免疫接种粒子群的聚类算法

郑晓鸣，吕士颖，王晓东

(福州大学数学与计算机学院 福州 350002)

【摘要】将粒子群优化算法和K均值算法结合进行聚类分析，同时引入了免疫系统中的免疫接种和免疫选择机制来指导粒子的迭代过程，提出了一种基于免疫接种粒子群的聚类算法，在粒子群迭代的过程中加入免疫接种机制指导粒子的飞行方向，再通过免疫选择机制对接种的结果进行选择，确保粒子种群向更优的方向移动。实验结果证明，基于免疫接种粒子群的聚类算法基本克服了K均值算法容易受初始聚类中心影响的缺点，聚类结果稳定，而且比基于粒子群优化的聚类算法取得了更好的聚类效果。

关 键 词 聚类； 免疫选择； 免疫接种； K均值； 粒子群优化
中图分类号 TP311.13 文献标识码 A

Clustering with Immunity-Vaccination Based on Particle Swarm Optimization Algorithm

ZHENG Xiao-ming，LÜ Shi-ying，WANG Xiao-dong

(College of Mathematics and Computer Science, Fuzhou University Fuzhou 350002)

Abstract This paper proposes a clustering algorithm based on Particle Swarm Optimization Algorithm with Immunity-Vaccination (IV-PSO-KMEANS). It combines Particle Swarm Optimization (PSO) algorithm and *K*-means for clustering. Synchronously, Immunity-vaccination and immunity-selection mechanisms of immune system are introduced into the iterative procedure. mmunity-vaccination is used to direct the procedure of particle swarm and immunity-selection is applied to select from the results of vaccination. In result, the swarm is made to move towards a better direction. The experiments show that the IV-PSO-KMEANS algorithm overcomes the problem of *K*-means algorithm that the results are related to the initial clustering centers, and the results of clustering are steadier and better than algorithms based on PSO.

Key words clustering; immunity-selection; immunity-vaccination; KMEANS; PSO

聚类是一种无指导的分类方法，该方法在没有预先定义分类的情况下，将一个大的数据集分成若干个类，要求同一类中的数据尽可能相似，而不同类之间的数据尽可能不同。聚类作为数据挖掘中的一种重要方法，越来越为人们重视。在已有的聚类算法中，K均值算法因其算法简单快速，被广泛地应用于数据挖掘和知识发现领域中。但是K均值算法存在着两个固有缺点：(1) 聚类结果受初始聚类中心影响，不同的初始聚类中心可能产生不同的聚类结果；(2) 容易陷入局部极值，得到的聚类结果可能不是全局最优。

文献[1]模拟鸟群觅食的过程首次提出了粒子群优化(Particle Swarm Optimization，PSO)算法。该算法模拟鸟群飞行觅食的行为，通过鸟群之间的集体协作而使群体达到最优。PSO是一种较好的优化算法，它对优化目标函数的形式没有特殊要求，而且算法简单，具有全局寻优能力，已经在函数优化、模糊控制系统优化等许多应用领域中得到了广泛的研究和应用。但是PSO算法存在着收敛速度慢，在多峰值函数测试中容易过早收敛的缺点。

文献[2-3]把粒子群优化算法用于解决K均值算法存在的问题，提出基于粒子群优化的聚类算法(PSO-KMEANS)，取得了较好的结果。

本文提出一种基于免疫接种的粒子群聚类算法(IV-PSO-KMEANS)，在粒子群迭代过程中加入免疫算子(免疫接种)指导粒子的飞行方向，同时通过免疫选择机制对接种的结果进行选择，使粒子向更优的方向飞行，整个种群能更快更准确地向全局最优的方向飞行。与基于粒子群优化的聚类算法以及传统的聚类算法(KMEANS)进行对比实验，结果表明，

收稿日期：2007－09－09
作者简介：郑晓鸣(1983－)，女，硕士生，主要从事WEB数据挖掘方面的研究；吕士颖(1983－)，男，硕士生，主要从事数据挖掘方面的研究；王晓东(1957－)，男，教授，主要从事数据结构、算法设计与分析方面的研究.

IV-PSO-KMEANS受初始聚类中心的影响很小，聚类结果比较稳定，比基于粒子群优化的聚类算法(PSO-KMEANS)具有更好的聚类结果。

1 K均值算法

1.1 K均值聚类问题的数学描述[5-7]

*K*均值算法的数学描述为：假设给定一个包含 n 个数据的集合 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$，其中元素 $\boldsymbol{x}_i\in\boldsymbol{X}(i=1,2,\cdots,n)$ 为 d 维向量，要求通过聚类分析把该数据集划分为 k 类，也就是求数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$ 的一个划分 $C=\{C_1,C_2,\cdots,C_k\}$ 使得：

$$\boldsymbol{X}=\bigcup_{i=1}^{k}C_i \tag{1}$$

$$C_i\neq\phi \qquad i=1,2,\cdots,k \tag{2}$$

$$C_i\cap C_j=\phi \qquad i,j=1,2,\cdots,k;\ i\neq j \tag{3}$$

在聚类分析过程中有一个衡量聚类结果好坏的聚类准则函数：

$$J=\sum_{j=1}^{k}\sum_{\forall\boldsymbol{x}_l\in C_j}D(\boldsymbol{x}_l,m_j) \tag{4}$$

式中　$D(\boldsymbol{x}_l,m_j)$ 为聚类结果中数据 $\boldsymbol{x}_l$ 与所属的类 C_j 的类中心 m_j 之间的距离；$\sum_{\forall\boldsymbol{x}_l\in C_j}D(\boldsymbol{x}_l,m_j)$ 为所有属于 C_j 的数据与类中心 m_j 的距离之和；J为所有类内距离之和，J值越小，表示聚类效果越好。

K均值算法采用欧氏距离公式：

$$D(\boldsymbol{x}_l,m_j)=\sum_{a=1}^{d}\sqrt{\left|\boldsymbol{x}_{la}^2-m_{ja}^2\right|} \tag{5}$$

式中　$\boldsymbol{x}_{la}$ 为数据 $\boldsymbol{x}_l$ 第 a 维上的值；m_{ja} 为聚类中心 m_j 第 a 维上的值。

K均值算法实际上是寻找数据集 $\boldsymbol{X}$ 的一个划分 C，使得聚类准则函数 J 最小的问题。所以本文使用粒子群优化算法完成寻找使得 J 最小的聚类过程。

1.2 K均值算法描述[4,6]

设有输入为簇的数目 k 和包含 n 个对象的数据集；输出为 k 个簇，使 J 最小。

(1) 任意选择 k 个对象作为初始的簇中心；

(2) 根据各簇中心，将数据集中，并将各对象赋给最类似的簇；

(3) 根据聚类结果，用 $(\sum_{\boldsymbol{X}\in C_j}\boldsymbol{X})\Big/|C_j|$ 更新计算各簇中心；

(4) 计算准则函数 J；

(5) 重复(2)～(4)，直到 J 不再明显地发生变化。

2 免疫接种的粒子群聚类算法

2.1 粒子群优化算法

假设在一个目标搜索空间中有*N*个粒子组成一个群体，其中第*i*(*i*=1,2,…,*N*)个粒子在*s*维搜索空间中的位置 $\boldsymbol{Z}_i$ (*i*=1,2,…,*N*)表示为一个*s*维的向量，每个粒子的位置都是一个潜在的解；第*i*个粒子的“飞行”速度 $\boldsymbol{V}_i$ (*i*= 1,2,…,*N*)也是一个*s*维的向量。

在粒子群优化算法中，适应度函数是衡量粒子位置好坏的标准。在迭代寻优的过程中，算法会记录下迄今为止每个粒子的最好的位置 P_i (*i*=1,2,…,*N*)和整个粒子群的最好位置(目前为止的全局最优解) P_g。每个粒子下一次“飞行”速度 $\boldsymbol{V}_i^{t+1}$ 和下一步“飞行”位置 $\boldsymbol{Z}_i^{t+1}$ 的计算方法如下：

$$\boldsymbol{V}_i^{t+1}=\omega\boldsymbol{V}_i^{t}+c_1r_1(P_i^t-\boldsymbol{Z}_i^t)+c_2r_2(P_g-\boldsymbol{Z}_i^t) \tag{6}$$

$$\boldsymbol{Z}_i^{t+1}=V_i^{t+1}+\boldsymbol{Z}_i^t \tag{7}$$

式中　*i*=1, 2,…,*N*，*N*为粒子个数；*t*=1,2,…，*t*为迭代次数；ω 为惯性权重；c_1 和 c_2 为两个正常数，称为认知(cognitive)和社会(social)参数；r_1 和 r_2 为[0,1]之间的随机数。迭代的终止条件一般为最大的迭代次数或者全局最优解的适应度值满足预定阈值。

2.2 免疫接种的粒子群聚类算法

2.2.1 粒子编码

本文粒子位置采用的是基于聚类中心的实数编码方式，也就是粒子编码是由聚类产生的一组聚类中心组合成的。由于聚类分析中数据维数为 d，要求把数据分成 k 类，每个粒子是由 k 个聚类中心组成的向量，每个聚类中心是*d*维的向量，所以每个粒子的位置是 $k\times d$ 维向量，具体结构如图1所示。由于粒子的速度和粒子的位置具有相同的数据结构，所以粒子的速度也是 $k\times d$ 维变量。其中*i*=1,2,…,*N*，*N*为粒子个数。

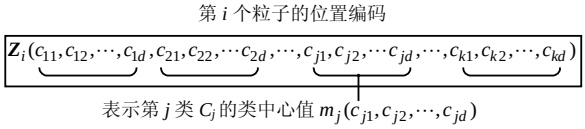

图1　粒子位置编码

2.2.2 适应度函数

在粒子群优化算法中，适应度函数用于判断种群进化过程中粒子所在位置的好坏。聚类结果的好坏，取决于式(4)的 J 的结果，J 越小聚类结果越好。所以本文定义粒子适应度函数为：

$$\text{fitness}(\boldsymbol{Z}_i)=\frac{C}{J+J_0}=1/(\sum_{j=1}^{k}\sum_{\forall\boldsymbol{x}_i\in C_j}D(\boldsymbol{x}_i,m_j)+J_0) \tag{8}$$

式中 c是一个正常数；J_0 是一个很小的正数，表示为了避免出现 J 趋向于0而零溢出的情况；J 表示聚类结果的所有类内距离和，其中的符号表示意义与前文的式(4)相同。算法通过粒子群的迭代来搜索适应度函数 fitness($\boldsymbol{Z}_i$) 的最大值。

2.2.3 免疫接种的粒子群聚类算法

传统的粒子群算法可以使用实数编码，需要用户确定的参数不多，操作简单，但缺点是搜索精度不高，算法后期收敛速度慢。文献[8]用人工免疫系统中的免疫调节和免疫接种机制对粒子群优化算法进行改进，本文借鉴其中使用的免疫接种的机制，在粒子群优化和聚类算法相结合的粒子群聚类的迭代过程中引入一个新的算子(免疫算子)，把求解过程中的当前全局最优解作为免疫疫苗，在下一次迭代过程中对粒子群接种，然后再引入免疫选择机制，从产生的粒子及其父代中选择更有利于种群向最优方向“飞行”的粒子，并增加种群的多样性，使得种群能更快更准确地向全局最优的方向飞行，从而提高聚类算法的聚类效果，抑制算法迭代过程中可能出现的退化现象。

曾经出现过关于粒子群优化算法在聚类分析中应用的文章[2-4]，粒子群的初始化过程都是将每个数据随机分配到某一类，作为最初的聚类划分，并计算各类的中心，作为粒子的初始位置编码。但每个数据都以相同的概率被分配到任意类中，容易造成所有类的聚类中心差不多都分布在所有数据的中心位置的附近，因此根据式(8)计算出来的粒子适应度值也都差不多，容易造成粒子群在刚开始就收敛于局部极值。

为了避免初始粒子群的聚类中心集中分布，本文在粒子群被初始读入数据集的过程中，记录下数据集每一维上的最大值 $\boldsymbol{x}_{\max_i}$ 和最小值 $\boldsymbol{x}_{\min_i}$。粒子 i 的位置 $\boldsymbol{Z}_i(c_{11},c_{12},\cdots,c_{1d},c_{21},c_{22},\cdots,c_{2d},\cdots,c_{k1},c_{k2},\cdots,c_{kd})$ 中的 $c_{jl}\in[\boldsymbol{x}_{\max_l},\boldsymbol{x}_{\min_l}]$，因此在 $\boldsymbol{x}_{\max_l}\sim\boldsymbol{x}_{\min_l}$ 中随机取一个数作为 c_{jl} 的取值，其中 $j=1,2,\cdots,k;\ l=1,2,\cdots,d$。本文提出的所有算法都采用这样的方法形成初始粒子群 $A_0(\boldsymbol{Z}_1,\boldsymbol{Z}_2,\cdots,\boldsymbol{Z}_N)$。该方法使初始的聚类中心的随机性比较大，生成的初始粒子群 A_0 更加具有多样性。

基于免疫接种的粒子群聚类算法描述如下。

步骤1 读入数据集，记录下每一维上数据的最大值 $\boldsymbol{x}_{\max_i}$ 和最小值 $\boldsymbol{x}_{\min_i}$，其中 $i=1,2,\cdots,d$。

步骤2 确定数据类别数 k 和粒子个数 N，设定常数 c_1 和 c_2、惯性因子 ω、最大进化代数 $t_{\max}$。设置当前迭代次数为 $t=0$，按前文提出的方法产生 N 个粒子 $\boldsymbol{Z}_1,\boldsymbol{Z}_2,\cdots,\boldsymbol{Z}_N$，组成初始种群 A_0。

步骤3 计算种群 A_t 中每个粒子的适应度值 fitness($\boldsymbol{Z}_i$)，其中 $i=1,2,\cdots,N$。更新 $pbest_i$、$gbest$，并把 $gbest$ 作为免疫疫苗。判断 $t<t_{\max}$ 或 $gbest$ 达到预定最优值是否成立，如果不成立，继续往下计算，否则，结束计算，输出全局最优粒子 $gbest$ 以及聚类结果(每个数据所属的类号)。

步骤4 用式(6)和(7)更新粒子的位置和速度形成粒子群 Q_t。由于聚类结果中类中心 l 维的值都不会超出 $[\boldsymbol{x}_{\max_l},\boldsymbol{x}_{\min_l}]$，所以对 Q_t 中粒子 $\boldsymbol{Z}_i(c_{11},c_{12},\cdots,c_{1d},c_{21},c_{22},\cdots,c_{2d},\cdots,c_{k1},c_{k2},\cdots,c_{kd})$ 中的 c_{jl} 进行如下处理：

$$c_{jl}=\begin{cases}c_{jl} & x_{\min_l}<c_{jl}<x_{\max_l}\\ x_{\max_l} & c_{jl}>x_{\max_l}\\ x_{\min_l} & c_{jl}<x_{\min_l}\end{cases}$$

得到粒子群 R_t。

步骤5 根据粒子群 R_t 中粒子的位置编码对数据集进行k均值化处理：(1) 根据粒子的聚类中心编码，按照最近邻法则，确定对应粒子的聚类划分。(2) 按照聚类划分，计算新的聚类中心，取代原来粒子的编码值，得到粒子群 S_t。

步骤6 记录下粒子群 S_t，计算 S_t 中每个粒子的适应度值 fitness($\boldsymbol{Z}_i$)。然后从种群 S_t 中按免疫接种概率 β 随机抽取一定数量的粒子进行如下接种操作：(1) 设置一个 $k\times d$ 维变量 $\boldsymbol{B}_i(b_{11},b_{12},\cdots,b_{1d},b_{21},b_{22},\cdots,b_{2d},\cdots,b_{k1},b_{k2},\cdots,b_{kd})$ 和 $\boldsymbol{Z}_i(c_{11},c_{12},\cdots,c_{1d},c_{21},c_{22},\cdots,c_{2d},\cdots,c_{k1},c_{k2},\cdots,c_{kd})$ 相对应。(2) 对 $\boldsymbol{B}_i(b_{11},b_{12},\cdots,b_{1d},b_{21},b_{22},\cdots,b_{2d},\cdots,b_{k1},b_{k2},\cdots,b_{kd})$ 中的 b_{jl} 随机地取0或1，然后根据 B_i 对 $\boldsymbol{Z}_i$ 中的 c_{jl} 进行如下处理：

$$c_{jl}=\begin{cases}gbest_{jl} & b_{jl}=1\\ c_{jl} & b_{jl}=0\end{cases}\qquad j=1,2,\cdots,k;\ \ l=1,2,\cdots,d$$

步骤7 对发生变化的粒子进行免疫选择操作，然后计算新产生粒子的适应度值，若适应度值不如原有的值，则取消接种，保持原有粒子。由此形成新一代粒子群 A_{t+1}。

步骤8 $t=t+1$，转步骤3。

3 实验结果

实验开发环境为CPU：P4 2.0 GHz；内存：DDR512 MHz；Windows XP操作系统：Visual C++6.0。

本文实验分别使用基于免疫接种的粒子群聚类算法(IV-PSO KMEANS)、基于优化的聚类算法(PSO KMEANS)和K均值聚类算法(KMEANS)对数据iris.dat进行聚类性能比较。实验数据为国际公用的测试数据UCI中的一组iris.dat。

实验中粒子群算法的聚类采用的参数如下。

(1) 式(8)中 $c=100$； $J_0=0.01$。

(2) 式(6)中惯性因子[9]为：

$$\omega=\omega_{\max}-\frac{\omega_{\max}-\omega_{\min}}{\omega_{\max}\_\text{generation}}\text{recent}\_\text{generation}$$

式中 $\omega_{\max}=1.4$；$\omega_{\min}=0.9$；ω随着当前迭代次数的增加而不断变小；学习因子取 $c_1=c_2=2$。

表1为IV-PSO-KMEANS算法、PSO-KMEANS算法重复进行50次实验得到的适应度最小值、最大值和平均值，以及KMEANS算法重复进行50次(每次的初始聚类中心独立随机产生后)，根据式(8)计算出的最小值、最大值和平均值。其中IV-PSO- K-means算法中粒子数为10个，免疫接种概率为0.2，迭代50；PSO-KMEANS算法中粒子数为10个。

表1　IV-PSO-KMEANS算法、PSO-KMEANS算法与KMEANS算法的适应度值比较

算法	最小值	最大值	平均值
KMEANS	0.818 335	1.029 640	0.972 775
PSO KMEANS	1.028 590	1.029 110	1.028 750
IV-PSO KMEANS	1.028 710	1.029 510	1.029 020

从表1可以看出，KMEANS算法因初始聚类中心不同，得到的聚类的最好和最坏结果与根据式(8)计算得到的最大值和最小值的差距很大；PSO-KMEANS的适应度值相对稳定；而本文提出的IV-PSO-KMEANS算法得到的结果更加稳定，而且适应度的平均值与KMEANS算法根据适应度函数计算得到的最大值比较接近。实验结果表明，基于免疫接种粒子群的聚类算法解决了传统聚类算法易受初始聚类中心影响的情况，聚类结果稳定，并且聚类效果比基于粒子群优化的聚类算法更加接近全局最优。

4 总　结

受人工免疫思想启发，本文提出了基于免疫接种粒子群的聚类算法，它在种群进化过程中，加入了免疫接种机制以及免疫选择机制指导粒子群的迭代过程，克服了聚类算法容易受初始聚类中心影响的缺陷，而且能更好地指导种群朝着最优方面进化。实验结果表明，基于免疫粒子群优化的聚类算法受初始聚类中心影响较小，聚类结果比较稳定，而且比基于粒子群优化的聚类算法具有更好的聚类结果。但是该算法要求用户事先确定聚类类别个数，这是一般用户没有办法完成的。如何使算法根据待处理数据确定类别个数，还有待进一步研究。

参 考 文 献

[1] KENNEDY J, EBERHART R C. Particle swarm optimization [C]//Proceedings IEEE International Conference on Neural Networks. Piscataway: IEEE Service Center, 1995.

[2] 刘靖明, 韩丽川. 粒子群优化K均值的混合聚类算法研究[J]. 中国管理科学, 2004, 12(专辑): 96-99.

[3] 刘向东, 沙秋夫, 刘勇奎, 等. 基于粒子群优化算法的聚类分析[J]. 计算机工程, 2006, 32: 201-202.

[4] VANDER M D W, ENGELBRECHT A P. Data clustering using particle swarm Optimization[C]// In: Proceedings of IEEE Congress on Evolutionary Computation 2003. Canbella: [s.n.], 2003.

[5] 毛国君, 段立娟, 王　实, 等. 数据挖掘原理与算法[M]. 北京: 清华大学出版社, 2005.

[6] JAIN A K, MURTY M N, FLYNN P J. Data clustering: A survey [J]. ACM Computer Survey, 1999, 31: 264-323.

[7] MACQUEEN J. Some methods for classification and analysis of multivariate observations[C]//In: Proceedings of the 5th Berkeley Symposium on mathematics Statistic Problem. Berkeley: [s.n.], 1967.

[8] 高　鹰, 谢胜利. 免疫粒子群优化算法[J]. 计算机工程与应用, 2004, 6: 4-6.

[9] SHI Y, EBERHART R C. A Modified particle swarm optimization[C]// In: Proceedings of the 1999 Conference on Evolutionary Computation. Piscataway: IEEE Press, 1998.

编　辑　熊思亮