基于信息熵的蚁群聚类组合算法的研究 *

田力威, 曹安得
(沈阳大学 科学技术研究中心, 沈阳 110044)

摘 要: 提出一种基于信息熵的蚁群聚类算法,将信息熵引入到 LF 算法中,数据对象的归属由信息熵来决定,减少了参数,测试并验证了算法的有效性;同时,针对信息熵的蚁群算法早期数据分散、收敛过慢、容易陷入局部最优等缺点,提出了一种蚁群聚类组合方法。改进思路是引入 K-means 作为熵蚁群算法的预处理过程,通过 K-means 快速、粗略地确定聚类中心,利用 K-means 方法的结果作为初值,再进行改进的熵蚁群算法聚类,有效地解决了蚁群算法早期收敛过慢等问题。

关键词: 聚类; 蚁群聚类; 信息熵; K-均值
中图分类号: TP18 **文献标志码**: A **文章编号**: 1001-3695(2011)04-1269-03
doi:10.3969/j.issn.1001-3695.2011.04.019

Analysis of ant colony clustering combination based on information entropy

TIAN Li-wei, CAO An-de
(*Science & Technology Research Center*, *Shenyang University*, *Shenyang* 110044, *China*)

Abstract: Proposed a new ant colony clustering based on information entropy, introduced the entropy into the LF algorithm, which determined the state of the data, and reduced the parameters to test and verify the effectiveness of the algorithm. At the same time, for the information entropy of ant colony algorithm's early data were too scattered so convergence was slow. Vulnerable to the shortcomings of local optimum, presented a combination method to improve the ant colony clustering. The paper introduced K-means to the pre-computation process of ant colony algorithm. Through K-means, it determined cluster center fast and sketchily, and got the starting value using the K-means result, then clustered by the improved algorithm. It effectively solve the slow convergence of ant colony algorithm for the early issues.

Key words: clustering; ant colony clustering; information entropy; K-means

聚类分析是数据挖掘的重要组成部分。20 世纪 80 年代, Deneubourg 等人首次模拟幼蚁自动分类及蚁尸聚积现象,提出了聚类基本模型(BM);后又由 Lumer 等人对基本模型进行改进提出了 LF 算法。近年来,国内外学者将蚁群算法的群体智能应用到聚类问题,得到了比较满意的结果,并且出现了在 LF 基础上改进的蚁群聚类算法。由文献[1,2]提出的基于信息熵的聚类算法和文献[3]提出的一种基于信息熵的蚁群聚类算法,将信息熵引入到 LF 算法中,改变了蚂蚁拾起和放下对象的判定规则。

在研究中发现,LF 本身和基于信息熵的蚁群聚类算法都有一定的局限性。LF 算法需要调节很多参数,参数设置就有一定难度,对收敛产生了不确定性,同时在复杂问题上收敛速度过慢;基于信息熵的蚁群算法和 LF 同时都有早期收敛过慢、容易陷入到局部解现象。本文应用 K-means 对信息熵蚁群算法进行预处理,再运用改进的熵蚁群算法。实验结果表明,改进后的方法在聚类的准确性和收敛速度方面都得到了很好的结果。

1 K-means 聚类算法

K-means 算法是在科学和工业应用中较流行的聚类工具。算法的名字源于利用簇类点均值或加权平均值 c_i(质心)作为簇 C_i 的代表点。该算法不断计算每个聚类的中心,也就是聚类中对象的平均值作为新的聚类种子。通常 K-means 算法采用的目标函数形式为平方误差准则函数:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \| p - c_i \| \tag{1}$$

其中:p 表示数据对象,c_i 表示簇 C_i 的均值(聚类中心)。

K-means 算法具体描述如下[4]:

输入:n 个对象的数据库,期望得到的簇的数目 k;

输出:使得平方误差准则函数最小化的 k 个簇。

a)选择 k 个对象作为初始的簇的质心;

b)repeat;

c)计算对象与各个簇的质心的距离,将对象划分到距离其最近的簇;

d)重新计算每个新簇的均值;

e)until 簇的质心不再变化。

K-means 算法试图找出使平方误差函数值最小的 k 个划分。当结果簇密集并且各簇之间的区别明显时,它的效果较好;处理大数据集时,K-means 算法具有较好的可伸缩性和高效率。K-means 聚类算法存在的问题是:当结果簇密集,但区别不明显时则效果较差。该算法的缺点在于要事先给出期望生成簇的数目 k。

收稿日期: 2010-09-13; **修回日期**: 2010-10-29 **基金项目**: 辽宁省自然科学基金资助项目(20082002)

作者简介:田力威(1973-),男,科技处处长,教授,博士后,主要研究方向为敏捷 ERP、信息管理与商务智能;曹安得(1986-),男,湖北荆州人,硕士研究生,主要研究方向为电子商务、数据挖掘与知识发现、蚁群算法应用(cad19861014@163.com).

2　LF 算法和信息熵的 LF 算法

2.1　LF 算法的主要思想

在一个 $Z\times Z$ 的网格中,蚂蚁在地点 r 可以观察到周围 $S\times S$ 的区域中的物体(下面称对象)。对象 O_i 在地点 r 与周围对象的相似度按式(2)计算。其中,α 是一个衡量相异度的参数,$d(O_i,O_j)$是两个对象 O_i 和 O_j 的距离,通常使用欧几里德距离。

$$f(O_i)=\begin{cases}\dfrac{1}{S^2}\sum\limits_{O_j\in \text{neigh}_s(r)}\left[1-\dfrac{d(O_i,O_j)}{\alpha}\right] & \text{if } f(O_i)>0\\ 0 & \text{otherwise}\end{cases} \tag{2}$$

$$P_p(O_i)=\left(\frac{k_1}{k_1+f(O_i)}\right)^2 \tag{3}$$

$$P_d(O_i)=\begin{cases}2f(O_i) & \text{if } f(O_i)<k_2\\ 1 & \text{if } f(O_i)\geqslant k_2\end{cases} \tag{4}$$

在 LF 算法中,蚂蚁拾起和放下一个对象的可能性按式(3)(4)分别计算。拾起或放下的规则是:将取一个随机数 r 与计算所得的拾起或放下可能性值比较,若随机数小则执行拾起或放下操作。这个随机数 r 会导致一个对象多次被拾起或放下,从而聚类速度较慢,且易出现一个或多个对象未能放下,使得系统出现停滞,收敛于局部点。

2.2　信息熵的 LF 算法

这里采用 Shannon[5,6] 给出的关于信息熵定义:假设 x 是一个随机变量,X 是其可能的取值集合(连续型数据需要离散化),$p(x)$是取 x 值的可能性函数。信息熵 $E(x)$的定义如下:

$$E(X)=-\sum_{x\in S(X)}P(x)\lg p(x) \tag{5}$$

假定变量的各个属性独立不相关,则

$$E(\hat{x})=-\sum_{x\in s(X_1)}\cdots\sum_{x\in s(X_n)}(p(x_2)\cdots p(x_n))\lg(p(x_2)\cdots p(x_n))=E(X_1)+E(X_2)+\cdots+E(X_n) \tag{6}$$

$$p(x)=\frac{\text{number\_of\_}x}{\text{number\_of\_objects}} \tag{7}$$

其中:number_of_x 是 $S\times S$ 区域中满足 $A_i=x$ 的对象个数;number_of_objects 是 $S\times S$ 邻域中的对象总数。

本文在传统的 LF 算法中引进了信息熵(entropy ant cluster,EAC)的概念,从而改变了蚂蚁拾起或放下的判断规则。其主要思想如下:

a)一个未负载的蚂蚁移到对象 O_i 处,计算周围 $S\times S$ 区域中的对象信息熵。假设未拾起对象 O_i 前的信息熵为 E_1,拾起对象 O_i 后该区域的信息熵变为 E_2,拾起规则为 if $E_1>E_2$,则拾起对象 O_i。

b)一个负载对象 O_i 的蚂蚁移到空白之处,计算周围 $S\times S$ 的区域中的对象信息熵。假设未放下对象 O_i 前的信息熵为 E_1,放下对象 O_i 后该区域的信息熵变为 E_2,放下规则为 if $E_1>E_2$,则放下对象 O_i。

LF 算法首先通过式(2)求出相似度,然后通过概率转换函数(式(3)(4))求出对象与邻域内相似概率,最后给定一个随机数来判断是否满足拾起放下条件。相似度与概率转换带有很大的随机性,参数选取没有理论上的方法。基于 EAC 算法减少了参数数目,加快了聚类速度,但是由于聚类前期数据散布模糊,EAC 算法也会出现频繁拾起或放下等情况,且易陷入局部最优。本文提出了基于 K-means 的改进息熵蚁群聚类算法。

3　基于 K-means 的改进信息熵蚁群聚类组合算法

通过实验验证,K-means 算法收敛速度比熵蚁群聚类算法快,因而引入 K-means 作为蚁群算法的预计算过程[7,8]。通过 K-means 快速、粗略地确定聚类中心,即“食物源”,利用 K-means 方法的结果作为信息熵蚁群聚类算法初值,改善了初始数据太过分散产生蚁群聚类算法早期收敛过慢等现象,即分出相应的聚类堆,然后对聚类实行两阶段法:(a)对象通过类间熵值比较找出与此对象熵值最小的聚类,将对象放在此类中,让对象进行各个类放下的熵值比较,以改进 K-means 算法分类点粗糙的缺点,进行深加工,可以得到聚类数目未变(依然是 K-means 算法得出的数目)、聚类中心精确度比 K-means 算法更高的簇。(b)在类间聚类的基础上实现蚁群信息熵的全局聚类,即原始信息熵蚁群聚类算法。一方面蚁群算法的鲁棒性(稳定性)可以有效地克服初始化的敏感度问题,如聚类中心的不确定等;另一方面它的并行分布式计算可以加速收敛,提高聚类效率,获得更多模式的解。

因此引入 K-means 作为预计算求解聚类问题的信息熵蚁群算法,作为一种蚁群聚类组合方法(KIEAC)思想如下:

a)任选 K 个初始聚类中心:$C_1,C_2,C_3,\cdots,C_K$。

b)逐个将数据集$\{X\}$中各个数据对象按最小距离原则分配给 k 个聚类中心的某一个 C_i。

c)计算新的聚类中心 $C_i'(i=1,2,\cdots,k)$,即 $C_i'=\dfrac{1}{N_i}\sum\limits_{x\in S_i}X$,其中 N_i 为第 i 个聚类域 S_i 包含的个数。

d)停止准则。(a)直到簇的质心不再变化且未快速分类到设定聚类效果阈值 γ,即 $C_i'=C_i$;(b)迭代代数大于指定代数 max_cn;否则转 b)。

e)初始化信息熵。K-means 由算法分类结果计算出的聚类中心 $C_i(i=1,2,\cdots,N)$,cn 为循环次数,n 为蚂蚁数,ε,初始蚂蚁无负载。

f)对象到各个类的信息熵进行类间比较,按最小信息熵 $E(X)$放下对象,直到$|C_i'-C_i|\leqslant\varepsilon$ 转 g)。

g)全局聚类。蚂蚁向任意方向移动,即原始信息熵蚁群聚类,通过拾起/放下前后的信息熵 E_1 和 E_2 的比较来判断对象拾起或放下状态。

h)until 停止条件。可以使用如下两种条件来终止:(a)$|C_i'-C_i\leqslant\varepsilon|$,此时 C_i 中的数目不同于 K-means 算法生成的聚类中心数;(b)迭代代数大于最大迭代代数,后者一般容易出现局部解而停滞。否则转 g)。

基于 K-means 的信息熵蚁群聚类组合算法流程如图 1 所示。

4　算法测试

实验数据取于 UCI 机器学习数据库的 iris 和 wine 及 balance 数据集(表 1)。这些数据库有自己的分类,可用于聚类性

能的评价。本文分别用 KMEAC、LF 和 K-means 算法进行测试。

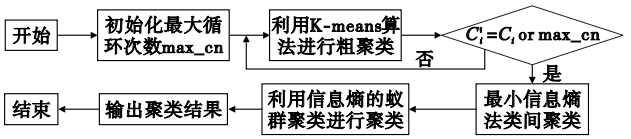

图1　基于K-means的信息熵蚁群聚类组合算法

表 1　数据库描述

数据库描述	iris	wine	balance
数据大小	150	178	214
属性个数	4	12	4
分类数目	4	4	4

本文对 KIEAC、EAC 和 K-means 算法的平均执行时间、平均出错率、出错百分比、最大迭代次数进行比较,如表 2 所示。K-means 中 K 值为 3,KMEAC 没有参数;对每个测试数据集重复进行 100 次测试。

表 2　EAC、KIEAC 和 KEAC 对数据集合的测试结果

比较项	iris			wine			balance		
	EAC	KIEAC	KEAC	EAC	KIEAC	KEAC	EAC	KIEAC	KEAC
最大迭代次数	10 万	5 500	5 500	10 万	5 500	5 500	10 万	5 500	5 500
平均执行时间/s	56.52	1.32	1.27	62.24	1.42	1.38	106.01	2.34	2.30
平均错误率	6.71	4.35	4.43	8.11	4.75	4.91	10.34	7.53	7.67
出错百分率/%	4.47	2.90	2.95	4.56	2.67	2.76	4.83	3.52	3.58

由表 2 可知,在同等情况下 EAC 算法要迭代 10 万次,而 KIEAC 和 KEAC 迭代 5 500 次就远比 LF 算法分类精确了,明显可以看出 EAC 算法比 KEAC 和 KIEAC 算法所需要的时间代价大。这是因为初始数据分布太分散,蚂蚁对信息熵依赖,而在拾起和放下数据过程中会出现频繁拾起放下某物体的现象,以致于大量的时间花费在寻找数据和聚类的质量上。KIEAC 相比 KEAC 时间较多,但是准确率较高,可以看出 KIEAC 算法在最小熵法类间聚类花的时间较多,但是却增加了算法的精度。

表 3 分析出 KIEAC 除了时间开销高于传统的 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,因此本文为它预先设定了类的正确个数,使得它的聚类速度较快;而 KIEAC 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。

表 3　KIEAC 和 K-means 对数据集合的测试结果

比较项	iris		wine		balance	
	KIEAC	K-means	KIEAC	K-means	KIEAC	K-means
平均执行时间/s	1.32	0.03	1.42	0.03	2.34	0.18
平均错误率	4.35	15	4.75	51	7.53	>10
出错百分率/%	2.90	10.00	2.67	28.65	3.52	>50

5　结束语

本文提出了一种引入 K-means 作为预处理过程的改进熵蚁群算法(KIEAC),该方法通过 K-means 快速、粗略地确定聚类中心;通过最小信息熵对 K-means 得到的聚类中心过滤—重定位,在聚类个数不变的情况下得到更精确的类堆,然后用 EAC 算法进行二次聚类。该算法避免了 EAC 算法初始阶段学习缓慢的缺点,使得初始值的选择具有更多可参考的指导经验,同时减小了确定初始参数的盲目性。

参考文献:

[1] YANG Yan, CAMEL M. Clustering ensemble using swarm intelligence [C]//Proc of IEEE Swarm Intelligence Symposium. Piscataway: IEEE Service Center, 2003: 65-71.

[2] 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.

[3] WU Bin, SHI Zhong-zhi. A clustering algorithm based on swarm intelligence[C]//Proc of IEEE International Conference on Info-tech & Info-net. 2001: 58-66.

[4] 李雄飞,李军. 数据挖掘与知识发现[M]. 北京:高等教育出版社,2003:274-391.

[5] BARBARA D, COUTO J, LI Y. COOLCAT: an entropy-based algorithm for categorical clustering[C]//Proc of the 11th International Conference on Information and Knowledge Management. 2002: 1582-589.

[6] 颜宏文,马瑞,晏砺成. 基于信息熵构造判定树的数据挖掘算法的设计与实现[J]. 计算机工程与应用,2003,39(23):180-182.

[7] 严燕,卢宏涛. 基于信息熵的蚁群聚类改进方法研究[J]. 计算机仿真,2009,26(8):179-184.

[8] 邢洁清,朱庆生,郭平. 蚁群聚类组合方法的研究[J]. 计算机工程与应用,2009,45(18):146-148.

(上接第 1268 页)

参考文献:

[1] GOSAVI A. Reinforcement learning: a tutorial survey and recent advances[J]. INFORMS Journal on Computing, 2009, 21(2): 178-192.

[2] LIN C K. A reinforcement learning adaptive fuzzy controller for robots[J]. Fuzzy Sets and Systems, 2003, 137(3): 339-352.

[3] CAMPO I, ECHANOBE I, BOSQUE G, *et al*, Efficient hardware/software implementation of an adaptive neuro-fuzzy system[J]. IEEE Trans on Fuzzy Systems, 2008, 16(3): 761-778.

[4] ALAB E, DORRONSORO B. The exploration/exploitation tradeoff in dynamic cellar genetic algorithms[J]. IEEE Trans on Evolutionary Computation, 2005, 9(2): 126-143.

[5] TAN K C, CHIAM S C, MAMWN A A, *et al*. Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization[J]. European Journal of Operational Research, 2009, 197(2): 701-713.

[6] JOUFFE L. Fuzzy inference system learning by reinforcement methods[J]. IEEE Trans on Systems, Man and Cybernetics Part B, 1998, 28(3): 338-355.

[7] ER M J, DENG C. Online tuning of fuzzy inference systems using dymamic fuzzy Q-learning[J]. IEEE Trans on Systems, Man and Cybernetics Part B, 2004, 34(3): 1478-1489.

[8] VALI D, VAHID J M, MAJID N A. Fuzzy Sarsa learning and the proof of existence of its stationary points[J]. Asian Journal of Control, 2008, 10(5): 535-549.

[9] JUANG C F, HSU C H. Reinforcement interval type-2 fuzzy controller design by online rule generation and Q-value-aided ant colony optimization[J]. IEEE Trans on Systems, Man and Cybernetics Part B, 2009, 39(6): 1528-1542.