基于微粒群算法的聚类算法改进

张 丽,刘希玉
(山东师范大学,山东 济南 250014)

摘 要:现有的对多维数据进行聚类的常用聚类算法,通常需要事先给定聚类数k。但在大多数情况下,聚类数k事先无法确定,因此需要对最佳聚类数k进行优化处理。采用基于微粒群算法的聚类算法。为了解决微粒群聚类算法无法确定聚类数k的现象,通过k均值算法的引入,实现最佳聚类数k的求解和聚类有效性函数的构造,试验证明引入类间距离的聚类有效性检测函数对最佳聚类数判别科学,同时由于检测函数中类间距离权重的引入使该检测函数可以更好地应用于现实数据分析。

关键词:微粒群算法;聚类优化;有效性函数;距离权重
中图分类号:TP391.41 **文献标识码**:A **文章编号**:1673-629X(2010)11-0126-04

Improved Research of Clustering Algorithm Based on PSO

ZHANG Li, LIU Xi-yu
(Shandong Normal University, Jinan 250014, China)

Abstract: The existing common clustering algorithms of multi-dimensional data usually require giving the number of clusters k in advance. However, in most cases, the number of clusters k can not be determined in advance, so the best number of clusters k needs to be optimized. Use the clustering algorithm based on particle swarm optimization. In order to solve that the clustering algorithm based on PSO can not determine the number of clusters k, by the k-means algorithm, achieve the best number of cluster k and the structuring of the cluster validity function. The testing has proved the effectiveness of cluster detection function to determine the best number of clusters, and because of the introduction of the weights of classes, the detection function can be better applied to real data analysis.

Key words: particle swarm algorithm; clustering optimization; effectiveness of the function; distance weighted

0 引 言

微粒群算法(Particle Swarm Optimization, PSO)是1995年由Kennedy和Eberhart率先提出的,起源于对简单社会系统的模拟,它借鉴了鸟群或鱼群捕食过程的社会行为,其基本思想是受他们早期对许多鸟类的群体行为进行建模与仿真研究结果的启发[1]。经过短短十年时间的发展,PSO已广泛应用于人工神经网络训练、模糊系统控制、函数优化等领域,成为目前进化计算研究的一个新热点。其中微粒群算法在聚类分析中的应用也成为研究的热点。现有的动态聚类算法易陷入局部最优,聚类效果有待提高。而微粒群算法具有很好的全局寻优性,并且算法简洁,易于实现。因而,如何使用微粒群算法来解决聚类问题,并进一步提高算法的合理性和有效性,是值得研究的课题。

收稿日期:2010-02-22;**修回日期**:2010-05-23
基金项目:国家自然科学基金资助项目(60873058);山东省自然科学基金资助项目(Z2007G03)
作者简介:张 丽(1984-),女,硕士研究生,研究方向为数据挖掘、人工智能;刘希玉,博士,教授,博导,研究方向为数据挖掘、人工智能。

1 微粒群算法简介

PSO是一种有别于GA的并行进化计算技术。PSO算法将每一个可能产生的解表述为群中的一个微粒,每个微粒都具有自己的位置向量和速度向量,以及一个由目标函数决定的适应度(Fitness)。所有微粒在搜索空间中以一定的速度飞行,通过追随当前搜索到的最优值来寻找全局最优。

在PSO算法中,算法的主要特征体现在参数的选择上。PSO算法的主要参数有种群规模s,惯性权重w,加速度常数c1、c2。现在对这些重要参数进行详细的分析。

(1)种群规模s。

种群的大小一般根据具体的实际情况而定。经过大量的实验数据的验证,一般选择20~50之间比较合适。过小的种群容易使得最优解陷入局部最优,过大的种群则加重了计算的代价,同时当种群数目达到一

定程度后,再扩大种群的大小,对整个问题的研究没有带来显著的作用。

(2)惯性权重 w。

它决定着粒子的惯性运动,使其保持扩大搜索空间的趋势,不断地探索新的区域。过小的 w 有利于局部寻优,并且精度增加;过大的 w 则正好相反,有利于全局寻优,并且收敛速度增加。

(3)加速度系数 c1、c2。

c1、c2 分别为自我认知和社会认知的系数,用以调节微粒飞向自身最好位置方向和全局最好位置方向的步长,决定微粒个体经验和群体经验对微粒运行轨迹的影响,反映微粒群之间相互信息的交流。

如果 c1 = 0,则微粒只有社会经验,故其收敛速度较快,但容易陷入局部最优;如果 c2 = 0,则微粒只有个体信息,得不到种群中其它微粒的信息;如果 c1 = c2 = 0,则微粒只能以当前的惯性一直搜索下去,直到达到终止点,一般情况下,最优解在同一方向上发展的可能性几乎为零,所以找到最优解的可能也是微乎其微的。

2　微粒群聚类算法简介

微粒群聚类算法以聚类的中心位置作为单个微粒的编码,每个微粒表示一组候选的聚类中心,依据数据量的规模和目标精度确定微粒的数目,最终确定合理的聚类结果。微粒群聚类算法采用实数编码,一个编码对应于一个可行解,也就是每个微粒的位置是由 m 个聚类中心组成,微粒除了位置之外还有速度和适应值[2]。由于样本向量维数为 d,因此微粒的位置是 $m \times d$ 维变量,微粒的速度也应当是 $m \times d$ 维变量,另外微粒编码中还包含该微粒的适应度值。由此,微粒群聚类算法的微粒编码结构如图 1 所示 。

$X_{t1}X_{t2}X_{td}\cdots X_{m1}X_{m2}\cdots X_{md}$	$V_1V_2\cdots V_{m\times d}$	J_i

图 1　微粒编码结构示意图

微粒群聚类算法的流程如下:

Step1:初始化微粒,将样本随机指派为某一类,作为初始划分,计算出各类的样本聚类中心,作为微粒的位置编码,依据位置编码计算该微粒的适应度,并随机初始化微粒的速度,重复进行 N 次(N 为种群规模),生成初始微粒。其中判断样本 x 所属归属依据最近邻法则,计算公式如下:

$$d(x_p, C_{ij}) = \min_{\forall c=1\cdots Nc}\{d(x_p, C_{ic})\} \tag{1}$$

Step2:For $t = 1$ to k do

For each particle i do

For each data vector z_p

计算每个数据 x_p 到潜在聚类中心 C_{ij} 的欧式距离 $d(x_p, m_{ij})$;

依据公式(1)判断 x_p 所归属的聚类中心 C_{ij};

使用 J_e 判断微粒的适应度值;

依据微粒群算法的进化方程更新 P_{id};

依据微粒群算法的进化方程更新 P_{gd};

更新聚类中心 C_{ij},并确定微粒编码[3]。

3　问题的提出

现有的对空间数据进行聚类的常用聚类算法,通常需要事先给定聚类数 k。但在多数情况下,聚类数 k 事先无法确定,因此需要对最佳聚类数 k 进行优化处理[4]。对于如何求解最佳聚类数 k 和构造聚类有效性函数,国内外学者给出了多种答案,可以归纳为以下几种常用的聚类有效性函数[5,6]:

(1)分离系数。

$$F(U,k) = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n}(u_{ij})^2 \tag{2}$$

假设 Ω 为所有的聚类结果,则 K 的最优选择由下式给出:

$$\max_k\{\max_{\Omega} F(U,k)\} \quad k = 2,3,\cdots,n-1 \tag{3}$$

(2)紧致性与分离性效果函数。

$$S(U,k) = \frac{\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n}u_{ij}^2 \mid x_j - c_i \mid^2}{\min\limits_{i,j} \mid c_i - c_j \mid^2} \tag{4}$$

其中,K 的最优选择由下式给出:

$$\min_k\{\max_{\Omega} S(U,k)\} \quad k = 2,3,\cdots,n-1 \tag{5}$$

在上述聚类有效性函数中,试验表明,$S(U,k)$ 的性能较好,该测试函数反映了目标样本与其对应聚类中心距离的平均值与聚类中心最小距离的比值,理想的聚类应该使聚类中心间距离尽可能的大,而样本与对应聚类中心的距离尽可能的小[7]。通过对以上测试函数的分析,提出了文中的聚类有效性函数。

一般来说,当 m 增加时,类内距离 $J_c(x,m)$ 单调递减,类间距总体上呈增加趋势。为了限制 $J_c(x,m)$ 单调,文中所构造的准则函数由类内距离加上惩罚因子 J_d,当 K 增加时,$J_c(x,m)$ 变小而 J_d 增大,迫使 K 不能等于样本数目。为了区别两项的主次,将 $J_c(x,m)$ 的权重设置为 1,而惩罚因子 J_d 的权重为 $u \leqslant 0.5$,相对应于微粒群聚类算法中的适应度函数,定义准则函数如下:

$$J_1(x,m) = J_c + uJ_{d-a} \tag{6}$$

$$J_2(x,m) = J_c + uJ_{d-b} \tag{7}$$

依据此准则函数,当 m 增加时,类内距离 $J_c(x,m)$ 减小但类间距离增大,则适应度函数的最小值所

对应的 m 就是最佳聚类数。

4 微粒群聚类算法的改进

4.1 改进算法的思想

在基本微粒群聚类算法中引入 K 均值算法，同时引入类间距离到适应度函数中，定义了一种新的聚类算法有效性函数，通过在最大候选聚类数 k_{max} 的范围内迭代搜索，选取有效性函数的最小值，在确定合理聚类数的同时返回合理的聚类结果。

文中的微粒群聚类改进算法采用实数编码[8]，一个编码对应一个可行的解，采用的是基于聚类中心的编码方式，也就是每个微粒的位置由 m 个聚类中心构成。对于空间数据 $x = \{x_i,\ i = 1,2,\cdots,n\}$，其中 x_i 为 d 维模式向量，目标聚类数是 m，设 z_i 为候选聚类中心，则单个微粒 i 的构造如下：

$$p_1 = | z_{11}z_{12}\cdots z_{1d-1}z_{1d},\cdots,z_{m1}z_{m2}\cdots z_{md-1}z_{md} |$$

该微粒对应的速度阵如下：

$$V_1 = | v_{11}v_{12}\cdots v_{1d-1}v_{1d},\cdots,v_{m1}v_{m2}\cdots v_{md-1}v_{md} |$$

4.2 改进算法的流程

改进算法的流程如下描述：

(1)根据样本的数量 n，确定聚类数 m 的取值范围。通常 m 最大值不超过$\sqrt{n}$。

(2) 依据对应的 m 值确定微粒的维数，并且进行微粒群的初始化。先将每个样本随机指派到某一类作为最初的聚类划分，并计算各类的聚类中心作为初始微粒的坐标，计算微粒的适应度并初始化微粒的速度，反复进行 t 次生成规模为 t 的微粒群。其中对第 j 个微粒而言，它具有随机给定的位置 x 和速度 v。而该微粒的位置编码既是求得的聚类中心值。由微粒的编码根据公式求得该微粒对应的适应度的值。

(3) 初始化微粒群的局部最优位置 p_i 和全局最优位置 p_g。初始的局部最优位置 p_i 为初始化后的微粒及其对应适应值。初始全局最优位置 p_g 为初始局部最优 p_i 中适应度最小的微粒及其对应适应值。

(4) 更新微粒的速度和位置，对于微粒的优化，引入K均值算法进行优化：根据微粒的聚类中心编码，按照近邻法则，来确定对应该微粒的聚类划分；按照聚类划分，计算新的聚类中心，更新微粒的适应度值，取代原来的编码值。

如果在过程中可能出现空聚类，则从样本数目最多的非空聚类中选取离聚类中心最远的样本作为聚类中心，替换原空聚类，重复替换过程，直到新微粒中无空聚类[9]。

(5) 对优化的微粒比较它的适应度值和它经历过的最好位置 p_i 的适应度值，如果更好更新 p_i；对每个微粒比较它的适应度值和群体所经历的最好位置 p_g 的适应度值，如果更好更新 p_g。

(6) 重复(3) 到(4) 直到 C_i 中的微粒达到终止条件(最大迭代次数或足够好的位置)。

(7) 根据返回适应度的值确定最佳聚类数并返回聚类结果[6]。

5 实验分析

5.1 聚类数的实验分析

文中采用数据组合对所提出的改进算法进行验证。数据组合的参数设置如下：加速常数 C_1、C_2 分别取 2，微粒群的规模为 20，最大迭代代数为 400，最大速度 v_max 设定为 2，惯性权重 w 按以下公式随迭代进行更新：

$$w = w\_max - (w\_max - w\_min)xi / max\_gen$$

其中，w_min 为 0.4，w_max 为 0.9，max_gen 为 400。权重 u 的取值范围为 u≤0.5。

组合数据集有 4 类，每个类含有 100 个样本，3 个类中心分别为[3.00,3.00]、[8.00,4.00]、[9.00,9.00]。每一类样本皆为类中心各维加上服从标准正态分布 N(0,1)的数据点构成[10]。数据集通过文中确定的合理聚类数如图 2 所示。

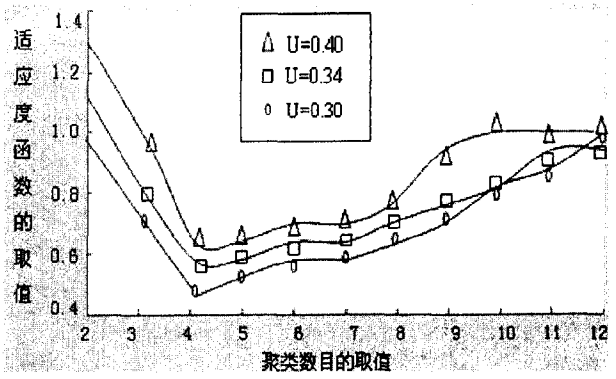

图 2 数据集适应度函数 J1 的变化曲线图

从图 2 可以显示文中提出的算法通过搜索给定的适应度函数的最小值可以正确地判定所给聚类的最佳聚类数。就单个数据集而言，依据两种不同的类间距离确定的聚类有效性函数对于最佳聚类数的确定均到理想结果。同时类间距离权重 u 虽然改变了适应度函数的值，但并未改变最佳聚类数。

5.2 聚类结果的实验分析

为了检验该聚类算法是否有效，文中建立了测试模型并对原始算法和改进的算法进行了对比实验[11]。实验数据 wine 选自 UCI 数据库，该测试数据集样本数为 178，条件属性个数为 13，聚类类别为 3，类别包括

的样本数分别为 59,71 和 48,见表 1。

表 1　wine 数据

no	1	2	3	4	…	11	12	13
1	14.23	1.71	2.43	15.6	…	1.04	3.92	1065
2	13.2	1.78	2.14	11.2	…	1.05	3.4	1050
3	13.16	2.36	2.67	18.6	…	1.03	3.17	1186
4	14.37	1.95	2.51	6.8	…	0.86	3.45	1480
…	…	…	…	…	…	…	…	…
178	14.13	4.1	2.74	24.5	…	0.61	1.6	560

文中用聚类算法和微粒群聚类改进算法对 wine 数据进行处理,其中学习因子 c1 = c2 = 2,惯性权重 $\omega_{max}=0.9$,$\omega_{min}=0.1$,实验结果如表 2 所示。

表 2　聚类结果对比

类别	样本个数	聚类算法			微粒群聚类改进算法		
		正确(个)	不正确(个)	准确率	正确(个)	不正确(个)	准确率
第一类	59	46	13		56	3	
第二类	71	63	8	0.7753	63	8	0.9382
第三类	48	29	19		48	0	

文中把 wine 的 178 个数据分成三组进行聚类,从聚类结果来看,用普通聚类算法进行聚类准确率较低,出错率较高,而微粒群聚类改进算法能够得到较高的且稳定的准确率,改进效果十分明显。

6　结束语

微粒群算法作为一个新的全局寻优算法,具有很强的寻优能力,将微粒群算法引入聚类算法中克服了现有聚类算法的容易陷入局部最优等缺陷。

文中在基本微粒群聚类算法的基础上,定义了两种类间距离的计算方法,并在此基础上定义了一种新的检测聚类算法的有效性函数,通过改进的聚类算法,可以得到更加准确的聚类数,同时可以得到更加准确的聚类结果。试验结果验证了改进算法的优越性以及有效性。

参考文献:

[1] 路克中,张秋华,孙兰娟.一种改进的粒子群优化算法及其仿真[J].计算机技术与发展,2007,17(11):223－225.

[2] 王胥鹏, 胡劲松. 一种改进的微粒群算法[J].计算机应用研究, 2009(10):113－116.

[3] 耶刚强,孙世宇,梁　彦,等.基于动态粒子数的微粒群优化算法[J].信息与控制,2008(1):93－99.

[4] Guha S, Rastogi R, Shim K. An Efficient Clustering Algorithm for Large Databases[C]//Proceedings of the ACM SIGMOD Conference, Int l Conf. on Management Ent of Data. Atlantic City:[s. n.],2008:73－84.

[5] 曾建潮,介　婧,崔志华.微粒群算法[M].北京:科学出版社,2004.

[6] Shi Y, Eberhart R C. Parameter Selection in Particle Swarm Optimization [C]//Evolutionary Programming Vll: Proc. EP98. New York:Springer Verlag,1998:591－600.

[7] Guha S, Rastogi R, Shim K. A Robust Clustering Algorithm for Categorical Attributes [C]//Proceedings of the 15th ICDE. Australia:[s. n.],2007:512－521.

[8] Hu Jing song, Hu Gui wu, Wang Jia bing. FCMAC based on mine－sweeping strategy[C]//Proc. of International Conference on Machine Learning and Cybernetics. Hong Kong: IEEE Press,2005:784－787.

[9] Han Jawei, Kamber M. 数据挖掘—概念与技术[M].范明,孟小锋 译.北京:机械工业出版社,2001:223－261.

[10] 袁代林,程世娟,陈　虬. 一种新形式的微粒群算法[J].计算机工程与应用,2008(33):234－239.

[11] 张更新,赵　辉,王红君,等.基于动态参数的微粒群算法的研究[J].天津理工大学学报,2005,21(4):42－44.

(上接第 125 页)

[M]. Amsterdam, The Netherlands: Elsevier Science Publishers B. V., 2000.

[9] 文坤梅, 卢正鼎, 吴杰文, 等. 基于描述逻辑的推理系统设计与实现[J]. 小型微型计算机系统,2008, 29(1):57－58.

[10] Wolter F, Zakharyaschev M. Satisfiability problem in description logics wit h modal operators[C]//Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning. San Francisco: Morgan Kaufmann Publisher, 1998: 512－523.

[11] Wang Ju, Jiang Yuncheng, Shen Yuming. Satisfiability and reasoning mechanism of terminological cycles in description logic vL[J]. Science in China, Ser. F,2009, 39(2): 206－207.

[12] Winslett M. Reasoning about action using a possible models approach[C]//Proceedings of the 7th National Conference on Artificial Intelligence. St. Paul, Minnesota:[s. n.], 1988: 89－93.

[13] 石　莲, 孙吉贵. 描述逻辑综述[J].计算机科学,2006, 33(1):194－196.

[14] Pothipruk P. Query Answering for Multiple Complex Resources: Description Logic in the Semantic Web Context[D]. Canberra, Australia: University of Queensland, 2007.

[15] Nutt W. Algorithms for constraints in deduction and knowledge representation[D]. Saarbrücken, Germany: University of Saarland, 1993.