一种动态调整的蚁群聚类算法

贾瑞玉,邢 猛,徐庆鹏,黄义堂
(安徽大学 计算机科学与技术学院,安徽 合肥 230039)

摘 要:蚁群算法是优化领域中新出现的一种仿生进化算法,基于蚁群算法的聚类算法已经在当前的数据挖掘研究中得到应用。文中针对早期蚁群聚类算法的缺点,提出动态调整的蚁群聚类算法,通过加入运动速度不同的蚁群、半径自适应调整、短期记忆、强行放下等策略,来指导蚁群的移动行为,降低蚁群移动的随意性,减少了蚂蚁的搜索时间,提高聚类性能。仿真实验表明:改进算法能有效地提高算法效率且取得较好的聚类结果。

关键词:蚁群算法;运动速度不同的蚁群;半径的自适应调整;短期记忆
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2009)02-0145-03

A Dynamic Adjustive Ant Colony Clustering Algorithm

JIA Rui-yu,XING Meng,XU Qing-peng,HUANG Yi-tang
(School of Computer Science and Technology,Anhui University,Hefei 230039,China)

Abstract:Ant colony algorithm is a novel category of bionic algorithm,the ant-based clustering algorithm has currently applications in the data mining community. Based the disadvantage of the classical algorithm,presents a dynamic adjustive ant-clustering algorithm, including different speed ant colony, adaptive radius adjustment, short memory, force drop action which guide the ant's movement. The algorithm can lower the randomness of ant's moving and reduce the time of ant's searching to improve the performance. Experiment shows that the new algorithm effectively advances the efficiency of algorithm and the result of clustering.

Key words:ant colony algorithm;different speed ant colony;adaptive radius adjustment;short memory

0 引 言

聚类分析是指按不同对象之间的差异,根据特定的准则进行模式分类,使得每组内部的数据尽可能相似而不同组之间的数据尽可能不同,从而发现数据集内在的结构。近年来聚类分析在科学数据探测、图像处理、模式识别、计算生物学、文档检索以及Web分析等领域起着非常重要的作用,它已经成为当前数据挖掘领域中一个非常活跃的研究课题[1]。

蚁群算法是对蚂蚁采集食物、构筑巢穴等过程的模拟,具有分布式、自组织、信息素通信、合作等性能,已经在组合优化、通信网络、机器人以及Web文档聚类等领域取得成果[2]。1991年Deneubourg J L等基于蚁群聚类现象建立了一种基本蚁群聚类模型[3],后来Lumer E,Faieta B等修改了这个算法并将之应用于数据分析[4]。目前用于聚类分析的蚁群算法主要分为两类:一类是灵感源于蚂蚁觅食的蚁群路由选择算法[5];另一类是灵感源于Lumer E,Faieta B等提出的基于蚂蚁堆积尸体和幼体的LF蚁群聚类算法及其改进蚁群聚类算法[6~8]。这些算法具有许多优点,如自治性(聚类不再是根据所要求的对数据进行原始分割和分类,而是通过蚁群搜索行为自然地形成)、灵活性(为了避免局部最优不再采用决定性搜索,而是采用随机搜索)、并行性(代理操作是固有的并行),但仍存在显而易见的缺陷,如算法具有较高的误分类错误率、收敛性差和较长的时间花费。在此基础上,文中提出了一种动态调整的蚁群聚类算法来弥补这些缺陷。

1 基本蚁群聚类算法

Deneubourg J L等提出基于蚁群聚类现象建立了一种基本蚁堆模型,基本机制是工蚁堆积蚂蚁尸体过程,小蚁堆不断吸引工蚁堆积更多的死蚂蚁,通过正反馈导致蚁堆逐渐增大。后来,Lumer E和Faieta B改进了基本模型,提出LF蚁群聚类算法,其基本思想是人工蚂蚁沿着网格单元移动,每个单元只含一个对象,没有搬运数据对象的蚂蚁碰到对象时就会以某个概率 p_p 拾起它,这个概率依赖对该对象周围的相同对象密

收稿日期:2008-05-16
基金项目:安徽省自然科学基金项目(KJ2008B092)
作者简介:贾瑞玉(1965-),女,副教授,硕士生导师,研究方向为计算机图形学、数据挖掘、人工智能。

度 $f(o_i)$ 的评估，p_p 和 $f(o_i)$ 计算公式如下：

$$p_p(o_i) = [\frac{k_p}{k_p + f(o_i)}]^2 \quad (1)$$

$$f(o_i) = \max\{0, \frac{1}{s^2}\sum_{o_j \in s\times s}[1 - \frac{d(o_i,o_j)}{\partial}]\} \quad (2)$$

如果密度高则拾起的概率就低；携带对象的蚂蚁遇到空单元或搬运的对象与邻近的对象相似时就会以某个概率 p_d 放下它，放下的概率也依赖对周围对象类型密度的评估，p_d 的计算公式如下：

$$p_d(o_i) = \begin{cases} 2\ f(o_i) & \text{if } f(o_i) < k_2 \\ 1 & \text{if } f(o_i) > k_2 \end{cases} \quad (3)$$

如果密度大时放下的概率就高，结果相同类型的对象都被聚集在一起。

其中 $f(o_i)$ 为相似度密度，$o_j \in s \times s$ 表示物体 o_i 的 $s \times s$ 的邻域，∂ 为相异度因子。k_p 和 k_d 都是阈值常量。在研究过程中发现此算法还是存在聚类速度慢、聚类结果不理想等缺陷，文中在此基础上进行改进，通过半径自适应调整、短期记忆等策略，提高聚类性能。

2 动态调整的蚁群聚类算法

2.1 改进的密度评价函数

蚂蚁的运动速度影响聚类的效果。运动速度快的蚂蚁能很快将对象粗略地分为大的几类，而运动速度慢的蚂蚁能更精确地细分对象。利用 Lumer E 等在文献中提到的设置运动速度不同的蚂蚁来改进算法，速度 v 设置方法为：蚂蚁的速度为一个范围，从 1 到 $v$max 的随机数[9]。则区域密度评价函数计算公式如下：

$$f(o_i) = \max\left\{0, \frac{1}{s^2}\sum_{o_j \in s\times s}[1 - \frac{d(o_i,o_j)}{\partial(1 + (v-1)/v_{\max})}]\right\} \quad (4)$$

其中 v 表示蚂蚁运动速度，$v_{\max}$ 是蚂蚁最大运动速度。

2.2 半径的自适应调整

通过实验发现算法在迭代初期，邻域半径 $r(r = (s-1)/2)$ 越小，越有利于数据的聚类。但如果半径保持不变，聚类的速度会明显地减慢，随着迭代次数的增加，形成许多小的聚类而影响了进一步合并，导致聚类性能的下降。如果 r 过大，同样会降低初始化的效率。为防止这种情况的发生，采用 André L. Vizine 在文献[10]中提到的领域半径自适应调整策略，每个蚂蚁都具有一个参数 s_i^2，它可以动态和各自独立的调整，当每个蚂蚁遇到一个比较“大”的类时，则开始增加自己的 s_i^2。计算公式如下：

If $f(o_i) > \theta$ and $s_i^2 < s_{\max}^2$

Then $s_i^2 = s_i^2 + n_s$ (5)

直到不满足条件为止。本实验把参数做如下设定：$s_{\max}^2 = 8 \times 8, \theta = 0.8, n_s$ 为一个单位半径的领域。

2.3 短期记忆

针对数据是由蚂蚁随机选择的，这样就不能充分利用已有信息，影响算法效率，利用 Lume E 等在文献中提到的改进思路，利用短期记忆的思想，让每个蚂蚁记住它们最近几次放下数据的位置。当新的数据被拾起后，分别计算该数据与各记忆位置周围的 $s \times s$ 区域中的密度 $f(o_i)$，选择其中最大值所在的位置为“最佳匹配”位置，然后蚂蚁直接跳到该位置（如果该位置已有数据则选择最近的空闲区域），随后计算在这个位置区域中的 p_d 来决定是否放下该数据，如果 $p_d > p_r$（随机概率）则放下该物体并更新记忆，否则，记忆失效，蚂蚁随机选择一个位置，当放下成功后更新记忆。

2.4 强行放下

经过多次仿真实验，发现当一些较小聚类形成后，负载的蚂蚁多次寻找放下位置，如果出现 $f(o_i)$ 过小，则拾起的数据不能放下，这些蚂蚁在较长迭代过程中没有起到聚类的作用。为了尽量防止这种情况的出现，让蚂蚁在放下数据失败次数达到一定次数后（实验中取 80 次），强行放下该数据，以便加快聚类的进程。

动态调整的蚁群聚类算法描述如下：

(1)初始化蚁群中蚂蚁个数 ant_ number，短期记忆个数 ant_ memory，最大迭代次数 M，参数 $\partial, k_p, k_d, \theta$ 等；

(2)将数据对象随机投影到一个平面，及给每个对象随机地分配一些坐标值 $(x;y)$；

(3)每个蚂蚁初始化未负载，并初始化其相关参数 $v, v_{\max}, s_i^2, s_{\max}^2$ 等，并随机地选择一个空闲对象；

(4)for $i = 1,2,\cdots,M$

for $j = 1,2,\cdots,$ ant_ number

(4.1)根据公式(4)计算对象的平均相似性，根据公式(5)进行半径的自适应调整，直到不满足条件为止；

(4.2)如果蚂蚁未负载，根据公式(1)计算拾起概率 p_p。若 $p_p > p_r$，且该对象未被其它蚂蚁拾起，则蚂蚁拾起该对象。检查蚂蚁短期记忆，寻找最佳匹配位置，如果查找成功则把对象捡起移到该位置，并标记自己负载；如果查找不成功，则随机移到别处，并标记自己负载；若 $p_p < p_r$，蚂蚁拒绝拾起该物体，并随机选择其他空闲对象。最后初始化 fail_ number；

(4.3)如果蚂蚁为负载状态，若 fail_ number 小于阈值，根据公式(3)计算放下概率 p_d，若 $p_d > p_r$，则蚂蚁放下该物体，标记自己未负载，更新记忆表，并随机选择其他空闲对象；否则蚂蚁根据移动规则移动该对象到新的空闲位置，更新 fail_ number 和记忆表；若 fail_

number大于设定阈值，强行放下该对象，重置 fail_number；

(5)for $i = 1,2,\cdots,$data_ number

(5.1) 如果一个对象是孤立的或它的邻域对象 b 个数小于某一个常数，则标记该对象为孤立点；

(5.2) 否则给该对象分配一个聚类序列号，并递归地将其邻域对象标记为同样的序列号。

3　实验结果及分析

实验采用 UCI 机器学习仓库的数据集：Iris(150,4,3)和 Wine(178,13,3)进行测试，实验采用了三个性能评价指标：聚类数目(C_n)，误分类错误率(C_e)和总的运行时间(C_t(s))。算法性能评价指标比较(见表 1)给出了 k－均值算法，LF 蚁群聚类算法，动态调整的蚁群聚类算法在两个数据集上的三个性能评价指标的均值(50 次实验)。算法迭代次数统计比较(见表 2)给出了 LF 蚁群聚类算法和动态调整的蚁群聚类算法在迭代次数上的统计结果。

表 1　算法性能评价指标比较

数据库	评价指标	算法		
		k－均值	LF 蚁群	动态蚁群
Iris	C－n	3	3	3
	C－e	11.2%	4.5%	3.8%
	C－t(s)	0.03	58.2	8.5
Wine	C－n	3	3	3
	C－e	30.2%	4.6%	3.1%
	C－t(s)	0.03	62.8	9.3

表 2　算法迭代次数统计比较

算法	迭代次数
LF 蚁群	100000
动态蚁群	5000

表 1 显示的是对两个数据集用 k－均值算法、LF 蚁群聚类算法和动态蚁群聚类算法进行测试结果的比较，对于 k－均值算法，在实验中使用的 k 值假定已知，可以明显看出：LF 蚁群聚类算法和动态蚁群聚类算法除了时间开销高于 k－均值算法外，在聚类的性能上要远优于 k－均值算法。这是因为 k－均值算法必须预知类的个数，否则无法进行，因此为它预先设定了类的个数，使得它的聚类速度较快。而蚁群聚类算法要在聚类过程中探索聚类的个数，这需要花费大量时间。此外，在进行一定迭代次数后，k－均值算法很快收敛，无法继续进行，所以得到的解的质量较差。

表 1 和表 2 显示，动态蚁群聚类算法在时间开销上远优于 LF 蚁群聚类算法，且性能也要比 LF 蚁群聚类算法性能好。此外，动态蚁群聚类算法只需要 5000 次迭代就可以达到 LF 蚁群聚类算法经过 100000 次迭代的效果，因此，LF 蚁群聚类算法所需要的时间代价较大。这是因为在 LF 蚁群聚类算法中蚂蚁在捡起数据、放下数据的过程中，大量的时间花费在寻找数据上；另一方面，参数缺少自适应的变化，也使得聚类的效果在短时间内不明显。但是，动态蚁群聚类算法也有不完善的地方，如：算法的性能对参数的设置仍然很敏感，有时对算法效率和聚类结果影响较大。

4　结束语

文中提出了一种动态调整的蚁群聚类算法，该算法结合 LF 蚁群聚类算法的优点，加入运动速度不同的蚂蚁，短期记忆，半径的自适应调整等策略，克服了基本蚁群聚类算法收敛速度慢，聚类效果不理想的缺点。实验结果证明该算法提高了算法收敛速度，改善了聚类质量，是一个可靠和有效的算法。尽管如此，该算法存在一些缺陷，众多参数的设置对算法的有效性影响比较大，需要进一步完善这方面的不足。

参考文献：

[1] Han Jiawei, Kamber M. Data Mining: Concepts and Techniques[M]. 北京：机械工业出版社，2001.

[2] 段海滨. 蚁群算法理论及其应用[M]. 北京：科学出版社，2005.

[3] Deneubourg J L, Goss S, Franks N, et al. The Dynamics of Collective Sorting: Robot－Like Ants and Ant－Like Robots [C]//In: From Animals to Animates: Proceedings of the First International Conference On Simulation of Adaptive Behavior. [s.l.]: MIT Press, 1991: 356－363.

[4] Lumer E, Faieta B. Diversity and adaptation in population of clustering ants[C]//Proc of the 3rd International Conference on Simulation of Adaptive Behavior: From Animals to Animates. Cambridge: MIT Press, 1994: 501－508.

[5] 杨欣斌，孙京浩，黄　道. 一种进化聚类学习新方法[J]. 计算机工程与应用，2003，39(3)：60－62.

[6] 赵伟丽，孙艳蕊. 基于信息熵的蚁群聚类算法的改进[J]. 沈阳化工学院学报，2005，19(4)：296－300.

[7] 刘念涛，刘希玉. 基于改进的启发式蚁群算法的聚类问题的研究[J]. 计算机技术与发展，2007，17(8)：37－39.

[8] 徐晓华，陈　崚. 一种自适应的蚂蚁聚类算法[J]. 软件学报，2006，17(9)：1184－1189.

[9] 杨　燕，靳　蕃，Kamel M. 一种基于蚁群算法的聚类组合方法[J]. 铁道学报，2004，26(4)：64－69.

[10] Vizinel A L, Leandro N, Eduardo R, et al. Towards Improving Clustering Ants: An Adaptive Ant Clustering Algorithm [J]. Informatics, 2005(29): 143－154.