小生境排挤聚类算法*)

业 宁[1,2] 董逸生[1]

(东南大学计算机科学与工程系 南京210096)[1] (南京林业大学信息学院计算机系 南京210037)[2]

Clustering Algorithm Based on Crowding Niche

YE Ning[1,2] DONG Yi-Sheng[1]

(Department of Computer Science and Engineering, Southeast University, Nanjing 210096)[1]

(Department of Computer ,Nanjing Forestry University, Nanjing 210037)[2]

Abstract A new clustering algorithm is proposed in this paper, which is based on crowding niche. Homogeneity spontaneous to withstands heterogeneity when organisms are evolving. Contemporary, Individual in same class compete each other to strive for limited resource. Individual that has bad fitness will be eliminated. We propose a clustering algorithm based on this idea. Experiment evaluation has proved its efficiency.

Keywords Clustering algorithm,Crowding ,Niche,Genetic algorithm

1 引言

在生物学中,小生境(Niche)是指特定环境下的一种生存环境。在生物进化过程中,相同的物种一般生活在一起,共同繁衍后代,它们往往生活在特定的区域,如热带动物很难在北极生存,而北极的动物很难在赤道存活。受大自然的物竞天择,优胜劣汰的思想启发,De Jong 提出了基于排挤机制的(Crowding)的小生境方法[1],排挤的思想源于在一个有限的空间中,各种不同的生物为了能够延续生存,它们之间必须相互竞争有限的资源。该方法已经成功地用于解决多峰函数的极限问题。

近年来,人们通过遗传算法解决聚类问题取得了很多成果[2~5],但是基于小生境排挤机制的聚类方法并未见报道。本文提出了一种小生境排挤聚类算法,将待聚类的数据随机分配到 k 个小生境中,适应度差的个体受到排挤,被迫离开原来的小生境加入到最适合自己的小生境,经过若干代的进化,各个个体找到自己最适合的小生境。

2 基于排挤的小生境聚类模型

小生境聚类模型定义如下:

CrowdingNicheGa={C,E,P_0,N,Λ,Φ,Γ,Ψ,T,Θ,Ξ,K}

其中,C——编码方法,E——适应度评价函数,P_0——初始群体,N——群体大小,Λ——排挤算子,Φ——选择算子,Γ——交叉算子,Ψ——变异算子,Θ——合并算子,Ξ——分裂算子,T——遗传运算终止条件,K——聚类期望达到的数目。

(1)评价函数 E 设数据集合 $X=\{x_1,x_2,\cdots,x_n;j=1,\cdots,N\}$,$x_n\in R^D$,假定数据可分为 K 类,则指标矩阵 $A_{kj}=\begin{cases}1 & \text{代表数据 } x_j \text{ 属于 } k \text{ 类}\\ 0 & \text{代表数据 } x_j \text{ 不属于 } k \text{ 类}\end{cases}$。每类的均值为 $m_k=\frac{1}{N_k}\sum_{i=1}^{N}A_{ki}x_i$,式中 $N_k=\sum_{i=1}^{N}A_{kj}$表示第 k 类数据的数量。对数据聚类时,我们希望不同类别数据之间的距离最大,而每一类数据内部之间的距离最小。因此定义评价函数 $E=\frac{D_{in}}{D_{out}}$,式中类内距离 $D_{in}=\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N}A_{ki}(x_i-m_k)(x_i-m_k)^T$,类间距离 $D_{out}=\frac{1}{2K}\sum_{k=1}^{K}\sum_{i=1}^{K}(m_i-m_k)(m_i-m_k)^T$。寻找 min(E)的过程就是数据聚类的过程。

(2)编码策略 C 将指标矩阵 A 作为基因编码,并用随机数0和1初始化矩阵,A 中的数据反映了数据的分布情况。

$$A=\begin{bmatrix}1&0&\cdots&0&0&1\\0&0&\cdots&1&0&0\\0&1&\cdots&0&1&0\end{bmatrix}_{K\times N};$$

$$A[i,j]=\begin{cases}1 & \text{代表个体 } j \text{ 在 } i \text{ 类中}\\ 0 & \text{代表个体 } j \text{ 不在 } i \text{ 类中}\end{cases}\quad \text{且}\sum_{k=1}^{K}A_{ki}=1。$$

(3)交叉算子 Γ 使用指标矩阵 A,可以设计两种类型的交叉算子,一种是利用自然界中的无性繁殖规律,进行单亲繁殖。如图1所示:通过交换部分基因可以交换个体到不同的类。另外一种是有性双亲繁殖交叉算子,如图2所示:双亲通过交换部分基因而生成新的后代。

$$\begin{bmatrix}1&0&1&0&0&1\\0&0&0&1&0&0\\0&1&0&0&1&0\end{bmatrix}\Rightarrow\begin{bmatrix}1&0&0&0&1&0\\0&0&0&1&0&0\\0&1&1&0&0&1\end{bmatrix}$$

图1 无性繁殖交叉算子

$$\begin{bmatrix}1&0&0&1&0\\0&0&1&0&0\\0&1&0&0&0\\0&0&0&0&1\end{bmatrix}\quad\begin{bmatrix}0&1&0&0&0\\1&0&0&1&0\\0&0&0&0&1\\0&0&1&0&0\end{bmatrix}$$

Parent1 Parent2

$$\Downarrow$$

$$\begin{bmatrix}1&0&0&0&0\\0&0&1&1&0\\0&1&0&0&1\\0&0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&1&0&1&0\\1&0&0&0&0\\0&0&0&0&0\\0&0&1&0&1\end{bmatrix}$$

offspring1 Offspring2

图2 有性双亲繁殖交叉算子

*)江苏省九五重点攻关课题(BJ98017-1)、江苏省十五高科技项目(BJ2001013)和校科研基金重点课题(X02-070-1(Z))资助。业 宁 博士研究生,主要研究方向为数据库技术,人工智能。董逸生 教授,博士生导师,主要研究方向为信息系统、数据库、软件技术。

(4)选择算子 Φ　在群体 P 从 t 代到 t+1代的进化过程中，一些优良的聚类方案将进化到下一代。选择策略一般采用轮盘赌模型。

(5)变异算子 Ψ　变异的存在可以维持聚类方案的多样性。随机地选择 A 中的一点，将该点的值 v 和列方向相邻的一个非 v 值互换。

$$\begin{bmatrix}1&0&0&0&0\\0&0&1&1&0\\0&1&0&0&1\\0&0&0&0&0\end{bmatrix}\Rightarrow\begin{bmatrix}1&0&0&0&0\\0&1&1&1&0\\0&0&0&0&1\\0&0&0&0&0\end{bmatrix}$$

图3　变异算子

(6)排挤算子 E　$D_{ki}=A_{ki}(x_i-m_k)(x_i-m_k)^T$ 表示每一个数据到其类中心的距离，排挤算子 $\Lambda_{ki}=D_{ki}/\sum_{i=1}^{N}D_{ki}$，$\Lambda_{ki}$的范围在[0,1]之间，其值越大，所对应的数据受到排挤的可能性就越大。

$$\begin{bmatrix}0&0&0&1&0\\0&0&1&0&0\\0&1&0&0&0\\1&0&0&0&1\end{bmatrix}\Rightarrow\begin{bmatrix}0&0&0&1&1\\0&0&1&0&0\\0&1&0&0&0\\1&0&0&0&0\end{bmatrix}$$

图4　排挤算子

(7)合并与分裂算子 Θ 和 Ξ　$DC_{ki}=(m_i-m_k)(m_i-m_k)^T$ 是类间距离矩阵，合并算子 $\Theta_k(DC_{ki})\subset DC_{ki}$，其计算结果是类间距离矩阵中具有较小值的子集，该子集中的数越小，其代表的两类合并概率越大。分裂算子 $\Xi_k(D_{ki})\subset D_{ki}$，其计算结果是类内距离矩阵中具有较大值的子集，该子集中的数越大，受到的排挤也越大，分裂概率就越高。

3　算法

输入：数据集 DataSet

输出：记录聚类信息的矩阵 A

步骤：

1. 给出大约的聚类数目 K($2\leqslant K\leqslant N$)

2. 初始化矩阵 $A_{K\times N}$为0，随机将1置入到矩阵中，使得每一列只有一个数为1。

3. 初始化交叉算子的概率为 $P_\Gamma=0.5$，变异算子的为 $P_\Psi=0.1$，排挤概率为 $P_\Lambda=0.8$，本例由于采用了无性单亲繁殖，没有使用选择算子。

4. while 进化代数<最大进化代数 T OR 非(连续多次没有合并和分裂操作)

{

1)计算并标识出个体到其类中心的距离 D_{ki}，类间距离矩阵 D_{out}，以及适应度 E

2)小生境之间进行排挤运算，排挤出每个小生境中的一些适应度差的个体

3)小生境之间进行交叉、变异运算，防止陷入局部最优

4)用合并算子 Θ_k 进行合并运算，如果为真，则 K=K−1，自动减少一类

5)用分裂算子 Ξ_k 进行分列运算，如果为真，则将该行的数据分裂为两行，K=K+1

}

4　实验结果及分析

实验1：采用人工生成数据，分别以坐标(0,0)、(35,35)、(40,0)为中心，以 10、10、2为半径，按正态分布生成100、100、4个数据，共204个数据。

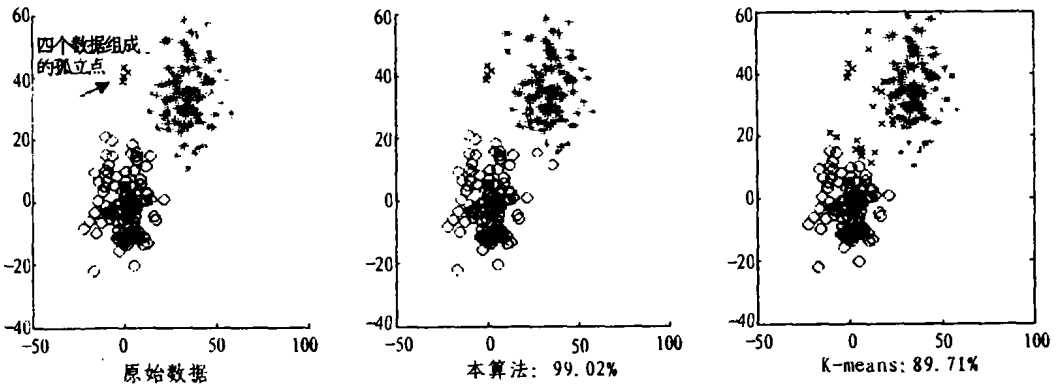

图5　本算法与 K-means 算法精度比较

由图5可见，4个数据组成的孤立点对 K-means 聚类产生很大的影响，另外两类中分别有9个和12个数据被错分到孤立点类。而使用本算法，孤立点的4个数据被正确地分为一类，另外两类中有两个数据被错分，聚类精度为99.02%。

实验2：选用 UCI 机器学习数据 Iris benchmark[6]，共3类150个数据，每类数据都是50个。使用本算法的聚类过程如图6所示。

图6是小生境聚类算法的计算结果，第1代时数据被随机分配到 K 类中。进化到第10代，聚类的正确率达到96%。进化到第30代，聚类的正确率达到98.67%。进化到50代时，聚类的正确率仍然是98.67%，没有发生变化。

K-means 算法对本数据集聚类，第二和第三类分别有2个和4个数据被错分，正确率为96%。

图7显示了小生境聚类算法的收敛情况，当进化到第15代时，以后的曲线几乎没有变化了，说明该算法收敛很快。

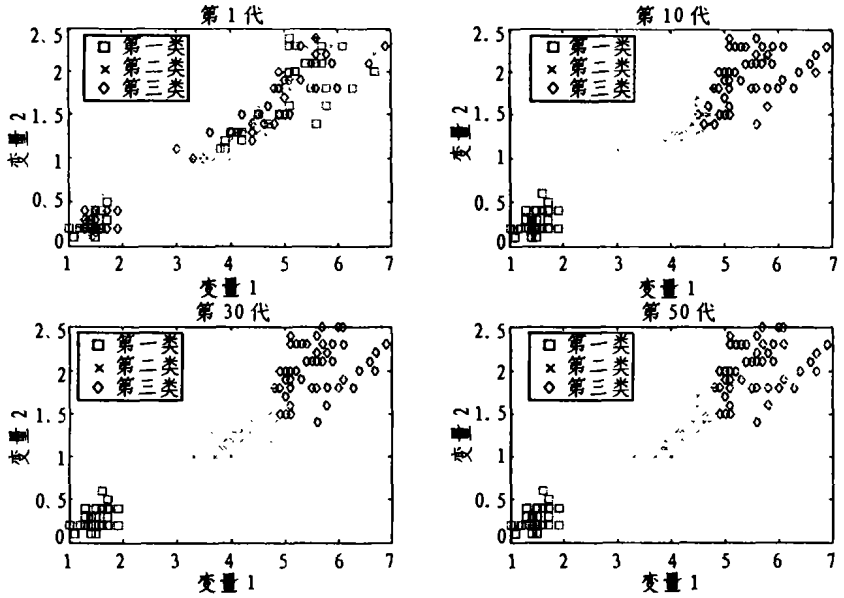

图6 小生境聚类算法

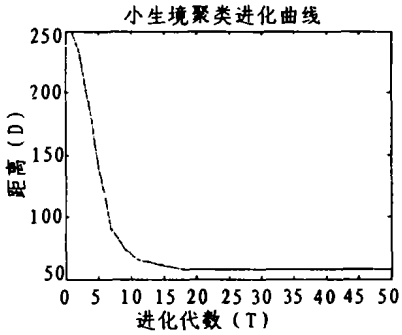

图7 小生境聚类算法的收敛速度

5 算法分析

5.1 时间复杂度

在CPU为PⅢ866,RAM为128兆的PC机上测试,测试数据如下:

数据集	本文算法	K-means
人工数据	1.412秒	0.1100秒
Iris	0.8210秒	0.0500秒

本算法的时间复杂度约为$O(TN^2/K^2)$,T为进化的代数,N为数据的个数,K为类的个数。

K-means聚类算法的时间复杂度为O(TNK),T为迭代次数,N为数据的个数,K为类的个数。一般情况下,$K \ll N$,$T \ll N$。

5.2 小生境聚类算法的优势

对于数据数量相差悬殊的类别,用K-means算法有很大的偏差,而用本算法通过排挤算子将孤立点排挤到单独的一类,可以检测出孤立点(小数量)数据,因此具有较高的聚类精度。K-means算法需要先验地确定一个类别数,并且如果初始点选择不恰当,会聚类出错误的结果。本算法可以在准确类别数未知的情况下自适应聚类。在聚类前先给出一个大约的类别数,再在聚类的过程中自动地两类合并为一类或一类分裂为两类,最后收敛到一个最佳类数。本算法是一种特殊的遗传算法,由于交叉和变异算子的作用,可以得到全局最优近似解。

参考文献

1 De Jong K A. An Analysis of the Behavior of Class of Genetic Adaptive Systems:[Ph. D Dissertation]. University of Michigan, No. 76-9381,1975

2 Kuncheva L I,Bezdek J C. Nearest prototype classification: clustering, genetic algorithms, or random search. IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews,1998,28(1):160～164

3 Tseng L Y, Yang S B. A genetic clustering algorithm for data with nonspherical -shape clusters. Pattern Recognition,2000,33:1251～1259

4 于歆杰,王赞基.一种新的聚类方法及其在多峰优化中的应用.清华大学学报,2001,41(4-5):159～162

5 王磊,戚飞虎.大矢量空间聚类的遗传k-均值算法.上海交通大学学报,1999,33(9):1154～1156

6 Blake C, Keogh E, Merz C J. UCI repository of machine learning databases. http: // www.ics.uci.edu/～ mlearn/MLRepository.html, Irving, CA: University of California, Department of Information and Computer Science, 1998