ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.9, September 2006, pp.1884−1889
DOI: 10.1360/jos171884

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种自适应的蚂蚁聚类算法*

徐晓华[1], 陈 崚[2,3+]

[1](南京航空航天大学 信息科学与技术学院,江苏 南京 210016)

[2](扬州大学 计算机科学与工程系,江苏 扬州 225009)

[3](计算机软件新技术国家重点实验室(南京大学),江苏 南京 210093)

An Adaptive Ant Clustering Algorithm

XU Xiao-Hua[1], CHEN Ling[2,3+]

[1](College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

[2](Department of Computer Science and Engineering, Yangzhou University, Yangzhou 225009, China)

[3](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210093, China)

+ Corresponding author: Phn: +86-514-7978307, E-mail: lchen@yzcn.net, http://www.yzu.edu.cn/xyweb/xxxy/xykk/xrld_cl.htm

Xu XH, Chen L. An adaptive ant clustering algorithm. *Journal of Software*, 2006,17(9):1884−1889.
http://www.jos.org.cn/1000-9825/17/1884.htm

Abstract: Enlightened by the behaviors of gregarious ant colonies, an artificial ant movement (AM) model and an adaptive ant clustering (AAC) algorithm for this model are presented. In the algorithm, each ant is treated as an agent to represent a data object. In the AM model, each ant has two states: sleeping state and active state. In the algorithm AAC, the ant's state is controlled by both a function of the ant's fitness to the environment it locates and a probability function for the ants becoming active. By moving dynamically, the ants form different subgroups adaptively, and consequently the whole ant group dynamically self-organizes into distinctive and independent subgroups within which highly similar ants are closely connected. The result of data objects clustering is therefore achieved. This paper also present a method to adaptively update the parameters and the ants' local movement strategies which greatly improve the speed and the quality of clustering. Experimental results show that the AAC algorithm on the AM model is much superior to other ant clustering methods such as BM and LF in terms of computational cost, speed and quality. It is adaptive, robust and efficient, and achieves high autonomy, simplicity and efficiency. It is suitable for solving high dimensional and complicated clustering problems.

Key words: swarm intelligence; ant clustering

摘 要: 受蚂蚁分巢居住行为的启发,提出一种人工蚂蚁运动(ant movement,简称 AM)模型和在此模型上的一个自适应的蚂蚁聚类算法(adaptive ant clustering,简称 AAC).将人工蚂蚁看成一个行为简单的 Agent,代表一个数

* Supported by the National Natural Science Foundation of China under Grant No.60074013 (国家自然科学基金); the Chinese National Foundation for Science and Technology Development under Grant No.2003BA614A-14 (国家科技攻关项目); the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2005047 (江苏省自然科学基金); the Open Foundation of State Key Laboratory for Novel Software Technology (Nanjing University) of China (计算机软件新技术国家重点实验室(南京大学)开放基金)

Received 2004-06-13; Accepted 2005-08-29

据对象.在AM中,人工蚂蚁有睡眠和活跃两种状态.在AAC算法中,定义了一个适应度函数用来衡量蚂蚁与其邻居的相似程度.人工蚂蚁通过其适应度和激活概率函数来决定处于活跃态或者睡眠态.整个蚂蚁群体在移动中动态地、自适应地、自组织地形成多个独立的子群体,使不同类别的蚂蚁之间相互分离;而同类的蚂蚁之间高度紧密地排列,从而形成聚类.提出了对参数的自适应的更新方法,使得人工蚂蚁的移动仅仅使用少量的局部信息,这对加快聚类速度和提高聚类质量有非常显著的效果.模拟实验充分显示出,该蚂蚁聚类算法与BM和LF算法相比,在模型上更直观,操作上更简单,可自适应地修改参数,对参数的限制少,计算成本较小,聚类质量高,具有速度快、高效、自组织性和鲁棒性的优点,适用于解决高维、复杂的聚类问题.

关键词: 群体智能;蚁群聚类
中图法分类号: TP301 文献标识码: A

科学家们发现,自然界中的蚂蚁群体能够完成单个蚂蚁所无法胜任的工作.蚂蚁群体的这种能力与它们的协同作用有关.协同作用是指由多个个体或部分合作产生的“整体大于局部之和”的效果.人们受蚂蚁等社会性昆虫的繁衍后代、寻找食物、构造巢穴、清除垃圾、保卫领地等行为的启发,已经设计了一系列蚁群算法,并成功地应用于组合优化[1−4]、网络路由[5]、机器人技术[6]等领域.Grassé等人用术语stigmergy[7]来解释蚂蚁通过改变周围环境来进行间接交流的形式.一些研究人员模拟含有stigmergy的人工蚂蚁的群体智能来处理数据挖掘中的问题,已经取得了一定成效.Deneubourg[7]等人提出了一种基本模型(basic model,简称BM)用来解释蚂蚁尸体堆积成蚂蚁墓的行为.Lumer和Faieta[7]扩展了BM模型,给出了数据对象的相似性度量表达式,设计了用于数据聚类的LF算法.Ramos等人[8]和Handl等人[9]又将LF算法应用于文本的聚类,取得了很好的效果.Holland等人[9]则将它们运用到机器人领域.吴斌等人[10]将群体智能聚类模型运用于Web文档聚类,取得了满意的效果.但是,BM和LF算法在处理聚类问题时,存在着比较难以解决的问题:一方面,人工蚂蚁在捡起一个数据对象或丢下一个数据对象之前,都是在做大量、无效的随机移动,有一些位置可能重复多次达到,从而使得算法的时间成本较高,要形成高质量的聚类则需要很长的时间;另一方面,BM和LF算法中对参数设置具有敏感性,特别是关键参数α的确定,非常依赖于使用者的经验,使得聚类缺少鲁棒性,聚类的效果受到影响.虽然Handl等人[9]已经做了一些提高LF算法性能方面的工作,但基于蚂蚁尸体堆积模型的聚类效果还有待改善.鉴于BM的时间消耗过大的弱点,本文提出了一种人工蚂蚁聚类模型,即蚂蚁运动模型(ant movement,简称AM)和在此模型上的一个自适应的蚂蚁聚类算法(adaptive ant clustering,简称AAC).AM模型通过模拟蚂蚁因为安全需求而产生聚集的行为,利用一个人工蚂蚁代表一个数据.人工蚂蚁遵循我们提出的激活概率函数和聚类规则而不停地寻找合适位置,从而使得蚂蚁群体动态自组织地形成聚类.我们的实验证明:AM比BM更直观,操作更简单,它可以自适应地修改参数,对参数的限制少、计算成本小、聚类质量高,具有快速、高效、自组织等优点.我们的方法对于解决高维的、复杂的数据聚类问题也十分有效.

1 蚂蚁运动模型

生物学家在考察蚂蚁群居住的巢穴时发现,蚂蚁的巢穴是相互靠近的,这样蚂蚁之间可以互相照应,共同抵御外来者的入侵.但蚂蚁的巢穴之间不是均匀分布的,而是分群而居,各方面习性比较相近的蚂蚁会居住得比较近;反之,习性相对较远的就居住得比较远.即使是在同一群蚂蚁内部,也存在着蚂蚁之间关系的亲疏差别.蚂蚁在栖息的时候,总是倾向于跟同类或者习性相近的在一起.即使在同一蚂蚁群体内部,蚂蚁更喜欢与熟悉的伙伴作邻居.蚂蚁因为安全性的需求,总是会选择对它来说更有利、更舒适的环境来栖息.因为单个蚂蚁的能力非常微小,它需要与同伴相互协作才能满足自身对安全性的需求.所以,对于像蚂蚁这样弱小的个体,为了安全性的考虑,很自然地便会产生同类相聚、异类相斥的行为.

AM模拟蚂蚁在生存环境中“寻找一个舒适的位置来休息”的行为,其中的每个人工蚂蚁表示一个简单的Agent.它的行为很简单:当它没有找到适合的位置休息时,就继续移动去寻找,而当它找到了相对舒适的位置就停在原地;当周围环境的改变使得它不满意当前位置时,它又活跃起来,开始新的移动以寻找新的合适的位置.

如此重复,直到所有蚂蚁都找到合适的位置.AM 不同于 BM:在 BM 中,人工蚂蚁承担着搬运数据对象的任务;而在 AM 中,一个 Agent 代表一个数据对象,这更加接近自然蚂蚁的行为.自然界中的蚂蚁之所以能够归类而居,是因为蚂蚁有"物以类聚,人以群分"的智能,而不是因为蚂蚁有对物体进行分类的能力.

在 AM 中,蚂蚁活动的空间是一个[0..$w(n)$−1]×[0..$h(n)$−1]的二维网格.这里,$w(n)=h(n)=ceil(n^{0.5})$,其中 *ceil* 为上取整函数.网格的上、下边界相连,左、右边界相连.在 BM 及 LF 模型中,网格是一般的长方形;而这里选用的是拓扑等价于球面的网格,这样可以使得 Agent 生存环境中,每个位置的邻域都是相似的,避免有中心与边界、角落的差异,给 Agent 的移动和计算带来方便.我们将 $agent_i$ 的状态记为 $q_i,q_i=(x_i,y_i,c_i,s_i)(1\le i\le n)$,这里,$n$ 为数据个数;x_i 和 y_i 为 $agent_i$ 所在位置的坐标;c_i 表示它所在类的类号;s_i 反映它当前是处于活跃还是睡眠状态.我们使用 Moore 型邻域,使得蚂蚁在网格上的任何一个位置的周围都有 8 个邻居,并记 $N(agent_i)$为 $agent_i$ 的邻域.定义 $agent_i$ 在当前位置的适应度函数 $f(agent_i)$为

$$f(agent_i)=\max\left\{0,\frac{1}{9}\sum_{agent_j\in N(agent_i)}\left(1-\frac{d(agent_i,agent_j)\cdot d(agent_j,agent_i)}{\alpha_{ij}}\right)\right\} \tag{1}$$

这里,$d(agent_i,agent_j)$表示 Agent 所代表的数据对象 i 和数据对象 j 之间的距离,即相异度,通常使用欧几里德距离,它可以在聚类的预处理阶段被计算出来.参数α_{ij}的值由如下公式确定:

$$\alpha_{ij}=\left(\frac{1}{n}\sum_{k=1}^{n}d(agent_i,agent_k)\right)\cdot\left(\frac{1}{n}\sum_{k=1}^{n}d(agent_j,agent_k)\right) \tag{2}$$

蚂蚁在环境中转为活跃状态的激活概率 p_a 用来表示:

$$p_a(agent_i)=e^{-\beta\cdot f(agent_i)} \tag{3}$$

这里的参数$\beta\in R^+$,称为激活阈值.在 AM 中,用δ来表示聚类规则的集合,$agent_i$ 的类别信息 c_i 通过δ中以下的规则来更新:(a) 若 $agent_i$ 到达一个合适的位置而变为睡眠态,那么它的类号将改变,取它邻域中与其相异度最小的 Agent 的类号;(b) 若 $agent_i$ 由睡眠态变为活跃态,那么它的类号也将改变,以它的标号作为其在移动期间的类号;(c) 若 $agent_i$ 继续保持睡眠态不变,则其类号也将保持不变.

在 BM 中,蚂蚁和被聚类的数据分开,增加了处理对象,使用了较多的参数和信息.而且,因为被聚类的数据对象不能直接地、自主地运动,蚂蚁在未负载数据时的运动是无效的随机运动,由此会带来大量额外的信息存储和计算负担.在 BM 中,一方面存在着蚂蚁饥饿现象,即蚂蚁随机运动中找不到相应的负载,这会耗费大量的时间;另一方面存在着数据累赘现象,即当蚂蚁捡起的数据是孤立点时,它就很难找到恰当的位置将数据放下来,从而使得聚类过程处于停滞状态,这对于蚂蚁较少的情况尤为明显.以上这些问题极大地影响了基于 BM 的聚类速度和效果.而我们提出的 AM 中,用蚂蚁代表数据对象,减少了处理对象,降低了计算时间和存储空间的需求,避免了蚂蚁的无效移动,提高了聚类的速度和质量.AM 操作简单,无中心控制,Agent 只需要局部信息就可以更新其状态,Agent 通过相互协作,从而动态地形成聚类.AM 具有直观性,可以从可视的网格上获得直观的 Agent 聚类信息.此外,在算法中,聚类的结果由数据对象所在的类号给出,比 BM 更直观、更简单.

2 蚂蚁聚类算法及其参数设置

通过以上的定义和说明,AM 处理聚类问题的算法可描述如下:

算法. 自适应的蚂蚁聚类(AAC).

01. 初始化参数设置和数据预处理
02. foreach Agent do
03. 将 Agent 随机放置于网格的某个格点中,并置 Agent 的类号为其标号
04. end for
05. while (not termination)
06. foreach Agent do

07. 计算 Agent 的活跃概率 p_a(Agent)
08. 产生一个[0,1]区间中满足均匀分布的随机数 r
09. if $r \le p_a$(Agent) then
10. Agent 在 N(Agent)中选择一个位置,如果该位置空闲,则 Agent 移动到该位置
11. end if
12. 依照δ规则更新 Agent 类号
13. end for
14. 更新参数β值
15. end while
16. 输出所有 Agent 的聚类信息

参数β的初始值设为常数 10.在算法的运行过程中,为了提高聚类的质量,我们让β值自适应地变化,目的是快速地形成高质量的聚类.记第 t 代 Agent 的平均适应度为 $f_{avg}(t)$,它可用来衡量该时刻的聚类质量.在算法的第 14 行更新参数时,根据 Agent 的平均适应度的变化,相应地对 t 时刻的β值$\beta(t)$作自适应的调整,其增量$\Delta\beta(t)$为

$$\Delta\beta(t)=k\cdot\Delta f_{avg}(t) \tag{4}$$

这里,k 为常数,我们取 $k=-10$.为了避免过于频繁地更新β值,可以设定每间隔 10 次迭代更新一次参数β值.这样的设置在聚类的初始阶段,使β值相对较小,Agent 的激活概率就比较大,这样可以快速地形成粗糙的聚类轮廓;而在整体聚类质量提高时,β值相对变大,已经处于合适的类中的 Agent 的激活概率就比较小,这样有利于维持并形成质量较高的聚类.这样,通过 Agent 的移动策略和自适应的参数更新进行整体调节,使得本算法能够比较好地解决蚂蚁聚类算法中的收敛速度和聚类质量之间的矛盾,有效地提高了聚类的速度,改善了聚类的质量.

3 实验结果

我们选用的测试数据集是来自 UCI 机器学习库[11]的 Iris,Wine 和 Glass.测试数据集分别用 LF,AAC 和 k-means 算法进行测试.实验结果表明,对各个测试数据集,AAC 在第 5 000 代时的聚类结果都比 LF 算法 1 000 000 代时的要好.因而我们在每次测试中,对 LF 算法迭代到 1 000 000 代,而对 AAC 算法迭代到 5 000 代.LF 算法的参数设定:k_1=0.10,k_2=0.15[7],AAC 算法的参数可以由聚类数据直接定出.我们对每个测试数据集重复进行 100 次测试.

图 1 所示的是 Iris 数据集中 150 个数据的前两个属性组成的投影图.图 2 是 AAC 算法对 Iris 数据集聚类的结果.图中分别用符号 o,+,*表示 Setosa,Versicolor 和 Virginica 三个类别的数据.对 3 个数据集用 AAC 及 LF 进行测试的结果见表 1.

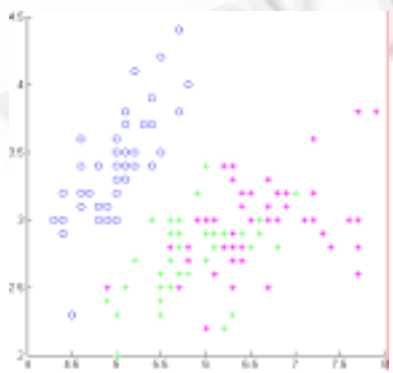

Fig.1 Distribution of the first attributes of Iris
图 1 Iris 数据按前两个属性的分布

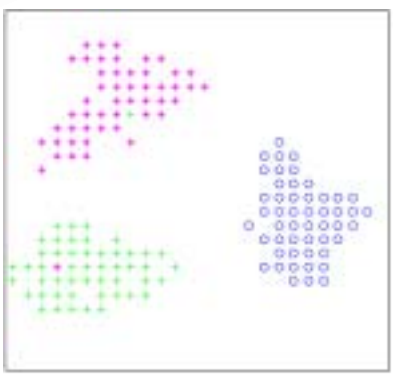

Fig.2 Results of clustering by AAC on Iris
图 2 AAC 的聚类的结果

Table 1 Test results of AAC and LF on Iris, Wine and Glass

表 1 AAC 与 LF 对 Iris,Wine 和 Glass 数据集合的测试结果

DataSet	Iris		Wine		Glass	
Algorithm	LF	AAC	LF	AAC	LF	AAC
Maximum iterations	1 000 000	5 000	1 000 000	5 000	1 000 000	5 000
Average running time (s)	56.81	1.36	62.37	1.44	106.21	2.37
Average errors	6.68	4.39	8.05	4.83	10.25	7.59
Percentage of the errors (%)	4.45	2.94	4.52	2.71	4.79	3.55

从实验结果可以看出,AAC 算法只需不超过 5 000 代就可以达到 LF 算法迭代 1 000 000 代的聚类效果.因此,LF 算法所需要的时间代价较大.这主要是因为蚂蚁在捡起数据、放下数据的过程中,大量的时间花费在寻找数据上.在聚类的质量上,一方面因为 LF 算法参数确定较为困难,特别是α值的敏感性会显著影响聚类效果;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.而 AAC 算法的聚类由 Agent 直接进行,可以加快聚类的速度.我们通过β值自适应地变化,可以快速地形成聚类,有效地改善了聚类的质量.

表 2 显示的是对 3 个数据集用 AAC 与 *k*-means 进行测试结果的比较,对于 *k*-means 算法,在实验中使用的 *k* 值我们假定已知.可以明显看出:AAC 除了时间开销高于 *k*-means 外,在聚类的性能上要远远优于 *k*-means 算法.由于 *k*-means 必须预知类的个数,否则无法进行,因此我们为它预先设定了类的正确个数,使得它的聚类速度较快;而 AAC 要在聚类过程中探索聚类的个数,这需要花费大量时间.在进行了一定的迭代次数后,*k*-means 算法很快收敛,无法继续进行,但所得到的解的质量较差.例如:*k*-means 对于数据集 Glass 的测试时不能将前 3 类分开,从而使得结果正确性较差,平均错误率大于 50%;而 AAC 的平均错误率仅有 3.55%,聚类质量远优于 *k*-means.

Table 2 Test results of AAC and *k*-means on Iris, Wine and Glass

表 2 AAC 与 *k*-means 对 Iris,Wine 和 Glass 数据集合的测试结果

DataSet	Iris		Wine		Glass	
Algorithm	AAC	*k*-means (*k*=3)	AAC	*k*-means (*k*=3)	AAC	*k*-means (*k*=6)
Average running time (s)	1.36	0.03	1.44	0.03	2.37	0.17
Average errors	4.39	16	4.83	53	7.59	≥107
Percentage of the errors (%)	2.94	10.67	2.71	29.78	3.55	≥50

4 结 论

本文提出了一种简单的蚂蚁运动模型 AM,并用 AM 有效地处理数据挖掘中的聚类问题.在 AM 中,用人工蚂蚁即一个 Agent 代表一个数据.Agent 在二维网格上移动,根据它对生存环境的适应度和激活概率来确定它的下一个要移动的位置.同时依照聚类规则集合δ,动态更新 Agent 的类号.Agent 的移动使得 Agent 与它的邻域内的邻居相互影响、相互作用,经过一定代数的迭代后,自组织地形成聚类.基于 AM 的自适应的人工蚂蚁聚类算法 AAC 无须中心控制,仅利用少量的局部邻域信息,就可以自组织地形成较好的聚类结果.我们给出了 AAC 中参数的自适应设置方法,使得本算法能够比较好地解决蚂蚁聚类算法中的收敛速度和聚类质量之间的矛盾.基于 AM 的自适应人工蚂蚁聚类算法 AAC 与 BM 和 LF 算法相比,模型上更直观,操作上更简单.由于自适应地修改参数,算法对参数取值的限制较少,使得计算成本减少,速度加快.实验的结果显示了 AM 对加快聚类有非常显著的效果,并且使得聚类的质量更高,它具有高效、自组织、自适应、动态可视等优点.实验表明,我们的方法对于解决高维的、复杂的数据聚类问题也十分有效.

致谢 美国伊利诺斯大学 Urbana-Champaign 计算机科学系的陈一昕给本文的研究工作提出了有益的建议,扬州大学计算机科学与工程系的屠莉协助我们为本文做了一些工作,在此表示感谢.

References:

[1] Bonabeau E, Dorigo M, Théralaz G. Swarm Intelligence: From Natural to Artificial Systems. Santa Fe Institute in the Sciences of the Complexity. New York: Oxford University Press, 1999.

[2] Dorigo M, Maniezzo V, Colorni A. Ant system: Optimization by a colony of cooperative learning approach to the traveling Agents. IEEE Trans. on Systems, Man, and Cybernetics, 1996,26(1):29−41.

[3] Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evolutionary Computation, 1997,1(1):53−66.

[4] Stutzle T, Hoos H. MAX-MIN ant systems. Future Generation Computer Systems, 2000,16(8):889−914.

[5] Di Caro G, Dorigo M. AntNet: A mobile agents approach for adaptive routing. Technical Report, IRIDIA, 1997. 97−12.

[6] Holland OE, Melhuish C. Stigmergy, self-organization, and sorting in collective robotics. Artificial Life, 1999,5(5):173−202.

[7] Dorigo M, Bonabeau E, Théraulaz G. Ant algorithms and stigmergy. Future Generation Computer Systems, 2000,16(8):851−871.

[8] Vitorino R, Juan JM. Self-Organized stigmergic document maps: Environment as a mechanism for context learning. In: Alba E, Herrera F, Merelo JJ, eds. Proc. of the 1st Int'l Conf. On Metaheuristics, Evolutionary and Bio-Inspired Algorithms. 2002. 284−293.

[9] Handl J, Meyer B. Improved ant-based clustering and sorting in a document retrieval interface. LNCS 2439, 2002. 913−923.

[10] Wu B, Zheng Y, Liu SH, Shi ZZ. CSIM: A document clustering algorithm based on swarm intelligence. In: Proc. of the 2002 Congress on Evolutionary Computation. IEEE Press, 2002. 477−482.

[11] Blake CL, Merz CJ. UCI Machine Learning repository of machine learning databases. 1998. http://www.ics.uci.edu/~mlearn/MLSummary.html

徐晓华(1979－),男,江苏通州人,博士生,主要研究领域为并行计算,计算分子生物学,数据挖掘.

陈崚(1951－),男,教授,博士生导师,主要研究领域为并行计算,人工智能.