基于离散点的蚁群聚类算法的研究*)

李 瑞[1,2] 邱玉辉[1]

(西南师范大学计算机与信息科学学院 重庆4007152)[1] (渝西学院数学与计算机科学系 重庆永川402168)[2]

摘 要 蚂蚁等群居式昆虫具有分布式、自组织、基于信息素间接通信(pheromone)等群体协作能力,模拟其智能行为的蚁群算法解决了许多复杂的问题并在并在数据聚类分析领域取得成效。本文首先介绍了基于蚂蚁的聚类算法的基本理论,讨论了参数 σ 对邻域平均相似度的影响并做了实验分析比较,然后提出利用离散点对算法进行改进,通过对离散点的检测算法能够对蚂蚁行为进行控制,使蚂蚁快速地决定下一个负载节点,从而有效地缩短聚类分析的执行时间。实验表明改进后的蚂蚁聚类算法具有较好的聚类特性,其收敛性也得到了有效改善。

关键词 聚类,蚁群算法,平均相似度,相似度刻度因子,离散点

Study of Ants-Clustering Algorithm Based on Outlier

LI Rui[1,2] QIU Yu-Hui[1]

(College of Computer and Information Science, Southwest Normal University, Chongqing 400715)[1]

(Department of Mathematics & Computer Science ,Western Chongqing University, Yongchuan Chongqing 402168)[2]

Abstract Social insects such as ants have the ability of collaboration due to the swarm intelligence of ants and the mechanisms of their distributed behavior, self-organization and pheromone communication. Ant-based clustering has been applied in a variety of areas, such as problems arising in commerce, circuit design data clustering analysis in data-mining community. In this paper, we first present the basic theory of ant-clustering algorithm, discuss the scaling parameter which effect on neighborhood function, and analyze the experiment result. We also propose an improved algorithm based on outlier. The improved algorithm can check outlier to control the action of ants and decide the next load node quickly, then shorten the executive time and speed the convergence. At last, we compare our algorithm with related work and improve its effective.

Keywords Clustering analysis,Ants-clustering algorithm,Neighborhood function,Scaling parameter,Outlier

1 引言

"物以类聚,人以群分",聚类分析的目的就是将数据库中的数据划分成具有一定意义的子类,使得不同子类中的数据尽可能不同,而同一子类中的数据尽可能相似。

已知模式样本集$[X]$有 n 个样本和 k 个模式聚类$[c_j, j=1,2,\cdots,k]$,以每个模式样本到聚类中心的距离之和达到最小值为准则,聚类的数学模型[1]为:

$$\min \sum_{j=1}^{k}\sum_{X\in c_j}\|X-m_j\| \tag{1}$$

$$m_j=\frac{1}{\sum_{i=1}^{n} y_{ij}}\sum_{i=1}^{n} y_{ij}X \tag{2}$$

其中 k 为聚类数目,m_j 为 j 类样本的均值向量。若模式样本 i 分配第 j 聚类中心,则令 $y_{ij}=1$,否则,令 $y_{ij}=0$,$\sum_{j=1}^{n} y_{ij}=1$表示模式样本 i 只能分配到一个聚类中心上。

聚类是一个 NP 复杂度问题,目前没有有效的算法解此问题。经典的聚类方法包括分层算法,划分方法如 k-means[2] 算法,图论聚类法,神经网络法,基于统计的方法以及基于蚁群算法的聚类算法。蚁群聚类算法中,蚂蚁通过一个正反馈机制使得小的聚类中蚂蚁的躯体发出更多的信息素吸引工蚁存放更多的同类对象而形成更大的聚类,各单个蚂蚁间以及蚂蚁与环境的交互作用实现完全是分布式控制,使算法具有自组织、可扩展性、健壮性等特性,更加适应海量数据的聚类分析操作。

Deneubourg[3]等人受到蚂蚁的墓地构建行为的启发开发出 BM 模型来解释蚁群聚类现象,Lumer 和 Faieta[4]首先提出 LF 算法对 BM 模型进行改进并用于数据分析。Julia Handle[5]采用多维尺度和蚂蚁聚类算法的结合动态的产生表示网路查询结果的拓扑图,针对在线查询时间限制和处理任意数据集的收集对算法做了一系列的改进。Kuntz,Layzell 和 Snyer[6]将算法用于图像的分割,修改了每对图像节点的非相似度值函数,并对算法做了类间相关系数和类内相关系数的测试比较。Monmarche[7]引入了 ANTCLASS 系统,采用蚁群聚类算法和 k-means 算法交替执行的混合算法,弥补了蚁群聚类算法和 k-means 算法的不足。吴斌、史忠植[8]、乔银锋[9]等对蚁群聚类算法的改进作出了一定的成绩。

蚁群聚类算法中寻找与领域不相似点的方法通常有两种:一种是将全部未负载的与领域不相似点用向量来表示,蚂蚁每次放下负载后都从向量中选取下一个负载节点;另一种是通过蚂蚁的随机移动找到拾起概率大的点。前者需要相当的存贮空间并且由于聚类环境的动态变化需要对向量中数据经常刷新;而后者由于蚂蚁移动的随机性使得寻找过程具有盲目性,占用大量的反复移动时间。

本文将讨论蚁群聚类算法的邻域相似度函数中参数 σ 对聚类分析过程产生的影响,通过邻域中的离散点试图改进算法,使算法能够快速确定蚂蚁的下一个负载点,改善蚂蚁行为的随机性,缩短蚂蚁寻找负载节点的时间,提高算法的收敛速度。

*)基金项目:重庆市自然科学基金(cstc. 2004BB2086)。李 瑞 硕士生,讲师,研究方向:人工智能、智能推荐系统;邱玉辉 教授,博导,研究方向:人工智能,模式识别。

文章第2节介绍蚁群聚类算法的基本理论;第3节介绍相似度函数中参数σ对聚类分析过程产生的影响,通过实验作出相应的结论;第4节描述了利用局域离散点对原有的蚁群聚类算法进行改进的主要思路以及算法改进前后的实验比较,最后是结论和今后的研究方向。

2 蚁群聚类算法基本理论

2.1 邻域相似度函数

邻域相似度函数 $f(i)$ 是蚂蚁拾起或放下对象 i 与观察半径内邻域中对象间的平均相似度。

$$f(o_i)=\begin{cases}\frac{1}{\sigma^2}\sum_j(1-\frac{\delta(i,j)}{\alpha}) & \text{if } f(o_i)>0\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

其中,$\delta(i,j)\subseteq[0,1]$表示对象 i 和对象 j 之间的距离,常用欧氏距离或余弦夹公式计算,$\alpha\subseteq[0,1]$为刻度因子,σ^2为对象 i 观察区域的面积或对象个数,蚂蚁位于邻居区域中心,$(\sigma-1)/2$为蚂蚁的邻居区域观察半径。

2.2 拾起概率,放下概率

聚类分析过程中,蚂蚁总是拾起与邻域节点最不相似的节点,然后将节点放到与邻域节点最为相似的位子中。拾起概率、放下概率是蚂蚁根据邻域相似度值的大小决定是否拾起或放下一个对象的概率。分别用 p_{pick}和 p_{drop}表示。

$$p_{pick(i)}=(\frac{k^+}{k^++f(i)})^2 \tag{4}$$

$$P_{drop(i)}=(\frac{f(i)}{k^-+f(i)})^2 \tag{5}$$

数据对象与其邻域的平均相似度越小,说明该数据对象属于此邻域的可能性越小,因此拾起概率就大,反之亦然。公式中参数 k^+和 k^-决定着相似度函数 $f(i)$对 p_{drop}和 p_{pick}取值产生的影响。

2.3 离散点

离散点[10]是一些与数据的一般行为或数据模型不一致的数据对象,是对差异和极端特例的描述,如标准外的特例、数据聚类外的离群值等。

设 $X=\{x_1,x_2,\cdots,x_n\}$,$Y=\{y_1,y_2,\cdots,y_n\}$为 n 维向量,$X\subseteq D$,$Y\subseteq D$,D 是样本集,$Y=\{y_1,y_2,\cdots,y_n\}$称为离散数据,满足以下条件 $E'[Y-E'X]\geqslant\varepsilon$,其中 ε 为离散度,E' 为样本集合整体特征化算子。

2.4 空间熵(spatial entropy)

空间熵反映了样本空间中聚类的程度。计算公式为:

$$E(\Theta)=-\sum_{k\in R}\frac{N_k}{N}*\log(\frac{N_k}{N}) \tag{6}$$

其中 N_k 为第 k 个子类中节点的个数,N 为数据样本空间 R 节点的个数。随着聚类分析的过程进行,空间熵逐渐变小,并达到最小值。

2.5 平均适度(mean-fit)

平均适度反映了整个样本空间中节点间的相似度。计算公式为:

$$F(\Theta)=\frac{1}{N}\sum_{i=1}^{N}f(i) \tag{7}$$

$f(i)$为节点 i 的相似度函数,为样本空间中节点的个数。$F(\Theta)$ 取值范围为[0,1],随着聚类分析的过程进行,平均适度逐渐变大,并达到最大值。

空间熵和平均适度分别作为聚类分析的评价函数,从不同的方面反映了蚁群聚类算法进行聚类分析时数据形成聚类的变化过程。

2.6 算法思想

蚁群聚类算法思想是:将待聚类的数据对象看作是蚁巢中的一个分类对象,每个对象随机分布在平面上,蚂蚁首先计算节点与观察半径内节点的相似度 f,然后利用概率函数公式计算 p_{pick}和 p_{drop},当 p_{pick}和 p_{drop}大于某个阈值,蚂蚁拾起或放下节点。蚂蚁由单个 agent 模拟,通过单个 agent 与单个 agent 的交互以及单个 agent 与环境的交互作用,实现数据对象的自组织聚类。

3 参数σ对算法的影响

参数σ值的变化直接影响着相似度函数的取值范围。由邻域相似度函数公式知,σ取邻域内格子的个数或邻域内节点的个数。若σ取邻域内格子的个数,其中可能包含有未被节点占有的空格子数,相似度的值会受到邻域内节点的密度的影响,这样计算出的 f 值并不能很好地刻画节点跟邻域节点的相似关系。图1(a)、(b)刻画了节点 A 和节点 B 与邻域节点的分布情况,"—"表示空格。在图中,节点 A、B 的相似度 f 值均为0.36,通过拾起概率公式计算,两个节点都会被移走。但实际上,节点 B 与邻域节点相似,应留在原来的格子中。

0.4	0.4	0.4
0.4	A	0.4
0.4	0.4	0.4

(a)

0.8	0.8	0.8
0.8	B	—
—	—	—

(b)

图1

蚁群算法初期,节点随机分布在平面上,各节点邻域内密度均匀,如果σ值取邻域内节点的个数,则相似度值 f 只与节点邻域内相似度有关,而与邻域内密度无关,由图1(a)、(b)得节点 A、B 的相似度分别为0.36和0.8,这结论和蚂蚁工作的实际情况相符。

图2(a)展示了蚁群聚类算法中,参数σ分别取邻域内格子的个数和取邻域内节点的个数时聚类分析过程中空间熵和mean-fit 的变化情况。当参数σ取值为邻域内节点的个数时,聚类分析的速度加快,空间熵能够较快地降低,并取得更低的数值;同时,mean-fit 却以较快的速度增加,并逐渐稳定于最大值。

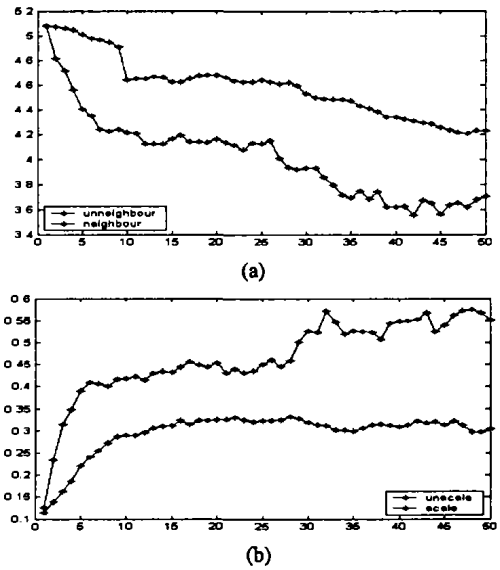


图2

4 算法的改进

4.1 算法的改进

离散点是一些与数据的一般行为或数据模型不一致的数

据对象，是对差异和极端特例的描述，如标准外的特例、数据聚类外的离群值等。这些特例包含很多潜在的知识，探测和分析它们可能有特别的意义，例如在常规商业行为中发现诈骗行为、网络安全管理中的网络入侵检测等。

一个人的噪声可能是另一个人的信号，除去离散点的集合会使领域中相似度最大地提高[11]。实验中，我们发现有负载的蚂蚁找到一个空格时，需要和半径为 radius 的区域中的节点相比较，如果放下的概率大，则放下负载。若放下负载的区域中存在和该负载不相似的离散节点，其相应信息会出现在比较过程中，分析并从该领域中移走这些离散点可以提高领域内相似度，将该信息用于蚂蚁的下一个负载节点的判定信息，可以减少蚂蚁从任意处获得下一负载节点的时间。因此笔者提出一个改进的蚁群聚类算法。图3表示聚类过程中离散点个数的变化情况，其数目和待聚类的数据项数目相比具有相当的比例。

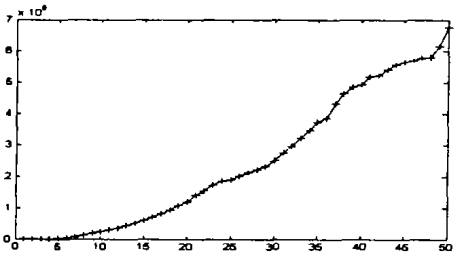


图3

算法中我们采用距离法探测离散点。蚂蚁放下负载的比较过程中，利用距离公式计算和存放领域中和负载节点距离最大的离散节点，计算公式为加权欧几里得距离公式：

$$d_{ij}=\sqrt{q_1|x_{i1}-x_{j1}|+q_2|x_{i2}-x_{j2}|+\cdots+q_n|x_{in}-x_{jn}|} \quad (8)$$

$p_i=(x_{i1},x_{i2},\cdots,x_{in})$和 $p_j=(x_{j1},x_{j2},\cdots,x_{jn})$是两个 n 维节点，q_k 为加权因子，由各分量对聚类的贡献不同而定。

蚂蚁放下负载后将具有最大距离的离散点作为下一个负载节点。聚类过程中蚂蚁总是选择领域中最不相似的节点负载，因此，数据对象领域中离散点是否是最匹配负载点要通过拾起概率公式计算，如果其拾起概率值大于某个给定的常数，则该节点被蚂蚁拾起；否则，从任意处获得负载点。

4.2 算法描述

1. 参数初始化，常数 k^+，k^-，观察半径 *radius*，算法循环次数 *st*，蚂蚁个数 *ant_number*，蚂蚁移动的步长 *step*；

2. 待聚类的模式随机地分配一对坐标(x,y)，其坐标表示模式在平面上的位置；

3. 通过拾起概率公式随机地从模式中选取与其领域节点不相似的节点让蚂蚁负载，将蚂蚁随机地分布在平面上；

4. 程序循环次数加1，判定循环结束条件；循环条件满足，结束，否则，执行下一步；

5. 随机选定一个有负载的蚂蚁；

6. 如果蚂蚁记忆空间中节点数不为零，将负载节点一次放于记忆空间中节点的位置，计算负载模式的相似度函数值 $f(i)$，转7；否则，以步长移动蚂蚁，在新位置对观察半径为 *radius* 的领域计算负载模式的相似度函数值 $f(i)$，用向量 *unsim* 记下和负载模式距离值大于某一概率值的点；

7. 用放下概率公式计算放下概率 p_d；

8. 与一随即概率值比较，若 p_d 大于该值，则蚂蚁放下模式；

9. 用相似度函数公式和拾起公式计算 *unsim* 中与放下模式领域内不相似的节点或任意未被蚂蚁负载的节点的拾起概率 p_p，若拾起概率 p_p 大于某一给定的概率值，蚂蚁负载该节点，转向4；否则，重复9，直到蚂蚁有负载为止；

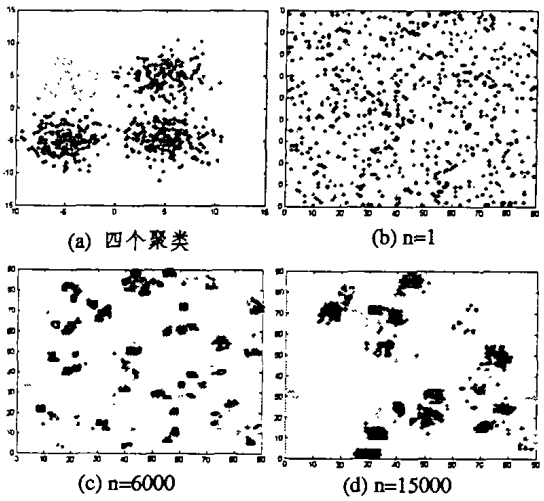
10. 若 p_d 小于该值，蚂蚁继续负载此模式，转向4；

11. 结束。

可以看出步4～10是整个聚类过程；算法的复杂性粗略分析为 $O(n\cdot ant\_number\cdot mem\_unmber\cdot(2\cdot radius+1)^2)$，其中 n 为预设的循环次数，*ant_number* 为蚂蚁的个数，*mem_unmber* 为蚂蚁记忆存储空间长度，*radius* 为观察半径。

4.3 实验仿真和比较

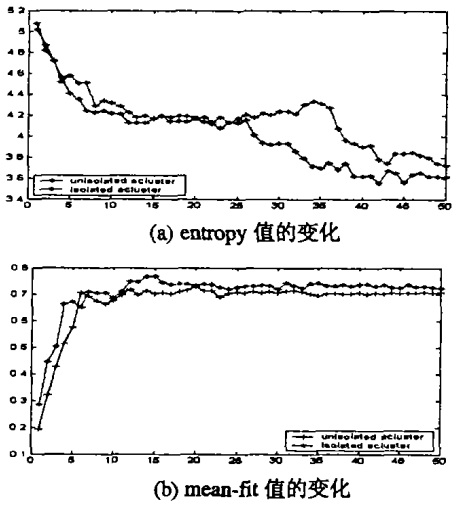
图4(a)展示了四个聚类在属性空间中的分布情况。图4(b)、(c)、(d)分别表示800个数据项随着聚类分析进行的分布情况。$n=1$时，数据随机地分布在二维平面上，随着智能蚂蚁不断地拾起和放下数据对象，各聚类逐渐形成。



(a) 四个聚类　(b) n=1

(c) n=6000　(d) n=15000

图4



(a) entropy 值的变化

(b) mean-fit 值的变化

图5

图5(a)展示了算法改进前后运行20次后的空间熵的平均变化结果，图5(b)展示了算法改进前后运行20次后的均值相似度 *mean-fit* 的平均变化结果。从图形中曲线的变化可知，随着聚类过程的进行，改进后算法的空间熵变小和 *mean-fit* 达到最佳值的速度显然比在原算法中要快，聚类效果也有明显的改善。

为了方便聚类结果的比较和说明，以上测试数据集采用

（下转第223页）

多个编辑工作站在编辑和处理同一个数据库的同一个图案工程时，它们共享同一个图案的数据文件，并且每台编辑工作站都从数据库服务器读取图案信息，在本地存有自己的数据，这些数据不仅有图案数据，还有与图案数据有关的参数数据，编辑工作站对图案的操作只是对自己的数据进行操作，只有在必要时才把数据更新到其他编辑工作站上。在协同时(即在多个编辑工作站在编辑和处理同一个数据库的同一个图案工程时)每台编辑工作站的操作区域在同一个数据文件(即同一个数据源)上不能冲突，这种冲突是在设定操作区域时通过提交操作区域由数据库端进行判断，并返回是否合法。

3.2 协同处理的数据更新

系统在处理协同数据的交换时，采用设计人员根据自己的意愿来进行交换，即认为必要时才去数据交换。采用主动更新和被动更新两种方式。所谓主动更新就是当设计人员认为对目前的操作已经确认，并希望其他设计人员能看到他操作的内容时，主动把当前操作区域的图案数据以及与图案有关的参数数据传送给其他编辑工作站。所谓被动更新就是设计人员当前希望看到其他设计人员的操作内容时，通知其他编辑工作站，把他们的当前操作区域的图案数据以及与图案有关的参数数据更新到本地来。这种采取"按需交换"的方式符合设计的需求，减少网络通讯量，具有快速性、实用性和简捷性等特点。

在协同设计时，如果设计人员需要临时存盘保存数据或者存盘退出时，系统采用"通写式"策略，首先把自己的数据(包括图案数据和参数数据)主动更新到其它有关的编辑工作站上，然后再把数据存储到文件系统。采用这种方式，可以保证当前共享的图案数据是最新的，而且能保证其它编辑工作站上的图案数据也是最新的。

结束语 本文阐述了计算机支持协同工作技术在CAD系统中一个具体实现，如图2，它是一个基于图案设计的计算机协同设计系统，该系统是在Window环境下基于TCP/IP网络环境采用Visual C++实现，该系统具有数据管理的分布性、可扩充性、实用性、快速性等特点，已成功应用于印染CAD系统、提花纹织CAD及陶瓷印花CAD系统，取得了比较好的效果，达到了预期的目的。

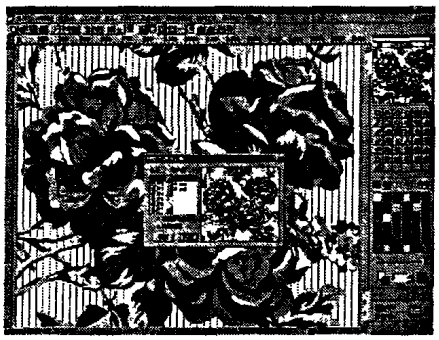


图2

参考文献

1 史美林，杨光信. 计算机支持的协同工作：过去、现在和未来. 计算机研究与发展，1999，36(增刊)：149～154
2 Anupam V. Shastra: Multimedia Collaborative Design Enviroment. IEEE Multimedia Summer 1994
3 Reinhard W. CSCW Tools: Concepts and Architectures. Computer，1994，27(5)
4 Tollmar K，Sundblad Y. The design and building of the graphic user interface for the collaborative desktop. Computer & Graphics，1995，19(2)：179～188
5 Gross T，Traunmuller R. Methodological considerations on design of CSCW. Cybernetics and System，1996，27：279～302
6 Sand M，Maher L. Shared understanding in computer-support collaborative design. Computer Aided Design，1996，28(3)：183～192
7 Sun C，Chen D. Consistency Maintenance in Real-Time Collaborative Graphics Editing Systems. ACM Trans. on Computer-Human Interaction，2002，9(1)：1～41
8 Qian S，Gross M D. Collaborative Design with NetDraw. In：Proc. of Computer Aided Architectural Design Futures'99，1999

(上接第113页)

文[6]中采用的数据集Square1，该数据集是一个具有四个聚类且每个聚类包含有200个数据项，每个聚类的产生服从正态随即分布函数。$\{(N(-5,2),N(-5,2)),(N(5,2),N(5,2)),(N(-5,2),N(5,2)),(N(5,2),N(-5,2))\}$，$n=10^5$，$ant\_numbe=80$，$mem\_unmber=10$，$radius\in[1,3]$。

结束语 聚类是数据挖掘中一门非常有用的技术，用于从大量数据中寻找隐含的数据分布和数据对象。本文提出了利用负载节点其邻域内离散点改进蚂蚁聚类算法，经过试验证明，该算法能够有效改善蚂蚁行为的随机性，避免随着状态空间的改变而进行的刷新操作，缩短蚂蚁寻找负载节点的时间，提高算法的收敛速度。

蚁群聚类算法还有许多值得探索的方面，如算法中σ^2取节点邻居区域中邻居节点的个数，使得相似度函数值的计算只与邻居区域内节点的相似度有关而与邻居区域的密度无关，算法的收敛速度明显加快，但是算法也容易陷入局部最优，这一点将在今后的工作中进一步研究；尽管改进后的蚁群算法有效地对初始数据进行了聚类，但由于蚂蚁的随意性，算法仍需要较长时间收敛。为了得到更好的聚类结果，常常在蚁群聚类算法的初始结果上结合其它聚类方法对聚类算法进行扩展，如k-means算法，基于密度的聚类方法等。因此，蚁群聚类算法和已有的聚类技术相结合也是值得进一步探索的课题。

参考文献

1 高尚，杨静宇，吴小俊. 聚类问题的蚁群算法[J]. 计算机工程与应用. 2004. 40(8)：90～91，232
2 Alsabti K，Ranka S，Singh V. An efficient k-means clustering algorithm. In：Proc. of the First Workshop on High Performance Data Mining，Orlando，FL，March 1998
3 Deneubourg J L，Goss S，Franks N，et al. The dynamics of collective sorting：Robot-like ants and ant-like robots. In：J -A Meyer and S Wilson，eds. Proc. of the First Intl. Conf. on Simulation of Adaptive Behaviour：From Animals to Animats 1，MIT Press，Cambridge，MA，1991. 356～365
4 Lumer E D，Faieta B. Diversity and Adaptation in Populations of Clustering Ants. In：Cliff D，Husbands P，Meyer J. Wilson S，eds. From Animals to Animats 3，Proc. of the 3rd Int. Conf. on the Simulation of Adaptive Behavior. Cambridge，MA：The MIT Press/Bradford Books，1994
5 Handl J，Knowles J，Dorigo M. Ant-based Clustering：A Comparative study of its relative importance with respect to k-means，average link and 1D-SOM：[Technical Report TR/IRIDIA/2003-24]. Universite Libre de Bruxelles，2003
6 Kuntz P，Snyers D. Emergent colonization and graph partitioning. In：Proc. of the third Intl. Conf. on Simulation of Adaptive Behavior：From Animals to Animats 3 (SAB 94)，D. Cliff，P. Husbands，J. A. Meyer，S W Wilson，eds. MIT Press，1994. 494～50
7 Monmarch'e N，Slimane M，Venturini G. On improving clustering in numerical databases with artificial ants. In：Lecture Notes in Artificial Intelligence，D Floreano J D Nicoud，F Mondala，eds. Swiss Federal Institute of Technology，Lausanne，Switzerland，(13-17 September 1999). Springer-Verlag，1999. 626～635
8 吴斌，郑毅，傅伟鹏，史忠植. 一种基于群体智能的客户行为分析算法[J]. 计算机学报，2003，26(8)：913～918
9 乔银锋，顾军华，张勇. 蚂蚁算法建立度限制树在聚类中的应用[J]. 天津理工学院学报. 2004，20(1)：18～19，23
10 Hawkins D. Identification of Outliers. London：Chapman and Hall，1980
11 Han Jiawei，Kamber M. 范明，孟小峰，等译. 数据挖掘概念与技术[M]. 机械工业出版社，2001