一种自适应惯性权重的并行粒子群聚类算法

廖子贞,罗 可,周飞红,傅 平
LIAO Zi-zhen,LUO Ke,ZHOU Fei-hong,FU Ping

长沙理工大学 计算机与通信工程学院,长沙 410076
Computer & Communication Engineering College,Changsha University of Science and Technology,Changssha 410076,China

LIAO Zi-zhen,LUO Ke,ZHOU Fei-hong,et al.Cluster algorithm based on parallel particle swarm optimizer using adaptive inertia weight.Computer Engineering and Applications,2007,43(28):166-168.

Abstract: Because of the defects of K-means cluster method and the cluster method based on genetic algorithm and the fact proved by experiments that the particle swarm optimization is superior to the genetic algorithm while solving the problems of real optimization,the cluster algorithm based on parallel particle swarm optimizer using adaptive inertia weight is proposed in this paper.Theoretics and experiments show that the proposed algorithm is obviously superior to the cluster method based on genetic algorithm since it have faster convergence rate and higher convergence accuracy.

Key words: Cluster Analysis;K-means;Genetic Algorithm;Particle Swam Optimization Algorithm;Parallel Computing

摘 要:针对 K-means 聚类算法和基于遗传(GA)的聚类算法的一些缺点,及求解实优化问题时粒子群算法优于遗传算法这一事实,提出了一种自适应惯性权重的并行粒子群聚类算法。理论分析和实验表明,该算法在收敛速度和收敛精度方面明显优于基于遗传算法的聚类方法。

关键词:聚类分析;K-均值;遗传算法;粒子群优化算法;并行计算
文章编号:1002-8331(2007)28-0166-03 **文献标识码**:A **中图分类号**:TP301

1 引言

K-means 算法[1]是一种聚类分析的常用方法,对小样本非常有效。然而,K-Means 算法是一种局部搜索技术,它可能受初始中心的影响而过早的收敛于局部最优解[2],尤其在大矢量空间中这种算法的性能会变得很差。为了克服上述缺点,很多学者提出了基于遗传算法的聚类分析方法,以及一些改进变异或交叉算子的算法。但是,很多实验表明,由于增加了交叉和变异步骤,使得算法时间增长,而且当样本数量、维数较大时,这些算法容易过早收敛于局部最优解。当算法出现早熟时,仅仅依靠较小的变异概率很难从局部最优解中跳出。遗传算法在进化过程中可能产生退化想象,将导致迭代次数过长及聚类结果精确程度不高,并且可能产生后期的波动。

粒子群优化算法(PSO)[3,4]是一种进化计算技术,由 Eberhart 博士和 kennedy 博士发明, 源于对鸟群捕食行为的研究,是一种基于叠代的优化工具,拥有记忆特点。同遗传算法相比,PSO 算法没有许多参数需要调整,不但具有遗传算法的全局寻优能力,同时还具有较强的局部寻优能力,并且 PSO 算法没有遗传算法用的交叉以及变异, 而是根据自己的速度来决定搜索。PSO 算法的编码采用浮点数编码,没有遗传算法中的编码和解码过程, 因此大多数情况下 PSO 算法比遗传算法更快收敛于最优解,且可以完全避免随机寻优的退化现象。本文对基本 PSO 算法进行改进,并与聚类算法结合,提出了一种自适应惯性权重的并行粒子群聚类算法,实验结果对 K-means 算法、遗传聚类算法、基本粒子群聚类算法和本文算法进行了比较,表明了本文算法的有效性。

2 聚类算法的数学模型

设待聚类的样本空间为 $X=\{x_1,x_2,\cdots,x_n\}$,其中样本 $x_i=(x_{i1},x_{i2},\cdots,x_{ip})$,$(i=1,2,\cdots,n)$为 P 维特征空间 R^P 中的一个点。聚类问题就是要找到一个划分 $C=\{C_1,C_2,\cdots,C_k\}$,满足:

$$X=\bigcup_{i=1}^{K}C_i$$

$$C_i\neq\varnothing \quad (i=1,2,\cdots,K)$$

$$C_i\cap C_j=\varnothing \quad (i,j=1,2,\cdots,K;i\neq j)$$

并且使得总的类间距离和:

$$J=\sum_{j=1}^{K}J_j=\sum_{j=1}^{K}\Big(\sum_{x_i\in C_j}\|x_i-z_j\|\Big) \tag{1}$$

达到最小,其中,z_j 表示第 j 个聚类中心;$\|x_i-z_j\|$,表示第 i 个数据点 x_j 到聚类中心 z_j 的距离。

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60474070,No.10471036);湖南省科技计划项目基金(No.05FJ3074)。

作者简介:廖子贞(1981-),男,硕士,主要研究方向为数据挖掘;罗可(1961-),男,博士,教授,研究方向:数据挖掘,数据库技术;周飞红(1980-),女,硕士,主要研究方向为电子系统设计与集成、现代网络与通信技术;傅平(1982-),女,硕士研究生,研究方向:数据挖掘,数据库技术。

3　粒子群优化算法(PSO)

3.1　基本 PSO 算法

PSO 算法是由美国的 Eberhart 和 kennedy 提出的一种模拟鸟群捕食行为的智能算法。设想这样一个场景：一群鸟在随机搜索食物，并且在这个区域里只有一块食物，所有的鸟都不知道食物在哪里，但是它们知道当前的位置离食物还有多远，那么找到食物的最优策略是什么呢？最简单有效的办法就是搜寻目前离食物最近的鸟的周围区域。PSO 算法就是从这种模型中得到启示并用于解决优化问题。

在基本 PSO 算法中，每个优化问题的解都是搜索空间中的一只鸟，称之为“粒子”，所有的粒子都有一个由被优化的函数决定的适应值，每个粒子还有一个速度决定他们飞翔的方向和距离，粒子通过不断调整自己的位置来搜索新解。

设每个个体是 D 维向量，第 i 个粒子表示为 $X_i=(x_{i1},x_{i2},\cdots,x_{iD})$；它经历过的最好位置(当前局部最优解)，记做 $P_i=(P_{i1},P_{i2},\cdots,P_{iD})$；整个粒子群经历过的最好的位置(当前全局最优解)，记做 $P_g=(P_{g1},P_{g2},\cdots,P_{gD})$；第 i 个粒子的飞翔速度，记作 $V_i=(V_{i1},V_{i2},\cdots,V_{iD})$。则第 i 个粒子在 $d(1\leqslant d\leqslant D)$ 维上位置和速度更新操作如下：

$$V_{id}=\omega V_{id}+c_1 rand()(P_{id}-X_{id})+c_2 rand()(P_{gd}-X_{id}) \quad (2)$$

$$X_{id}=X_{id}+V_{id} \quad (3)$$

其中，ω 为惯性权重，一般取 $\omega\in[0.4,1]$，c_1、c_2 为学习因子，一般取 $c_1,c_2\in[0,2]$，$rand()$ 是[0,1]之间均匀分布的随机数。为了防止粒子飞行速度过快而导致算法过早收敛到局部最优解，每一维粒子的速度的绝对值都会被限制在一个最大速度 $V_{\max}>0$ 范围内，如果某一维更新后的速度的绝对值超过用户设定的 $V_{\max}$，那么这一维的速度就被限定为 $V_{\max}$ 或 $-V_{\max}$(如果速度为负)。设置较大的 $V_{\max}$ 可以保证粒子群的全局搜索能力，而较小的 $V_{\max}$ 使粒子群局部搜索能力增强。

3.2　自适应惯性权重的并行粒子群优化算法

3.2.1　自适应调整惯性权重

在基本 PSO 算法中，惯性权重 ω 的设置对算法的收敛速度和算法结果有很大影响，较大的 ω 有利于跳出局部最优解，较小的 ω 有利于算法收敛。有学者通过线性方程或采用压缩因子的方法来逐渐减小 ω。但是由于仅仅减小 ω，使得函数一旦进入局部极值点邻域内就很难跳出，极易收敛到局部最优解。笔者通过大量实验发现 ω 与局部适应度和全局适应度密切相关，因此提出了一种自适应非线性调整惯性权重的方法。

设 $\omega\in[a,b]$(一般取 $\omega\in[0.4,1]$)，$f_{gd}(t)$ 表示第 t 代所有粒子的全局最优适应度，$f_{id}(t)$ 表示第 t 代的第 i 个粒子局部最优适应度，则有如下计算第 t 代第 i 个粒子的惯性权重 $\omega_i(t)$ 公式：

$$f_i(t)=k\times\frac{f_{gd}(t)}{f_{id}(t)}$$

$$\omega_i(t)=b-(b-a)\times e^{(-f_i(t)/t)} \quad (4)$$

其中，k 为大于 0 的常数，t 为当前迭代的代数。

由数学极限定理易知，$0<e^{(-f_i(t)/t)}<1$，从而保证了 $\omega_i(t)\in[a,b]$。很显然 $f_i(t)/t$ 总体上会随着迭代次数 t 的增加而增大，因此 ω 总的趋势是随着迭代次数的增加而减小，从而加快了算法的收敛速度，但这样有可能收敛到局部最优解，因此用 $f_i(t)$ 根据当前粒子 i 自身情况来适当微调 ω 的值：如果 $f_i(t)$ 较小，说明粒子 i 与最优解接近，则 ω 会适当减小以缩短搜索步长，从而帮助找到更精确的解；如果 $f_i(t)$ 较大，说明粒子 i 与最优解相隔较远，则 ω 会适当增大以加大搜索步长，从而有利于寻找全局最优解，避免出现早熟。同时，公式中加入了限定因子 k，较小的 k 值会使 ω 急剧减小，较大的 k 值，会使 ω 缓慢减小，因此 k 可以用来限定减小惯性权重 ω 的快慢程度，好的 k 值更有利于加快收敛速度且避免出现早熟。

3.2.2　粒子群算法并行化

在基本 PSO 算法中，任何一次迭代循环中所有的粒子都以上次循环中确定的整体认知水平 P 进行搜索，即便是本次循环中出现了更好的位置点，可以将这种实现称为同步模式(Synchronous Pattern)[5]。PSO 的同步模式并不十分合理，因为 PSO 来自生物行为模拟，正如鸟的觅食，每只鸟都是独立的个体，它的每一次觅食动作都不会等待所有的鸟进行了一次觅食动作后再做反应的。实际上，在鸟群协同觅食的每一个时刻，任何一只鸟若是先发现了好的食物位置，就可以通过鸣叫等形式立即通知所有鸟，使之及时成为种群的整体认知。因此，若是考虑有着这样一种 “鲜明个体独立行为”和“即时种群通信行为”的寻优模式，效果会是怎样呢？

并行化问题的分解通常有两种形式，消息传递并行性的开发也有两种形式[6]：①域分解形式，即将一个大的问题区域分解成若干个较小的问题区域，然后对其并行求解；②功能分解形式，即将一个大的问题分解成若干个子问题，然后对其并行求解。由于 PSO 算法规模大小不是很敏感，因此本文算法采用第一种方式，具体实现是采用多线程技术，将所有粒子均匀地分配到各个线程(最后一个线程的粒子数为剩余未分配的粒子数目)，每个线程在迭代过程中发现当前粒子适应度比全局最优适应度好则进行更新操作，而不像串行粒子群算法要等到一轮计算结束后再更新。采用这种办法的好处是，其它粒子能更快地获得最新信息，这更加符合鸟在捕食时的情形。

4　基于自适应惯性权重的并行粒子群聚类算法

4.1　编码方案与适应度的选择

本文算法采用实数编码，每个粒子由 K 个聚类中心组成，粒子除了位置之外，还有速度和适应度 $f(x)$，而每个粒子的位置和速度都是 $K\times D$ 维变量，故粒子编码结构[7]如下：

$Z_{11}Z_{12}\cdots Z_{1d}Z_{21}Z_{22}\cdots Z_{2d}\cdots Z_{m1}Z_{m2}\cdots Z_{md}$	$V_{11}V_{12}\cdots V_{1d}V_{21}V_{22}\cdots V_{2d}\cdots V_{m1}V_{m2}\cdots V_{md}$	$f(X)$

由公式(1)知，J 越小说明聚类效果越好，反之越差，因此对于每个粒子，可以定义其适应度函数如下：

$$f(x)=\frac{M}{J} \quad (5)$$

其中，M 为大于 0 的常数。

4.2　基于自适应惯性权重的并行粒子群聚类算法描述

结合聚类算法和自适应惯性权重的并行粒子群算法，得到自适应惯性权重的并行粒子群聚类算法流程如下：

(1)初始化算法参数：c_1，c_2，$V_{\max}$，MaxGen(最大迭代次数)，CurGen=1(当前迭代次数)，设惯性权重 $\omega\in[0.4,0.9]$；

(2)按上述编码方案对所有粒子编码，并随机初始化其位置和速度；

(3)分配粒子到各个线程；

(4)各线程按公式(5)并行计算各粒子适应度;

(5)各线程并行更新粒子的个体最优值和以原子操作方式更新种群最优解;

(6)各线程粒子惯性权重 ω 按公式(4)调整;

(7)各线程并行按公式(2)更新粒子速度;

(8)设置互斥函数,让线程同步;

(9)按公式(3)更新所有粒子的位置;

(10)是否满足中止条件,如果是,算法中止;否则,转(3)。

5 实验结果与分析

本文中提到的各聚类算法采用标准 C++编程实现,考虑到移植性,多线程采用 pthread 包,算法程序在 VC 和 GCC 编译器上通过。数据由 matlab 中正态分布随机函数产生,通过大量的实验对算法有效性进行了验证。这里列举其中一组 600 个 2 维的数据,此数据分成三类,期望值分别是(4,8)、(6,6)、(7,5)如表 1 所示。

表 1 每种算法在同一机器上运行 10 次得到的比较结果

算法名称	最大迭代次数	实际平均迭代次数	最小适应度	最大适应度	平均适应度
K-means	100	3	982.119	1 334.48	1 072.64
GA	100	67	1 313.29	1 410.95	1 371.56
基本 PSO	100	45	1 415.90	1 422.11	1 420.10
本文算法	100	26	1 422.01	1 422.11	1 422.09

GA 算法, 基本 PSO 算法和本文算法中群体规模均为 30,适应度函数(公式(6))中的 M=1 000 000,聚类中心数为 3。以下为各算法的参数设置:

GA 算法:交叉概率 pc=0.9,变异概率 pm=0.08

基本 PSO 算法:ω=0.8,c_1=1.5,c_2=1.5

本文算法:$\omega \in [0.4,0.9]$(即 a=0.4,b=0.9),k=10,c_1=1.5,c_2=1.5,多线程数为 3

从实验结果可以看出,K-means 算法收敛速度最快, 但容易陷入局部最优解。GA 算法能得到大致近似最优解,但精度不高,并且需要迭代的次数多,同时也易陷入局部最优解。基本 PSO 算法得到的解精度较高,但需要的迭代次数也比较多。而本文算法得到的解精确度高, 迭代次数比基本 PSO 算法少很多,且每次结果类似,均非常接近最优解,没有出现收敛到局部最优的情况。

6 结语

本文对聚类算法中的几种智能算法进行了对比分析,并提出了一种自适应惯性权重的并行粒子群聚类算法,理论和实验证明了该算法在收敛精度、收敛速度和收敛的稳定性方面均取得了很好的效果。(收稿日期:2007 年 1 月)

参考文献:

[1] Han Jiawei,Micheline Kamber.数据挖掘概念与技术[M].北京:机械工业出版社,2005.

[2] Ansari N,Hou E.用于最优化的计算智能[M].李军,边肇祺,译.北京:清华大学出版社,1999.

[3] Kennedy j,Eberhart.Particle swarm optimization [C]//Proc IEEE Inr Conf on Neural Networks Perth,Australia,1995:1942-1948.

[4] Reynolds C W.Flocks,Herds and Schools:A Distrihuted Behavioral Model[J].Computer Graphic,1987,21(4):25-34.

[5] 李建勇,俞欢军.基于 Java 多线程技术实现的粒子群优化算法[J].计算机工程,2004,30(22):134-136.

[6] 陈国良.并行计算——结构、算法、编程[M].修订版.北京:高等教育出版社,2003.

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

(上接 133 页)

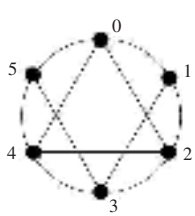

图 4 DLN(6;1,2)

证明 同样假设 $r \leqslant q$,通过观察发现,矩阵 $\boldsymbol{P}$ 共有 r 列数据块,分别分布在 r 个独立的环上。所以每一列的通信分别在 r 个独立的环上进行,仅需要 1 个波长。而行的通信要在[+1]边构成的环上进行,分析同环,在每个分组中的链路上均需要 2 个波长,分组间的链路则不需要波长。所以,按照算法 MRDR,在 WDM 双环网上实现并行矩阵乘的通信模式所需的最小波长数为 2。

5 结论

在光互连网络的各种拓扑结构中, 线性阵列、环、二维 mesh 和双环网都是重要的拓扑结构, 其上的波长分配问题具有重要的实用价值。本文讨论了在线性阵列、环、二维 mesh 和双环网上实现并行矩阵乘通信模式的波长分配问题,并给出了所需波长数。不同的并行算法具有不同的通信模式,光互连网络具有多种拓扑结构,进一步分析不同类型的通信模式在各种光互连网络拓扑结构上的波长分配问题,对于推动基于光互连的并行体系结构的发展具有较大的理论和实用价值。
(收稿日期:2006 年 10 月)

参考文献:

[1] 吴建平,迟学斌.分布式系统上并行矩阵乘法[J].计算数学,1999,21(1):99-108.

[2] 陈晶,黄曙光.分布式并行矩阵乘算法分析[J].兵工自动化,2005,24(5):52-54.

[3] 刘方爱,刘志勇,乔香珍.光 RP(k)网络上 Hypercube 通信模式的波长指派算法[J].软件学报,2003,14(3):575-581.

[4] 陈亚文,刘方爱.并行 FFT 的通信模式在一组规则 WDM 光互连网络上的波长分配[J].计算机研究与发展,2005,42(7):1231-1234.

[5] 陈亚文,刘方爱,张海波.并行 LU 分解的通信模式在 WDM 环网上的波长分配算法[J].小型微型计算机系统,2005,26(3):404-408.

[6] 陈亚文,刘方爱.在简单 WDM 光网络上实现 Hopfield 网络的波长分配算法[J].计算机工程,2005,31(3):131-133.

[7] Yuan X,Melhem R.Optimal routing and channel assignments for hypercube communication on optical mesh-like processor arrays[C]// Proceedings of the 5th International Conference on Massively Parallel Processing Using Optical Interconnection.Las Vegas,NV:IEEE Press,1998:110-118.