曹书锦, 朱自强, 鲁光银. 基于自适应模糊聚类分析的重力张量欧拉反褶积解 [J]. 中南大学学报(自然科学版), 2012, 43(3): 1033-1038.

[10] Han J W, Kambr M. Datamining concepts and techniques [M]. Beijing, China: Higher Education Press, 2001.

[11] Holland J H. Adaptation in natural and artificial systems: an introductory analysis with application to biology, control and artificial intelligences [M]. Ann Arbor, MI, USA: The University of Michigan Press, 1975.

[12] 金敏, 鲁华祥. 一种遗传算法与粒子群优化的多子群分层混合算法 [J]. 控制理论与应用, 2013, 30(10): 1231-1238.

[13] Murthy C A, Chowdhury N. In search of optimal clusters using genetic algorithms [J]. Pattern Recognition Letters: (S0167-8655), 1996, 17(8): 825-832.

[14] Bandyopadhyay S, Maulik U. An evolutionary technique based on K-means algorithm for optimal clustering in RN [J]. Information Sciences: An International Journal: (S0020-0255), 2002, 146(1): 221-237.

[15] 莫海芳, 康立山. 用GEP实现复杂函数的自动建模 [J]. 系统仿真学报, 2008, 20(11): 2828-2831.

[16] 李康顺, 黄浩华, 张文生. 一种基于 GEP 的多层物流网络 Prüfer 编码优化算法 [J]. 系统仿真学报, 2012, 24(3): 594-602

(上接第 805 页)

[16] 石一民. 离心泵性能曲线多项式拟合的一种简单方法 [J]. 上海煤气, 2005 (4): 39-41.

[17] 许景辉, 何东健, 张成风. 基于ANN的离心式水泵特性曲线拟合方法研究[J]. 水力发电，2005, 31(6):38-40.

[18] T Ahonen, J Tamminen, J Ahola. Frequency-converter-based hybrid estimation method for the centrifugal pump operational state [J]. IEEE Trans. Industrial Electron. (S0278-0046), 2012, 59(12): 4803-4809.

[19] O I Kiselichnik, M Bodson. Nonsensor control of centrifugal water pump with asychronous electirc-drive motor based on extended Kalman filter [J]. Russian Electrical Engineering (S1068-3712), 2011, 82(2): 69-75.

[20] D J Kernan, E P Sabini, N W Ganzon, et al. Method for determining pump flow without the use of traditional sensors [P]. U. S. Patent 7945411, 2011.