第 26 卷第 6 期　　四川理工学院学报(自然科学版)　　Vol. 26　No. 6
2013 年 12 月　　Journal of Sichuan University of Science & Engineering(Natural Science Edition)　　Dec. 2013
文章编号:1673-1549(2013)06-0028-05　　DOI:10.3969/j.issn.1673-1549.2013.06.006

基于 k-means 的量子微粒群动态聚类

付柳强, 张洪伟,徐开阔

(成都信息工程学院计算机学院, 成都 610225)

摘　要:k-means 算法原理简单、收敛速度快,但易陷入局部最优,且须将聚类的类簇数作为先验知识,为此,引入量子微粒群与 k-means 算法结合,提出了一种改进的动态聚类算法。改进算法具有量子微粒群的全局搜索能力,且对每个粒子采用 k-means 进行优化,从而加快算法的收敛速度。通过适应度函数值的调整,算法在聚类中能够搜寻到最优类簇数,这样类簇个数和中心就不受主观因素的影响。实验表明,算法有效。

关键词:k-means;量子微粒群;动态聚类;全局搜索

中图分类号:TP306.1　　**文献标志码**:A

引　言

聚类分析是将给定的数据集按照数据的内在性质划分为若干组或类簇的过程,类簇中的每一个样本数据称为一个对象。聚类目的是使同一簇中对象的特性尽可能相似,不同簇对象间的特性差异尽可能地大。相似或不相似的度量是基于数据对象的描述属性的取值来确定。在数据聚类中,通常采用距离来度量数据相似性,通过类簇内样本数据的紧密性和类簇间的离散性,来评价聚类的有效性。总体上说,聚类算法分为层次聚类和非层次的分割聚类两大类,层次划分是将数据集划分成不同的层次,可以自底向上也可自顶向下划分,形成类似树状结构,但数据一旦划分就不再改变,不利于寻优;而分割划分则是按着某种准则,将数据集划分成一定数目的不重叠的子集,可多次划分,以期最优。

k-means 聚类是由 MacQueen 提出的基于分割的非监督硬聚类算法[1],在准则函数的基础上将数据集划分成预定的类数。其原理简单,收敛速度快,在很多领域有良好的应用,但 M 值的选取依赖于数据集的先验知识,容易陷入局部最优。算法运行前,须先指定类簇数目 M,并初始化 M 个聚类中心,再根据一定的相似性度量准则,将每个样本分配到最近的类簇中,一次划分结束,更新 K 个聚类中心。重复执行该过程,直到算法达到最大迭代次数或者类簇稳定(聚类中心几乎不再改变或者改变很小)。虽然该算法效率高、具有线性的时间复杂度,但其须预先指定类簇数,而现实中很多的数据集无法预先确定其最佳类簇数,基于此,本文提出了一种基于 k-means 改进的动态聚类算法(dynamic clustering based on k-means with QPSO,CDKQPSO),算法尝试着将量子微粒群与 k-means 结合,并通过改进粒子的编码方式,使得聚类数目 M 随着迭代次数动态改变,以期聚类过程中自动确定最佳的聚类数目。

1　量子微粒群简介

微粒群优化(particle swarm optimization,PSO) 算法是由 Eberhart 和 Kennedy 等人[2]基于鸟群觅食行为提出的一种群体智能优化算法。该算法概念简明、模型简单、实现方便、收敛速度较快、参数设置少,而且,粒子在解空间能够追寻最优的粒子进行搜索,记忆粒子历史最优和全局最优的信息,是一种高效的搜索算法。但其仍

收稿日期:2013-06-13

基金项目:成都信息工程学院发展基金(KYTZ201108)

作者简介:付柳强(1987-),男,江西丰城人,硕士生,主要从事智能工程、数据挖掘方面的研究,(E-mail)flq870905@163.com

不是一个全局收敛算法[3],且算法的鲁棒性依赖于速度和位置上限的设置。因此,Sun 等人[4]从量子力学的角度,提出了一种新概率 PSO 算法——量子微粒群(quantum - behaved particle swarm optimization, QPSO)算法。该算法以 Delta 势阱为基础,认为粒子具有量子行为,在量子空间依靠群体智能搜索优化问题的解。已经取得的研究成果表明 QPSO 算法是一种全局收敛算法,目前在很多领域已经取得了优于 PSO 算法的效果。

1.1 PSO 模型

PSO 是群体智能(Swarm Intelligence, SI)的并行搜索技术,每个粒子在 D 维搜索空间追随两个"极值"进行搜索。在每个具体的时刻,每个粒子都有自己的位置 $X_i = (X_{i1}, X_{i2}, \cdots, X_{iD})$ 和速度 $V_i = (V_{i1}, V_{i2}, \cdots, V_{iD})$,并能记忆两个极值:一个是粒子本身所经历的最优解 $P_i = (P_{i1}, P_{i2}, \cdots, P_{iD})$,一个是整个种群目前所能找到的全局最优解 $P_g = (P_1, P_2, \cdots, P_D)$。在算法迭代过程中,每个粒子都追随着这两个极值进行搜索,并通过下面两个公式来更新自己的位置:

$$V_{ij}(t+1) = \omega V_{ij}(t) + c_1 r_1 [P_{ij}(t) - X_{ij}(t)] + c_2 r_2 [P_{gj} - X_{ij}(t)] \quad (1)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad (2)$$

其中,i 是粒子编号,$j = 1,2,\cdots,D$, c_1, c_2 是加速因子,为正常数,r_1, r_2 为[0 -1]之间的随机数;ω 称为惯性因子。

1.2 QPSO 模型

QPSO 作为 PSO 的改进模型算法,也是种基于 SI 的并行搜索技术,其粒子在量子空间搜索可行解,由于量子粒子的不确定性,量子空间的概率随机性,粒子的位置和速度无法同时被表征,但可采用蒙特卡罗模拟的方法来模拟粒子的位置,故量子微粒群采用下面三个公式来更新自己的位置:

$$mbest = \frac{1}{N}\sum_{i=1}^{N} P_i = \left(\frac{1}{N}\sum_{i=1}^{N} P_{i1}, \frac{1}{N}\sum_{i=1}^{N} P_{i2}, \cdots, \frac{1}{N}\sum_{i=1}^{N} P_{iD}\right) \quad (3)$$

$$p_{ij} = \varphi \times P_{ij} + (1 - \varphi) \times P_{gj} \quad (4)$$

$$X_{ij}(t+1) = p_{ij} \pm \alpha \left| mbest_j - X_{ij}(t) \right| \times \ln(1/u) \quad (5)$$

其中:$i = 1,2,\cdots,N$, $j = 1,2,\cdots,D$, N 是粒子群中的粒子数,D 是解空间的维数。$mbest$ 是所有粒子的最优位置的均值,p_{ij} 是粒子 i 的最优位置与全局最优位置之间的某个随机位置点。φ, u 是在[0 -1]之间的随机数,若对应的 $u > 0.5$,则式(5)取减号,否则取加号。α 是 QPSO 中的收缩扩张因子,也是迭代公式中唯一的参数,一般是让其随着迭代次数线性减少。

$$\alpha = (\alpha_{\max} - \alpha_{\min}) \times \frac{(T - t)}{T} + \alpha_{\min} \quad (6)$$

其中 T 是最大迭代数,$\alpha_{\max}$, $\alpha_{\min}$ 取 1.0 和 0.5。

2 算法描述

2.1 聚类的数学描述

设样本集 $X = \{X_i = (X_{i1}, \cdots, X_{iD}), i = 1, \cdots, n\}$ 其中 D 为样本特征属性的维数,样本的总数为 n,聚类过程中可能被分成的类簇数目为 M, $C_i = \{X_{k_j} \in X, 1 \leqslant k_j \leqslant n\}$, $i = 1, \cdots, M, M \leqslant n$, 从而:$X = \cup C_i = \{X_i = (X_{i1}, \cdots, X_{iD}), i = 1, \cdots, n\}$, $C_i = \{X_i = (X_{i1}, \cdots, X_{iD}), i = 1, \cdots, m\} \neq \Phi$, $C_i \cap C_j \neq \Phi, i \neq j$, 其中:$C_r$ 的中心 $m_r = \frac{1}{n_r}\left(\sum_{i=1}^{n_r} X_{i1}^r, \sum_{i=1}^{n_r} X_{i2}^r, \cdots, \sum_{i=1}^{n_r} X_{iD}^r\right)$, C_i 表示聚类过程中形成的第 i 个类簇,m_r 表示类簇 C_r 的中心。

2.2 粒子的编码

粒子的编码采用的是 2002 年 Omran 等人[5]在基于 PSO 聚类算法中提出的基于中心的实数编码规则,每个粒子对应于数据集的一次候选划分,也就是每个粒子包含有 M 个有效的类簇中心,由于在各个粒子中划分的类簇数 M 是不一定相同的,因此,设定一个最大可能的类簇数 $M_{\max}$, 则每个粒子可以定义为一个 $M_{\max} * D$ 的向量。即粒子可编码为:

$$e = (m_1, m_2, \cdots, m_{Mmax}) \quad (7)$$

若已知粒子 $e = (m_1, m_2, \cdots, m_{Mmax})$, m_r 是簇 C_r 的中心,则样本 X_i 按最近邻原则被指定在第 r 类, $X_i \in C_r$, 得 X 的一次划分 $X = \cup C_r$。本文的算法中,每次迭代的类簇数 M 是不同的,采用公式:

$$M = M_{\max} - ((t-1) \bmod (M_{\max} - M_{\min} + 1)) \quad (8)$$

式(8)中,t 表示当前的迭代次数,$M_{\max}$, $M_{\min}$ 分别表示可能被分的最大最小的类簇数目。M 随着迭代次数 t 在 $M_{\max}$ 和 $M_{\max}$ 中循环的改变。迭代过程中,粒子的有效聚类中心都是前 M 个。当 M 不等于最小类簇数时,每次都要将已经获得的类簇中具有最少样本数的类簇重新划分给其他类簇,这样就减少了一个类簇;若 M 为最小类簇的时候,需要补加 $M_{\max} - M_{\min}$ 个类簇中心,下次迭代就有个 $M_{\max}$ 类簇,如此,往复循环,就实现了每次类簇数动态改变。M 控制着粒子实际参与聚类的有效聚类中心,而 M 又可以被当前迭代次数表征。为加快粒子收敛,使用 k-means 对每个粒子进行优化,并将优化后的中心覆盖公式(5)更新的粒子中心。

2.3 定义适应度函数

适应度函数是用来评价聚类算法的有效性,从而确

认聚类的结果,以找出适合原始数据集的最佳划分。因此,类簇的有效性确认方法能够对一种聚类算法求得的结果进行量化评估。在已有的众多有效的评估函数中,本文引入 Davies – Bouldin's Index (DB 指数)[6-7] 函数来评价,该方法是通过类簇内紧密度和类簇间的分离度之比来实现的,其定义如下:

簇内紧密度:

$$S_i = \frac{1}{n_i}\sum_{X_r \in C_i} d(X_r, m_i) \tag{9}$$

簇间分离度:

$$S_{ij} = d(m_i, m_j) \tag{10}$$

$$R_i = \max_{i,j \neq i} \frac{S_i + S_j}{S_{ij}} \tag{11}$$

$$DB(M) = \frac{1}{M}\sum_{i=1}^{M} R_i \tag{12}$$

其中: n_i 表示第 i 个类簇中的样本个数, m_i 是第 i 个类簇的中心, C_i 表示第 i 个样本子集类簇, $d(\cdot)$ 是欧式距离,M 是类簇数。

DB 指数在 $[0, +\infty]$ 取值,可以用于比较具有不同类簇数之间的划分的优劣,DB 值越小,表示聚类效果越好。因此,可以定义适应度函数:

$$fitness = DB(M) \tag{13}$$

适应度值越小,表示聚类效果越好。从公式(12)可以看出,由于 DB 指数是基于类簇内和类簇之间的比例,所以,当类簇小于 2 时,DB 指数是无意义的,因此在算法的迭代过程中,需要确保 $M_{\min} \geqslant 2$。

3 算法步骤

(1) 初始化。初始化各个参数:种群规模 N、聚类类簇数上下限 $M_{\max}$ 和 $M_{\min}$、扩张收缩因子 α、随机参数 φ 与 u; 初始化粒子种群:粒子的编码采用上述的规则,粒子种群的初始化采用 k-means 聚类的返回的结果进行初始化,对每个粒子,先对样本数据集执行一次 k-means,获得 M 个聚类中心,将这 M 个聚类编码成一个粒子。重复 N 次,获得粒子种群。

(2) 计算粒子的适应度值 $fitness_i$。如果 $fitness_i$ 大于 P_i 的适应度值,则更新当前粒子的个体最优位置; 若 $fitness_i$ 大于 P_g 的适应度值,则用当前位置更新种群的全局最优位置。

(3) 利用公式(3)计算种群所有粒子的平均位置。

(4) 根据公式(4)与公式(5)更新粒子的位置。

(5) 对每个粒子用 k-means 进行优化,实现动态聚类,而且由于 k-means 具有强的局部搜索能力,这样能加快粒子的收敛速度。

首先,根据此时的 M 值,获取粒子的前 M 个类簇中心,按着最邻近法则,对样本数据集进行划分,并且记录各个类簇中的样本个数。当 M 不等于 $M_{\min}$, 则把样本数目最少的类簇中的样本划分给紧邻的其他类簇,从而类簇数减 1,获得 $M-1$ 个类簇中心; 当 M 等于 $M_{\min}$, 则此时具有最小的类簇数,但下次的迭代须有 $M_{\max}$ 个类簇中心。对样本执行一次 k-means,获取 $M_{\max} - M_{\min}$ 个类簇中心,将前面 $M_{\min}$ 个及获取的 $M_{\max} - M_{\min}$ 个类簇中心一起,编码成粒子。利用 $M_{\max}$ 个类簇中心,按着最邻近法则对样本再次进行划分。

然后,按照样本的划分,重新计算各类簇中心。若出现空类簇(按最临近原则此时可能出现某个类簇没有样本),则从最开始用 k-means 初始化的类簇中心,随机选择一个。将获得的类簇中心覆盖粒子的对应的类簇中心。

(6) 若达到算法最大迭代数,结束算法,输出全局最优解,否则,转到步骤(2)。

上述步骤的流程图如图 1 所示。

基于 K 均值的量子微粒群算法在产生下一代粒子时有很强的随机性,所以在很大程度上克服了 K 均值易陷入局部极小值的问题。而且每个粒子在迭代过程参考的是全局最优、自己最优以及整个种群最优的均值,在具备“自我学习”和向“社会学习”的优点的同时,又保证了不会突破种群,成为远离种群的孤立点,因此算法在运行过程有良好的随机性。在算法中,用 K 均值对新粒子个体优化,进一步加快了算法的收敛性能。

4 实验及结果

算法验证共测试了 4 个数据集,包含一个随机产生的数据集 dataset 和 UCI 中 iris、wine、glass 三个数据集(表 1)。随机产生的数据集 dataset 中共有 4 个类簇,每个类簇有 30 个样本数据,样本具有 3 维特征属性; iris 样本集包含 150 个样本数据,每个样本数据有 4 维特征属性,分成 3 类,每一类植物有 50 个; wine 数据集样本个数 178,包含了 13 个特征属性,分成 3 类; glass 数据集中共有 214 个样本数据,包含了 6 种不同的玻璃制品,每个样本有 9 维特征属性。

表 1 样本集的特性

样本集	样本个数	特征维数	类数
dataset	120	3	4
iris	150	4	3
wine	178	13	3
glass	214	9	6

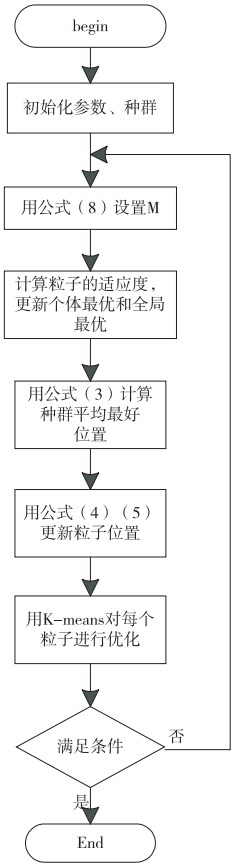

图 1　算法流程示意图

算法测试实验环境为 MATLAB2009a,Intel Core i5 @ 3.20GHz,4.0GRAM。类簇数目上下限 $M_{max}=10$, $M_{min}=2$;收缩扩张因子上下限 $\alpha_{max}=1.0$, $\alpha_{min}=0.5$; $T=1000$。每个数据集运行 30 次,UCI 的 Iris,glass 进行了归一化处理。结果取 30 次均值(表 2)。

表 2　聚类结果

数据集	算法	类簇数	DB	迭代次数
dataset	KPSO	4	7.97463	20
	KQPSO	4	7.97463	15
	CDKQPSO	4	8.62965	600
iris	KPSO	3	32.20805	100
	KQPSO	3	32.18232	60
	CDKQPSO	3.2	32.22811	800
wine	KPSO	3	39.75897	120
	KQPSO	3	37.47355	100
	CDKQPSO	3.9	68.63963	900
glass	KPSO	6	55.37359	100
	KQPSO	6	55.237784	90
	CDKQPSO	5.3	58.81264	800

基于 k-means 的 PSO[7-8] 与 QPSO 算法对样本集聚类时候需要给定 M 值,本文算法 CDKQPSO 采用公式(8)来确定 M 值,极大地降低了算法对初始条件的依赖性。从表 2 结果看出,算法基本能找到较优的类簇数。由于样本集 dataset 中的样本具有类簇内紧凑、类簇间离散的特点,且没有重叠的样本,而数据集 iris、wine、glass 中存在重叠样本,增加了算法的复杂性,因此由算法 CDKQPSO动态确定的类簇数没有 dataset 样本集准确、稳定。dataset 样本集在 30 次的测试中,有 28 次能够确定类簇数为 4;iris 数据集在 30 次测试中有 23 次确定类簇数为 3,其他实验中类簇数基本落在 2 和 4 之间,只有个别类簇数为 5。而样本集 wine、glass 中的样本由于重叠率较高,30 次实验中确定类簇数为 3 或 6 的次数较少,但均值较为接近 3 与 6。本文算法 CDKQPSO 确定的较优类簇数的均值与预设的较优类簇数非常相近,DB 值也较为接近,说明本文算法在迭代过程中,能够动态寻找到较优的类簇数,且结果也较为稳定。

5　结束语

k-means 是数据挖掘的经典算法,在聚类分析中有着广泛的应用。k-means 聚类是硬聚类,每个样本只能被划分到一个类簇,而且必须将类簇数目作为先验知识,这限制该算法的应用,本文提出的算法,能够克服这一限制,让算法在运行中能够自动最佳的类簇数目,并把样本准确地分到相应的类簇中。虽然实验证明了算法有效性,但是仍有很多地方需要进一步改进:(1)本文算法对于类簇是圆形的样本集具有较好的效果但较耗时,对于不规则类簇的样本集实验效果较差;(2)在动态确定 M 值过程中采用 k-means 优化类簇中心时易出现空类簇,浪费资源。

参 考 文 献:

[1] Macqueen J.Some methods for classification and analysis of multivariate observations[C]// Lucien M,Le C, Jerzy N.Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,Volume 1:Statistics,Berkeley,June 21-18,1965:281-297.

[2] Kennedy J,Eberhart R C.Particle swarm optimization [C]//IEEE.Proceeding of IEEE International Conference on Neural Networks,Perth,WA, Nov.27-Dec.1,1995: 1942-1948.

[3] Bergh F V d.An analysis of particle swarm optimeziers [D].South Africa:Universtiy of Pretoria,2001.

[4] Sun jun,Feng Bin,Xu wenbo.Particle swarm optimization with particles having quantum behavior[C]//IEEE.Proc of Congress on Evolutionary Computation,USA,June 19-23,2004:325-331.

[5] 李峻金,向 阳,芦英明.粒子群聚类算法综述[J].计算机应用研究,2009:26(12):4423-4427.

[6] Davies D L,Bouldin D W.A cluser separation measure [J].IEEE Trans on Pattern,Anaylsis and Machine Intelligence,1979,PAM(2):224-227.

[7] 陈 伟,陈 璟,孙 俊,等.一种量子行为粒子群优化动态聚类算法[J].计算机应用研究,2011,28(7):2432-2435.

[8] 刘靖明,韩丽川,侯立文.基于粒子群的 K 均值聚类算法[J].系统工程理论与实践,2005(6):54-58.

Quantum-behaved Particle Swarm Dynamic Clustering Based on K-means

FU Liu-qiang, *ZHANG Hong-wei*, *XU Kai-kuo*

(College of Computer, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: The traditional k-means is a simple and fast algorithm, but it's easy to get stuck at locally optimal value. Furthermore, the class number of clusters must be the priori knowledge in the k-means algorithm. Therefore, the quantum particle swarm algorithm combined with the k-means is introduced and an improved algorithm for dynamic clustering is put forward. The improved algorithm has global search capability that quantum-behaved particle swarm algorithm has. In addition, each particle trained by k-means algorithm is optimized, and the convergence speed is accelerated. Through the adjustment of the value of the fitness function, the algorithm can search for the optimal clustering number of clusters, so that the number of clusters and centers are not subject to subjective factors. The experiments show that the algorithm is effective.

Key words: k-means; quantum-behaved particle swarm; dynamic clustering; global search