粒子群与细菌觅食相结合的案例聚类算法

胡爱策[1],任明仑[1,2],王 浩[1]
(1. 合肥工业大学 管理学院,安徽 合肥 230009;
2. 过程优化与智能决策教育部重点实验室,安徽 合肥 230009)

摘 要:案例聚类是按照案例库中案例的相似度进行归类,目的是减少案例推理系统搜索相似案例的时间、提高案例推理系统的性能和降低案例库维护的复杂度。该问题的难度在于案例库的案例规模比较大和不同的聚类算法的选择对于聚类结果的影响。文中在粒子群算法与细菌觅食算法基础上,将两者结合起来,综合两个算法的优点,并将其应用在 k-prototypes 方法上对案例库中案例进行聚类。与流行的聚类算法进行比较,实验结果显示文中的算法具有更高的效率并且性能相对而言更加优秀。

关键词:案例库;粒子群算法;细菌觅食算法;k-prototypes 算法
中图分类号:TP18 **文献标识码**:A **文章编号**:1673-629X(2013)10-0044-04
doi:10.3969/j.issn.1673-629X.2013.10.011

Case Clustering Algorithm Combining Particle Swarm Optimization and Bacterial Foraging

HU Ai-ce[1], REN Ming-lun[1,2], WANG Hao[1]
(1. College of Management, Hefei University of Technology, Hefei 230009, China;
2. Key Laboratory of Process Optimization and Intelligent Decision-makingof Ministry of Education, Hefei 230009, China)

Abstract: Case clustering is classified by the similarity to cases in case-base, the object is to reduce the time for searching similar case, improve the performance of case-base system and reduce the complexity of maintaining the case-base. The difficulty problem lies in that the size of case base is very large, and the clustering results is influenced by the choice of the clustering algorithm. In this paper, combined the advantages of particle swarm algorithm and bacterial foraging algorithm, use in case clustering with k-prototypes. Compared with popular clustering algorithm show that this algorithm is efficient, has better performance.

Key words: case base; particle swarm optimization; bacterial foraging algorithm; k-prototypes algorithm

0 引 言

基于案例的推理(Case-Based Reasoning,CBR)的思想源于对人类解决问题过程的模拟,利用案例库中的历史案例来求解新问题的类比推理过程。CBR 通过不断地向案例库添加新的案例,提高系统解决问题的能力。但随着案例库中案例增加到某一上限,继续增加案例将会导致 CBR 系统的性能和运行效率下降,即“沼泽问题”。为了有效地解决该问题,很多学者考虑案例维护来解决问题。David B. Leake 和 David C. Wilson 在文献[1]中提出了案例推理中案例维护的基本定义;Qiang Yang 和 Jing Wu 在文献[2]中将聚类与信息论结合起来应用到案例维护的过程中;Abir Smiti 和 Zied Elouedi[3] 介绍了案例维护中常见的几种方法;Rabia Ali 等[4] 提出使用聚类算法来控制案例库的删除;Shixia Ma 和 Shiyong Li 在文献[5]中提出粗糙集与案例聚类进行案例维护;耿焕同等[6] 具体介绍了案例聚类在案例维护中的应用;王薇薇等[7] 使用粒子群算法进行案例聚类。

案例聚类是将案例库中案例按照相互之间的相似度进行归类,使得同一类别间相似度尽量大,不同类别间相似度尽量小。由于案例中存在大量的符号型属性,其中有些符号型属性还是相对重要的属性,如:用

收稿日期:2013-01-08 **修回日期**:2013-04-14 **网络出版时间**:2013-07-24
基金项目:国家自然科学基金资助项目(71271073,70871032);教育部新世纪优秀人才支持计划(NCET-11-0625)
作者简介:胡爱策(1988-),男,硕士生,研究方向为管理人工智能;任明仑,教授,博士生导师,研究方向为面向服务架构、决策支持系统等。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20130724.1012.058.html

户的等级、使用的营销工具等,这些符号型属性在聚类过程中不能简单地去除,故文中使用能同时处理混合属性的 k-prototypes 聚类方法。

针对传统的 k-prototypes 方法具有稳定性差、聚类结果与初始中心的选择相关等缺陷,文中提出了将细菌觅食算法(BFA)和粒子群优化算法(PSO)结合起来应用在 k-prototypes 聚类方法上,从而弥补 k-prototypes 的缺陷。与流行的聚类方法进行了比较,实验结果显示该算法是一种高效且性能更优的聚类方法。

1　问题描述

传统的案例推理花太多的时间在比较新案例与案例库中的案例之间的相似性[8],为了避免这种情况,在原始方法的基础上采用案例聚类算法,按照案例之间相似度进行归类,使得案例库中各分组组内相似性和组间差异性达到最大,然后通过在分组中比较目标案件,找出最相似的案例。下面给出该问题的具体定义。

定义 1 设为 C 案例库,$\{c_1,c_2,\cdots,c_n\}$ 为案例库中所有案例的集合。$\{x_{i1},x_{i2},\cdots,x_{im}\}$ 表示第 $i(i\leqslant n)$ 个案例 c_i 的属性,其中 $x_{i1},x_{i2},\cdots,x_{ip}$ 表示数值型属性,$x_{i(p+1)},\cdots,x_m$ 为符号型属性,则案例的相似度可以用距离表示为:

$$d(c_i,c_j)=\sum_{k=1}^{p}w_k d(x_{ik},x_{jk})+r\sum_{k=p+1}^{m}w_k\partial(x_{ik},x_{jk}) \tag{1}$$

其中,w_k 表示案例第 k 个属性的权重;r 表示符号属性相对于数值属性的权值。

定义 2 若属性 x_{ik},x_{jk} 为数值型属性,则其相似度用曼哈坦距离[9] 表示为:

$$d(x_{ik},x_{jk})=(x_{ik}-x_{jk})^2 \tag{2}$$

符号属性一般分为分类属性、二元属性和序数型属性[10],它们的相似度可以按以下定义来计算。

定义3 如果属性 x_{ik},x_{jk} 为普通的文本属性,称为分类属性,其属性值没有顺序关系。针对曼哈坦距离不能直接应用在此属性上,Ralambondramy 提出一种将该类型转换成二进制属性的方法。在算法中,把这些二进制属性当成数值来处理,通过这种方法很容易描述分类属性的距离:

$$\partial(x_{ik},x_{jk})=\begin{cases}1 & \text{if } x_{ik}\neq x_{jk}\\ 0 & \text{if } x_{ik}=x_{jk}\end{cases} \tag{3}$$

定义 4 若属性只有两种状态,则可以转化为 0 和 1 表示,称为二元属性,则其距离按式(3) 计算。

定义 5 如果符号属性 x 具有 m 个可能的值,这些值之间有一定的顺序关系,例如网站卖家的星级,称为序数型属性[11]。则其相似度的计算步骤为:将属性的值映射为秩 $r,r\in\{1,2,\cdots,m\}$。用秩 r 代替属性的值,按公式 $z=\dfrac{r-1}{m-1}$ 将值映射到[0,1] 区间中,然后按式(2) 来计算属性相似度。

最后根据定义 2 到定义 5,并且结合公式(1) 计算得出两个案例之间的相似度。

定义 6 将案例库 C 分为 k 类,表示为 $\{s_1,\cdots,s_k\}$,其中 $C=\bigcup_{i=1}^{k}s_i$,cen_i 为第 i 个聚类的中心,$s_i\neq\varnothing$。则聚类问题的适应度函数 J 定义如下:

$$J=\sum_{i=1}^{k}\sum_{\forall c\in s_i}d(c,\mathrm{cen}_i) \tag{4}$$

如上所述,可以将聚类问题描述为:获得一组案例聚类的结果 $\{s_1,\cdots,s_k\}$,使得适应度函数 J 的值最小。

2　基于粒子群算法与细菌觅食优化相结合的案例聚类算法(KPBFO)

2.1　粒子群算法

粒子群算法(PSO)最初由 Kennedy 和 Eberhart 提出,是一种通过模拟鸟群的捕食行为求优化问题的群智能进化算法。在 PSO 算法中,首先初始化一群随机粒子作为初始随机解。然后通过迭代找到最优解,在每一次迭代中,每个粒子通过跟踪两个极值来更新自己位置。第一个就是个体极值 P_{id},表示粒子本身所找到的最优解。另一个极值是全局极值 P_{gd},表示整个种群目前找到的最优解。粒子根据如下公式更新速度和位置:

$$v_{\mathrm{id}}(t+1)=w*v_{\mathrm{id}}(t)+c_1*r_1*(p_{\mathrm{id}}-x_{\mathrm{id}}(t))+c_2*r_2*(p_{\mathrm{gd}}-x_{\mathrm{id}}(t)) \tag{5}$$

$$x_{\mathrm{id}}(t+1)=x_{\mathrm{id}}(t)+v_{\mathrm{id}}(t+1) \tag{6}$$

其中,c_1,c_2 是 2 个正常数,称为加速因子;r_1 和 r_2 是 2 个 0 到 1 的随机数;w 称为惯性权值,较大时算法具有较强的全局搜索能力,而较小时有利于算法收敛,如下表示:

$$w=w_{\max}-\frac{w_{\max}-w_{\min}}{t_{\max}}t \tag{7}$$

式中,t 为当前迭代次数;$t_{\max}$ 为最大迭代次数。

2.2　细菌觅食优化算法

细菌觅食算法(BFO)由 Passino 从大肠杆菌在人体肠道的觅食行为获取灵感提出的一种新的优化算法[12]。由于聚类可以被看作是一个函数优化的过程,所以 BFO 可以利用其全局搜索能力解决聚类问题。细菌觅食算法包括趋化、复制和迁徙三个步骤。一个细菌有两种移动方式:旋转和游动,在其整个生命周期,两种操作模式交替进行,这两种模式的交替行为称作是趋化操作。趋化操作是细菌觅食算法的核心组件。复制操作根据“优胜劣汰,适者生存”的原则,在

趋化操作完成后,以各细菌适应度值的累加和进行排序,较差的半数细菌被淘汰,复制较好的半数细菌,从而取代淘汰掉的细菌,这些细菌具有更强的搜索能力。为了减小算法陷入局部最小值的机会,BFO 引进了迁徙操作,细菌完成一定次数的复制操作后,以一定的概率将细菌随机迁徙到搜索空间中。

2.3 混合算法描述

粒子群算法与细菌觅食算法在优化问题中均体现了较好的性能,但由于各自特定的进化机制,也都存在各自的缺点。PSO 粒子在算法后期,由于缺乏有效的机制,容易陷入局部最优值;而 BFO 中细菌在趋化运动过程中运动具有随机性,收敛速度慢,没有充分利用其他细菌的信息。KPBFO 将两者相互结合,取长补短,提高了算法的性能和效率。

KPBFO 算法主要实现步骤如下:

第一步 载入相关数据,参数初始化。

设置相关参数,初始化细菌中的相关参数。

第二步 初始化细菌群。

对于细菌群中的每个细菌,根据给定的聚类数 k,任意选取 k 个案例 $\{c_1,\cdots,c_k\}$,作为细菌的初始位置。

第三步 趋化操作。

通过式(2) 计算并记录每个细菌的当前位置适应度值,并且按照适应度值更新每个细菌的个体最优位置 P_{id} 和细菌群的最优位置 P_{gd},记录细菌群最优位置的适应度值,然后按照公式(5)、(6) 计算细菌位置。判断趋化操作是否达到迭代次数,是则转入第四步,否则趋化次数加 1 并转入第三步。

第四步 复制操作。

对细菌的适应度进行累加和进行排序,根据“优胜劣汰,适者生存”的原则,淘汰掉一半适应度差的细菌,复制一半适应度好的细菌,以代替淘汰掉的细菌。判断复制操作是否达到迭代次数,是则转到第五步,否则复制次数加 1 并返回第三步。

第五步 迁徙操作。

细菌随机生成 0 到 1 的数,如果大于迁徙发生概率 p_{ed},则该细菌个体灭亡,重新选取 k 个案例作为该细菌的新位置。判断迁徙操作是否达到迭代次数,是则转入第六步,否则迁徙次数加 1 并返回第三步。

第六步 聚类操作。

将细菌群的最优细菌的位置作为 k-prototypes 算法中的聚类中心,运行 k-prototypes 算法,根据最终获得的聚类中心把所有的案例进行聚类,并将结果作为案例库最终的案例分类结果。

3 实验与分析

由于案例中不同属性的取值差距可能不同,使得差距大的属性值会对聚类的结果产生主要影响。为避免此情况的发生,应该在实验前进行标准化[13]。文中采用极大值标准化方法进行数据标准化:

$$x'_{ij}=\frac{x_{ij}}{\max\{x_{ij}\}}(i=1,2,\cdots,m;j=1,2,\cdots,n) \quad (8)$$

实验所采用的环境:Windows XP 操作系统,Matlab7.0 语言编程环境。

3.1 数据集测试

UCI 是数据挖掘中著名的数据集,实验中使用 UCI 中两个测试数据集。Iris 数据集包括 150 个实例,分成 3 个类,在实验中每个实例看作是一个案例。每个案例有 4 种数值属性。Soybean-small 数据集包含 47 个实例,分成 4 个类,共有 16 个符号属性。

通过实验比较,算法选择如下参数能获得较好的聚类结果,细菌个数为 50,趋化次数为 100,复制次数为 4,迁徙次数为 2,迁徙概率为 0.25,$c_1=c_2=2$,$w_{max}=0.9$,$w_{min}=0.2$。

实验采用相关算法运行十次得出的结果,然后进行平均计算得出结果(见表 1)。

表 1 算法的准确率比较

数据集	聚类算法	准确率/%	数据集	聚类算法	准确率/%
Iris	KPBFO	96.7	Soybean-small	KPBFO	97.96
	PSO	90.6		PSO	96.24
	BFO	92		BFO	96.52
	k- prototypes	78		k-prototypes	72.35

3.2 案例库测试

实验采用的案例集是某购物网站的用户信息,每个用户的相关信息作为一个案例。实验利用聚类有效性指标适应度函数来评价算法的性能。进行验证实验前需对案例库案例进行预处理,去除其中的缺失属性、冗余案例等,并按照式(8)对案例数值属性进行归一化处理,从而提高聚类操作的准确性。最终案例库拥有 1 500 个案例,每个案例拥有 23 个属性,其中数值属性 5 个,符号型属性 17 个。实验结果如表 2。

表 2 KPBFO 算法与相关算法的比较

聚类数目	k-prototypes		PSO		BFO		KPBFO	
	适应度	时间/s	适应度	时间/s	适应度	时间/s	适应度	时间/s
10	726.78	251	652.56	472	650.68	711	650.51	507
15	659.16	366	629.53	658	609.18	918	583.62	706
20	634.32	532	606.50	874	589.85	1 156	562.13	945

3.3 结果分析

由表 1 可以看出 KPBFO 聚类的准确性相比其他的算法要高。实验 2 中通过设置多个聚类数目值来考察聚类效果,表 2 总结了各个算法对于不同的聚类数目得到的聚类结果和运行时间,从表 2 可以看出 k-prototypes 算法虽然运算时间最短,但聚类结果最差。

总体而言,作为一个基于全局优化的聚类算法,KPBFO 比 BFO 和 PSO 更趋近全局最优解,显示出更好的收敛性。

4　结束语

文中给出了一个基于 BFO 与 PSO 结合的案例聚类算法,把聚类问题转化为通过优化目标函数来找到 k-prototypes 算法所需的最优初始聚类中心的问题。通过实验结果的比较,显示出 KPBFO 算法是一种高效的聚类方法,能够明显提高拥有混合属性的案例聚类的质量。下一步工作是将该聚类算法与 CBR 的维护过程结合起来,从而提高案例库维护的效率与性能。

参考文献:

[1] Leake D B, Wilson D C. Categorizing case-base maintenance: dimensions and directions[C]//Proceedings of the 1998 European Workshop on CBR (WECBR - 98). Berlin: Springer-Verlag, 1998.

[2] Yang Qiang, Wu Jing. Keep it simple: a case-base maintenance policy based on clustering and information theory[C]// Proc of 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence: Advances in Artificial Intelligence. Berlin: [s. n.], 2000: 102-114.

[3] Smiti A, Elouedi Z. Overview of maintenance for case based reasoning systems[J]. International Journal of Computer Applications, 2011, 32(2): 49-56.

[4] Ali R, Ather M, Ijaz R, et al. Clustering based deletion policy for case-base maintenance[C]//Proc of the 6th International Conference on Emerging Technologies (ICET). Islamabad: Springer-Verlag, 2010: 45-48.

[5] Ma Shixia, Li Shiyong. The case clustering algorithm based on rough set[C]//Proc of 2010 Second International Workshop on Education Technology and Computer Science. Wuhan: [s. n.], 2010: 300-303.

[6] 耿焕同,肖明军,邹　翔,等. 聚类算法在范例库维护中的应用研究[J]. 计算机工程,2005,31(12):166-168.

[7] 王薇薇,王清心,桑　海. 基于 tsPSO 的聚类案例检索策略[J]. 微型电脑应用,2011,27(9):63-64.

[8] Chang Pei-Chann, Lai Chien-Yuan. A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book forecasting[J]. Expert Systems with Applications, 2005, 29(1): 183-192.

[9] 张云涛,龚　玲. 数据挖掘原理与技术[M]. 北京:机械工业出版社,2003.

[10] 陈　韡. 基于划分的混合属性聚类算法研究[D]. 长沙:湖南大学,2010.

[11] 毛国君,段立娟,王　实,等. 数据挖掘原理与算法[M]. 第 2 版. 北京:清华大学出版社,2008.

[12] Wan Miao, Li Lixiang. Data clustering using bacterial foraging optimization[J]. Intell Inf Syst, 2012, 38(2): 321-341.

[13] 刘丽轻,丁巧林,张铁峰,等. 数据预处理方法对模糊 C 均值聚类的影响[J]. 电力科学与工程,2011,27(8):24-27.

(上接第 43 页)

结果验证了改进 R_EAODV 路由协议的正确性和有效性,特别是当节点能量不足时进行链路修复以发现新路由的机制,更适用于拓扑结构变化快的网络环境。

参考文献:

[1] Hong Li, Huang Tingpei, Zou Weixia, et al. Research of AODV routing protocol based on link availability prediction[J]. Journal on Communications, 2008, 29(7): 118-123.

[2] 陈稼婴,杨　震. Adhoc 网络中基于节能的 AODV 路由算法改进[J]. 南京邮电学院学报,2004,24(3):18-22.

[3] Razaa I, Hussain S A. Identification of malicious nodes in an AODV pure ad hoc network through guard nodes[J]. Computer Communications, 2008, 31(9): 1796-1802.

[4] Tseng Li-Pin, Yang Chun-Chuan. Fisheye zone routing protocol: A multi-level zone routing protocol for mobile ad hoc networks[J]. Computer Communication, 2007, 30(2): 261-268.

[5] 廖　登. 基于 NS2 的移动 Ad Hoc 网络典型网络协议比较[J]. 邵阳学院学报(自然科学版),2005,2(3):43-48.

[6] 王晓燕,郑明春. 基于 NS2 的网络仿真研究与应用[J]. 计算机仿真,2004,21(12):128-131.

[7] 詹鹏飞,陈前斌,李　云. 移动 AdHoc 网络 AODV 路由协议安全性分析和改进[J]. 计算机应用,2003,23(8):44-47.

[8] Rahman A H A, Zukarnain Z A. Performance comparison of AODV, DSDV and I-DSDV routing protocols in mobile ad hoc networks[J]. European Journal of Scientific Research, 2009, 32(4): 566-576.

[9] Chen Y, Yang H, Liu B, et al. Transmission power optimization algorithm in wireless ad hoc networks[C]//Proc of International Conference on Communications and Mobile Computing. [s. l.]: [s. n.], 2010: 358-363.

[10] 张晓辉,韩彬斌,王培康. 后备路径在自组网 AODV 协议中的应用[J]. 通信技术,2003,20(2):48-50.

[11] Chen Hongsong, Ji Zhenzhou, Hu Mingzeng, et al. Design and performance evaluation of a multi-agent-based dynamic lifetime security scheme for AODV routing protocol[J]. Journal of Network and Computer Applications, 2007, 30(1): 145-166.

[12] 臧婉瑜,于　勋,谢　立,等. 按需式 Ad Hoc 移动网络路由协议的研究进展[J]. 计算机学报,2002,25(10):1009-1017.

[13] 吴家皋,杨音颖,陈益新,等. 一种新的 QoS 覆盖多播路由协议研究[J]. 计算机学报,2006,26(11):1937-1946.

粒子群与细菌觅食相结合的案例聚类算法

万方数据 WANFANG DATA 文献链接

作者： 胡爱策， 任明仑， 王浩， HU Ai-ce， REN Ming-lun， WANG Hao

作者单位： 胡爱策,王浩,HU Ai-ce,WANG Hao(合肥工业大学 管理学院,安徽 合肥,230009)， 任明仑,REN Ming-lun(合肥工业大学 管理学院，安徽 合肥 230009; 过程优化与智能决策教育部重点实验室，安徽 合肥 230009)

刊名： 计算机技术与发展

ISTIC

英文刊名： Computer Technology and Development

年，卷(期)： 2013(10)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_wjfz201310011.aspx