具有全局指导的启发式蚁群聚类新算法

牛永洁
(延安大学 计算中心,陕西 延安 716000)

摘 要:蚁群聚类 LF 算法是基于蚂蚁堆形成原理而产生的群体智能算法,存在收敛速度慢、易陷入局部最优等缺陷。为了提高 LF 算法的收敛速度,在算法中提供具有全局意义的记忆中心,算法运行初期,蚂蚁根据全局记忆中心的启发信息运行,随着算法的迭代,不断更新全局记忆中心。为了避免算法陷入局部最优,在全局记忆中心的指导下,每只蚂蚁向距离最小的点运动,而不是采用直接跳转的方法。新算法使用 UCI 数据集中的 Iris 和 Wine 验证,算法的查准率和查全率要优于其他算法。

关键词:蚁群聚类;全局记忆;启发信息;查准率;查全率
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2013)09-0074-04
doi:10.3969/j.issn.1673-629X.2013.09.019

New Algorithm of Heuristic Ant Colony Clustering with Global Guidance

NIU Yong-jie
(Computing Center, Yan'an University, Yan'an 716000, China)

Abstract: LF ant colony clustering algorithms is swarm intelligence algorithm which is based on the principle of ant heap formation, slow to converge and easy to fall into the local optimum. In order to improve the convergence speed of the LF algorithm, memory center of global significance is provided, when the algorithm runs early, the ants run according to the heuristic information from global memory center, with the iteration of the algorithm, constantly update the global memory center. In order to avoid the algorithm into a local optimum, under the guidance of the global memory center, each ant moves to the minimum distance point, rather than directly jumps. The new algorithm uses UCI dataset Iris and Wine verification, the algorithm precision rate and the recall rate is better than the other algorithms.

Key words: ant colony clustering; global memory; heuristic information; precision rate; recall rate

1 LF 算法概述

根据蚂蚁会将尸体按照大小不同堆积成蚂蚁墓地行为的启发[1],Deneubourg 等人提出了一种基本模型(Basic Model,BM)[2]。Lumer 和 Faieta 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[3]。

LF 算法的运行原理很简单,一只蚂蚁在一个布满数据点的二维网格中爬行,如果该蚂蚁目前没有背负任何数据(空载),当它碰到一个数据点,计算该数据点和周围邻域中其他数据点的相似度,如果相似度小于某个随机数,说明该数据点与周围的数据不相似,蚂蚁不应该捡起该数据,蚂蚁将随机移动到其他位置;否则,蚂蚁将捡起该数据而变成负载状态,负载的蚂蚁随机移动到另一位置,如果该位置已经有数据点,蚂蚁将继续随机移动,如果新位置没有数据,计算背负的数据与邻域数据点的相似度,如果相似度大于某个随机数,蚂蚁将放下该数据点,然后随机移动到其他位置。经过一段时间的运行,蚂蚁会将相似点放在一起而形成一个聚类。

LF 算法的描述是[4-5]:将准备聚类的 D 维 N 个数据对象随机地投影到一个 $M \times M$ 的二维网格中,保证每个单元格中只有一个数据对象,然后将 A 只蚂蚁也随机地投影到该二维网格中,每个蚂蚁都有一个长度为 L 的短期记忆。单个蚂蚁如果找到一个数据对象 O_i,计算该数据对象的邻域半径为 r 的与其他数据对象的相似度 $f(O_i)$,如公式(1)所示。

$$f(o_i)=\begin{cases}1/S^2 * \sum\limits_{o_j \in S(o_i)} |\ 1-d(o_i,o_j)/\alpha\ | \\ \qquad f(o_i) > 0, S \neq 0 \\ 0 \qquad \text{otherwise}\end{cases} \tag{1}$$

收稿日期:2012-11-01 **修回日期**:2013-02-03 **网络出版时间**:2013-04-22
基金项目:陕西省高等继续教育教学改革研究项目(11J23)
作者简介:牛永洁(1977-),男,河南许昌人,讲师,硕士,CCF 会员,研究方向为软件工程、数据挖掘。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20130422.1727.051.html

其中 $r=(S-1)/2$,$d(o_i,o_j)$ 表示数据对象 O_i,O_j 之间的欧式距离,参数 $\alpha\in[0,1]$ 为相似度调整因子,它最终影响聚类的质量,若 α 过大,则不同对象之间的区分度不明显,可能导致不同簇的数据对象会聚为一个簇;反之若 α 过小,则会导致对象间差异度量被放大,从而使原本能聚合到一个簇中数据对象分裂为若干个簇,并且会导致收敛速度变慢。

通过 $f(O_i)$ 获得蚂蚁捡起 O_i 的概率 P_p,P_p 由公式(2)计算。

$$P_p(o_i)=(k_1/(k_1+f(o_i)))^2 \tag{2}$$

其中 k_1 是处于(0,1]之间的一个常数,如果 P_p 大于某个随机产生的数,该蚂蚁将捡起数据对象 O_i,根据蚂蚁本身拥有的短期记忆,让蚂蚁朝着匹配位置进行随机移动。否则随机移动到其他的数据点上。

捡起数据点的蚂蚁移动到新位置,首先判断该点是否有数据点,如果新位置处有数据点,让蚂蚁朝着匹配位置进行再次的随机移动,直到新位置处没有数据点。如果新位置没有数据点,根据公式(1)计算相似度 $f(O_i)$,然后按照公式(3)计算放下数据点的概率 P_d。

$$P_d(o_i)=\begin{cases}2f(o_i) & f(o_i)<k_2\\ 1 & \text{otherwise}\end{cases} \tag{3}$$

如果 P_d 大于某个随机数,蚂蚁将数据点放下,在自己的短期记忆中记录放下数据点的位置,然后随机移动到另一个没有被其他蚂蚁捡起的数据点上。否则蚂蚁朝着匹配位置进行随机移动。

蚁群在经过若干次的移动、拾起与放下动作后实现对所有对象的聚类,最终输出聚类结果。

2　算法的缺陷及改进

2.1　算法的缺陷

(1)从公式(1)可以看出,相似度 $f(O_i)$ 的计算受邻域 S 的影响,当一个数据点在二维表格的中部时,其四周的邻域是完全的,但是当一个数据点位于二维表格的边界区域时,其邻域遇到的二维表格的边界,使得其邻域的实际大小要小于理论值,造成计算相似度 $f(O_i)$ 时,对边界数据点的计算不准确。邻域 S 缺陷的示意图如图(1)所示,在图 1 中,黑色的圆点代表数据点,而灰色的矩形区域是半径为 1 的邻域区域,可以看出两个数据点邻域区域的大小因数据点位置的不同而存在差别。

(2)LF 算法在运行过程中,存在着收敛速度慢的问题,通过试验发现影响收敛速度的因素有参数 α 和蚂蚁空转问题。如果参数 α 设置的比较大,收敛速度会比较快,但是会将不相似的数据聚为一类,严重影响聚类的效果,但是如果将参数 α 设置较小,会发现蚂蚁在捡起数据点后,经过多次的迭代也不能将数据点放下,算法将会长时间停滞在所有蚂蚁都背负数据点而不能找到合适位置将数据放下的情况。

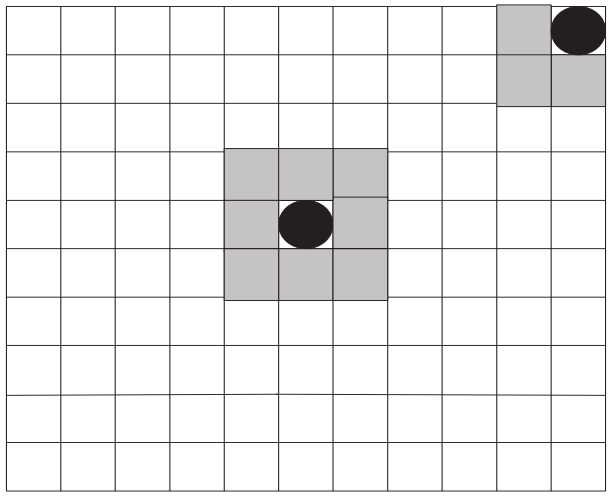

图 1　半径为 1 的邻域

在蚂蚁空载时,为了找到一个数据点,需要随机移动很多次,或者蚂蚁在负载情况下,为了找到一个空的数据单元,也需要移动很多次,这些情况被称为蚂蚁的空转,空转无疑消耗了算法有效的迭代资源。

2.2　算法的改进措施

针对由于数据点在二维表格中的位置不同而造成邻域不同的缺陷,新算法将采用球面坐标[6-7],即将二维表格的左右边界、上下边界连接在一起,保证每个数据点无论它位于二维网格的任何位置都有完整的邻域范围,而且在计算 $f(O_i)$ 时,将该邻域范围中已经被其他蚂蚁捡起的数据点排除。

为了减少蚂蚁的空转现象,在蚂蚁空载时,将蚂蚁直接随机地放置在某个数据点上,而在蚂蚁负载后,将蚂蚁直接随机放置到空的单元格中,这样就减少了蚂蚁的空转现象[8]。

在保证聚类效果的情况下,为了提高聚类的速度,在蚂蚁移动的过程中不能采用随机移动的原则,而应该是在一定的指导思想下进行。LF 算法引入了局部记忆的概念,每只蚂蚁将自己以前放下的数据点和位置存入自己的局部记忆中,一旦捡起一个数据,首先比较捡起的数据点与局部记忆中的哪个数据点之间的距离最短,然后将蚂蚁直接跳转到局部记忆中数据点的位置。

局部记忆的策略的确能够加速蚁群聚类的速度,但在算法运行初期,由于每个蚂蚁的局部记忆都是空的,造成算法运行早期,蚂蚁仍然是随机移动,浪费了大量的运行时间,而且由于每个蚂蚁的局部记忆中存入的数据不全面,造成很多数据点的错误移动,甚至会造成两只蚂蚁互相移动某个数据点的情况,即一只蚂蚁将一个数据点移动到甲位置,而另一只蚂蚁将该数

据移动到乙位置,接着另一只蚂蚁又将该数据点移动到甲位置,另一只蚂蚁又会将该数据移动到乙位置,或造成蚂蚁群体做功互相抵消的问题。而且采用直接将捡起的数据点移动到距离最短点的附近会很容易造成局部最优的问题。

针对以上问题,新算法引入了全局记忆的概念和启发信息的方法。在算法初始化阶段引入一个 $G \times 2$ 的矩阵 GlobalMemory,该矩阵的目的是为了记忆所有蚂蚁放下数据时的数据点和位置信息。矩阵的两列分别记录数据点和放下该点时的 P_d。

矩阵 GlobalMemory 在算法运行初期将随机选择 G 个数据点,放入的第一列,第二列全部置为 0,每只蚂蚁捡起数据后,立即可以将捡起的数据点与矩阵 GlobalMemory 中的点比较,找到距离最小的点的位置,然后将蚂蚁向距离最小点的位置移动,而不是采用直接跳转的办法,这样就可以有效地避免算法陷入局部最优[9]。

当一只蚂蚁放下一个数据点,将会更新全局记忆中的信息,更新的措施如下:

1)在全局记忆中查看有没有被其他蚂蚁捡起的数据点,如果有数据点被其他蚂蚁捡起,说明该数据点已经失效,将当前放下的数据点和位置信息替换已经失效的数据点和位置信息。

2)如果在全局记忆中,没有被其他蚂蚁捡起的数据点,寻找全局记忆中 P_d 值最小的数据点,然后将刚放下的数据点的 P_d 与 P_d 最小值的数据点比较,如果新的 P_d 值大于等于最小 P_d,替换原先信息,否则,保持原状。

算法迭代结束,还会有很多数据点没有被蚂蚁放下,采用强制放下策略,再放下数据时,每只蚂蚁在全局记忆中找到与当前背负的数据距离最小的数据点,然后得到该点邻域中的无数据点的单元格,随机选择一个,将背负的数据点放下。

综上所述,新算法的流程步骤如下:

1)待聚类数据的标准化,本算法采用归一化的方法对数据进行标准化操作。

2)将聚类数据随机分布到一个二维网格中,每个单元格中最多放置一个数据点,将蚂蚁直接随机放到数据点上,设置算法参数。

3)迭代开始。

4)对于任意一只蚂蚁,查看是否空载,如果蚂蚁空载,执行步骤 5。

5)直接将蚂蚁放置到任意一个数据点上。

6)计算该数据点与邻域数据点的相似度 $f(O_i)$ 和捡起概率 P_p,如果 P_p 小于一个随机数,随机将蚂蚁移动到另一个数据点上。否则,捡起该数据点,标记该点被捡起,从全局记忆中查找没有被其他蚂蚁捡起,并且距离当前数据点欧式距离最短的数据点,得到该数据点的位置坐标,让蚂蚁向该坐标随机移动,并到达一个空白单元格中。

7)如果蚂蚁已经负载,查看蚂蚁当前位置是否有数据点,如果有数据点,按照全局记忆中的信息指导移动到空白单元格中,如果蚂蚁所在位置没有数据点,执行步骤 8。

8)计算蚂蚁负载点与邻域中数据点的相似度 $f(O_i)$ 和 P_d,如果 P_d 小于一个随机数,将蚂蚁随机移动到任一空白单元格中,如果 P_d 大于随机数,将数据点放下,同时更新全局记忆。

9)算法迭代次数结束,强行放下还被蚂蚁背负的数据点。

10)计算算法的查准率和查全率。

3 算法验证

为了验证新算法,采用 UCI 数据集中的 Iris 数据和 Wine 数据,Iris 数据是一个 150×5 的矩阵,最后一列是数据的分类标准。Wine 数据是一个 178×14 的矩阵,其中第一列是分类标准。

算法中参数的设置[10-11]为 $K_1 = 0.1$,$K_2 = 0.15$,Iris 数据的 α 设置为 0.21,Wine 数据的 α 设置为 0.46,邻域 $S = 9$,二维网格的大小是 40×40,蚂蚁个数为 20,全局记忆的长度为 50,每个蚂蚁局部记忆的长度是 10,每次算法迭代 10 000 次。为了作为对比,同时实现了采用完全随机移动策略的随机蚁群算法(Random Ant Colony Optimization,RACO)、采用局部记忆移动方法的局部蚁群算法(Local Ant Colony Optimization,LACO),新算法命名为全局蚁群算法(Global Ant Colony Optimization,GACO)。算法的评价标准采用 F_{Measure},具体计算公式可参考文献[7]。

将每个算法单独运行 10 次,取其平均值。算法的运行结果如表 1 所示。

表 1 算法运行的 F_{Measure} 结果比较

	Iris 数据	Wine 数据
RACO	0.712 5	0.711 6
LACO	0.818 1	0.807 8
GACO	0.857 1	0.865 7

图 2 ~ 图 4 分别是 Iris 数据采用 RACO、LACO、GACO 算法运行的结果图。图 5 是 Wine 数据 GACO 算法上的运行效果图。

4 结束语

由于算法 GACO 在初始化阶段随机将部分数据点

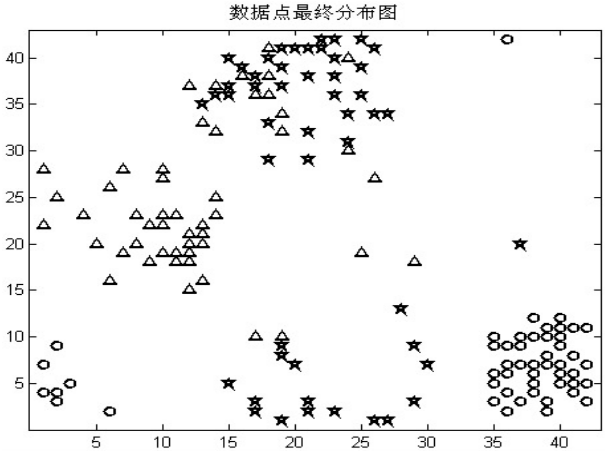

图 2　RACO 运行结果图

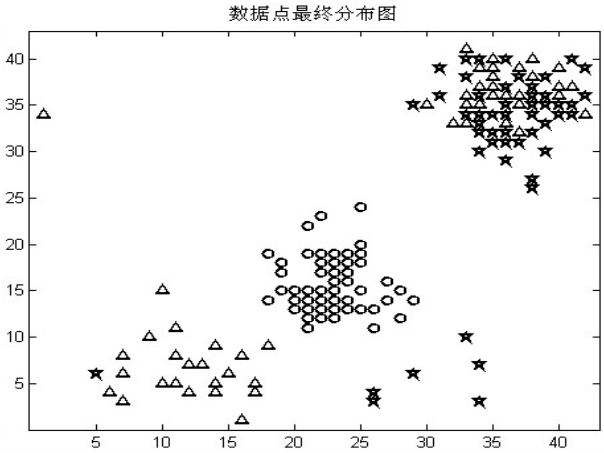

图 3　LACO 运行结果图

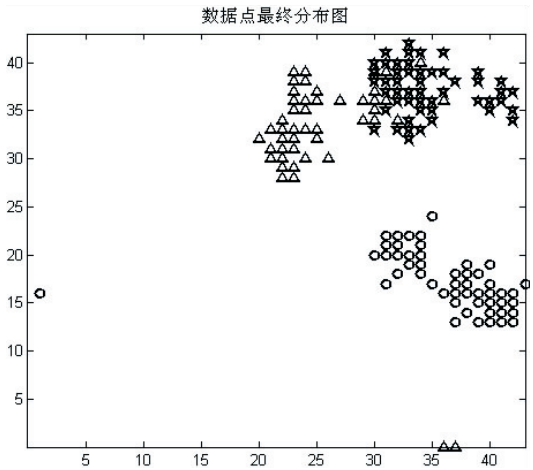

图 4　GACO 运行结果图

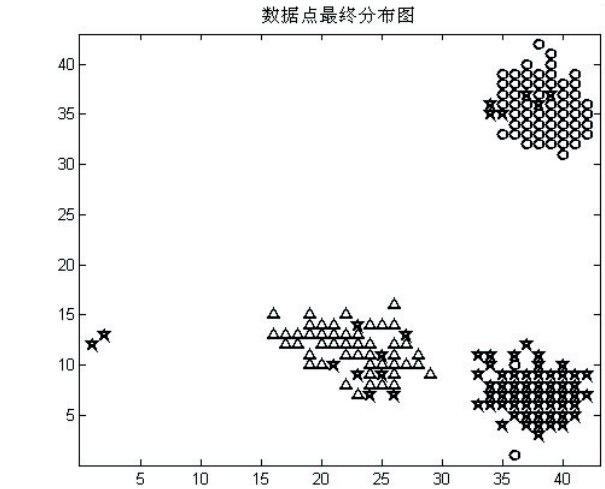

图 5　GACO 运行结果图(Wine 数据)

放入全局记忆中,对起始阶段蚂蚁放下捡到的数据点时具有一定的启发作用,顺着蚂蚁放下数据点时对全局记忆的更新,使得全局记忆对蚂蚁的运动的指导作用越来越强,指导的准确性也越来越高,由于采用向距离最小点的方向随机移动而不是直接跳转到该点,有效避免了算法局部最优化的缺陷。算法运行结束根据全局记忆的强制放下策略也比较科学。

通过试验验证,GACO 算法性能要优于其他传统算法。

参考文献:

[1] 张建华,江　贺,张宪超. 蚁群聚类算法综述[J]. 计算机工程与应用,2006,42(16):171-174.

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

[3] Lumer E, Faieta B. Diversity and adaptation in populations of clustering ants[C]//Proceedings of Third International Conference on Simulation of Adaptive Behavior. Cambridge, MA, USA: MIT Press, 1994:501-508.

[4] 朱　峰,陈　莉. 一种改进的蚁群聚类算法[J]. 计算机工程与应用,2010,46(6):133-135.

[5] 陈寿文. 混合均值聚类算法及 LF 蚁群聚类算法研究[D]. 南充:西华师范大学,2010.

[6] 梁君玲,肖人岳,王向东. 一种改进的自适应蚁群聚类算法[J]. 计算机应用研究,2011,28(4):1263-1265.

[7] 朱　峰. 蚁群算法在聚类分析中的应用研究[D]. 西安:西北大学,2009.

[8] 李玲娟,李　冰. 一种基于特征加权的蚁群聚类新算法[J]. 计算机技术与发展,2010,20(8):67-70.

[9] 孟　非,李静宜,朱人杰. 蚁群算法中蚂蚁更新方法之研究[J]. 计算机工程与应用,2011,47(25):54-57.

[10] Handl J, Knowles J, Dorigo M. Ant-based clustering: a comparative study of its relative performance with respect to K-means, average link and 1d-som[R/OL]. 2003. http://www.handl.julia.de.

[11] Handl J. Ant based methods for tasks of clustering and topographic mapping improvements evaluation and comparison with alternative methods[D]. UK: The University of Manchester, 2003.

具有全局指导的启发式蚁群聚类新算法

作者： 牛永洁， NIU Yong-jie
作者单位： 延安大学 计算中心,陕西 延安,716000
刊名： 计算机技术与发展
ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2013(9)

参考文献(11条)

1. 张建华. 江贺. 张宪超 蚁群聚类算法综述[期刊论文]-计算机工程与应用 2006(16)

2. Dorigo M. Bonabeau E. Théraulaz G Ant algorithms and stig-mergy[外文期刊] 2000(08)

3. Lumer E. Faieta B Diversity and adaptation in populations of clustering ants 1994

4. 朱峰. 陈莉 一种改进的蚁群聚类算法[期刊论文]-计算机工程与应用 2010(06)

5. 陈寿文 混合均值聚类算法及LF蚁群聚类算法研究 2010

6. 梁君玲. 肖人岳. 王向东 一种改进的自适应蚁群聚类算法[期刊论文]-计算机应用研究 2011(04)

7. 朱峰 蚁群算法在聚类分析中的应用研究[学位论文] 2009

8. 李玲娟. 李冰 一种基于特征加权的蚁群聚类新算法[期刊论文]-计算机技术与发展 2010(08)

9. 孟非. 李静宜. 朱人杰 蚁群算法中蚂蚁更新方法之研究[期刊论文]-计算机工程与应用 2011(25)

10. Handl J. Knowles J. Dorigo M Ant-based clustering:a com-parative study of its relative performance with respect to K-means,average link and 1d-som 2003

11. Handl J Ant based methods for tasks of clustering and topo-graphic mapping improvements evaluation and comparison with alternative methods 2003