基于快速蚁群的银行客户信息属性约简算法①

马胜蓝

(福建省农村信用社联合社科技服务中心, 福州 350001)

摘 要: 银行客户群体细分对于业务营销具有深远的意义, 客户信息具有数据量大、维度高、变化需求频繁的特点, 为此需要引入一种快速的属性约简算法, 以满足关键属性快速提取进而构建决策的要求. 本文通过改进传统的基于蚁群的属性约简算法, 优化每次迭代过程中的蚂蚁搜索的集合转移策略, 提出了一种基于快速蚁群算法的属性约简算法. 多个 UCI 数据集实验计算表明提出的新算法求解速度优于传统的基于蚁群算法的属性约简算法, 并且求解质量较优; 最后通过银行客户数据进行实践, 验证了该算法的可行性.

关键词: 客户信息; 粗糙集; 属性约简; 蚁群算法; 快速提取

Attribute Reduction of Bank Customer Information Algorithm Based on Quick Ant Colony Optimization

MA Sheng-Lan

(Science and Technology Service Center, Fujian Rural Credit Cooperatives, Fuzhou 350001, China)

Abstract: As bank customer segmentation has a profound significance for business marketing, while customer information has the characteristics of large amounts of data high dimensions and frequently-changing demand, we need to introduce a fast algorithm for attribute reduction to meet the needs of rapid attribute extraction to construct decisions. This paper proposes a new quick attribute reduction based on ant colony optimization by improving the collection for each iteration of the ant search transfer strategy. Numerical experiments on a number of UCI datasets show that the proposed new algorithm has a lower computational cost than the traditional ant colony-based attribute reduction algorithm and a better solution quality. Finally, the feasibility of the proposed algorithm is verified through the use of the bank customer data.

Key words: bank customer information; rough set; attribute reduction; ant colony algorithm; rapid attribute extraction

目前商业银行常用的客户分类方法是基于经验方法和统计方法的简单划分, 但是这些方法无法满足日益增长的数据量以及日益复杂的分析需求, 单是按照历史日均存款额度作为客户的细分属性, 就会造成客户的数据呈高维状态, 更何况现如今银行的客户数据都是千万级的, 数据的处理时效性也是需要重点考虑的[1,2]. 因此利用粗糙集能够保持客户数据本身的语义特征[3], 通过客户信息的约简就能很好地对客户进行分类. 由于求解信息系统最小约简是一个 NP 难问题, 若干学者也提出了不少最小属性约简算法[4–13]. 其中基于蚁群算法的属性约简算法搜索质量较好, 但是在每一次搜索迭代过程中, 每只蚂蚁都要不断逐个添加属性值, 如果条件属性非常大时, 属性选择过程将会非常耗时. 为此, 本文提出了一种基于快速蚁群算法[14][15]的属性约简算法, 通过优化蚁群的每次迭代的转移策略, 在每次迭代开始时仅删除最初始的移动集合元素, 并且让蚂蚁从当前的最新点继续搜索访问, 而不是从最初始的起点开始, 这样子就可以保证每只蚂蚁开始于不同的属性并且保留大部分的转移属性集合, 缩减蚂蚁的转移次数, 使之通过少量的迭代就可以获取到令人满意的属性约简结果, 满足时效性及高维数据约简的要求.

① 收稿时间:2015-02-04;收到修改稿时间:2015-04-02

本文后续部分的结构如下: 第二节简要介绍粗糙集理论; 第三节阐述基于快速蚁群算法的属性约简算法; 第四节结合 UCI 数据集进行实验结果比较, 并利用银行客户信息数据进行应用性验证.

1 基本概念和记号

为叙述方便起见, 首先简单回顾粗糙集理论的一些基本概念[16][17].

四元组 $S=(U,A,V,f)$ 是一个信息系统, 其中:

U: 对象的非空有限集合, 即论域;

A: 属性的非空有限集合, 信息系统中 A 常分为条件属性 C 和决策属性 D;

$V=\bigcup_{a\in A}V_a$, V_a 是属性的值域;

f: $U\times A\to V$ 是一个信息函数, 它为每个对象的每个属性赋予一个信息值.

设集合 $X\subseteq U$, P 是一个等价关系, 称 $\underline{P}X$ 为集合 X 的 P 下近似集; $\overline{P}X$ 为集合 X 的 P 上近似集;

$$\underline{P}X=\{x\mid x\in U,\text{且}[x]_P\subseteq X\}$$

$$\overline{P}X=\{x\mid x\in U,\text{且}[x]_P\cap X\neq\Phi\}.$$

设 P 和 Q 为论域上的等价关系, Q 的 P 正域记作 $\mathrm{POS}_P(Q)$:

$$\mathrm{POS}_P(Q)=\bigcup_{X\in U/Q}\underline{P}X$$

$$\gamma_{\mathrm{P}}(\mathrm{Q})=\frac{|\mathrm{POS_P(Q)}|}{|U|}$$

属性约简是去除一些冗余的属性而不减少原来集合的分类质量. 一个约简可以定义如下:

$$\mathrm{Red}=\{R\subseteq C\mid\gamma_R(D)=\gamma_C(D),\forall B\subset R,\gamma_B(D)\neq\gamma_C(D)\}$$

一个数据集可能有多个属性约简, 最小约简是指包含属性个数最少的一个属性约简.

2 基于快速蚁群算法的属性约简算法 O-AntRSAR

基于原始蚁群算法的属性约简算法(AntRSAR)[18]首先产生跟条件属性个数一样多的蚁群, 在每个条件属性处放置一只蚂蚁(该条件属性集合构成了一张图). 每只蚂蚁在每次迭代中根据当前走过的条件属性集合选择下一个条件属性, 一直到找到一个约简为止; 之后更新信息素浓度. 在下一轮循环中每只蚂蚁继续放置于不同的特征处, 重复计算约简. 对于第 k 只蚂蚁从特征点 i 移动到下一个特征点 j 的概率可以定义为:

$$p_{ij}^{k}(t)=\frac{\sup_{l\in tabuk}\{[\tau_{lj}(t)]^{\partial}[\eta_{lj}(t)]^{\beta}\}}{\sum_{s\in J_i^k}\sup_{l\in tabuk}\{[\tau_{ls}(t)]^{\partial}[\eta_{ls}(t)]^{\beta}\}}$$

J_i^k 为第 k 只蚂蚁未访问过的特征点, $J_i^k\supset C-R^k$; $\eta_{ij}(t)$ 为处于特征点 i 时选择特征点 j 的启发式知识; $\tau_{ij}(t)$ 为边(ij)的虚拟信息素:

$$\tau_{ij}(t+1)=\rho\tau_{ij}(t)+\Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t)=\begin{cases}1/|Rgb| & ij\in Rgb,i\neq j\\ 0 & others;\end{cases}$$

, Rgb 为到目前为止最优的约简.

基于原始的蚁群算法的属性约简算法在每一次迭代过程中, 每只蚂蚁都要不断逐个添加属性值; 如果条件属性 C 非常大时(例如银行客户属性信息), 属性选择过程将会非常耗时, 为此本文提出了一种基于快速蚁群算法的属性约简算法.

该优化的算法思路基于在第 t 次迭代结束后, 每只蚂蚁的移动集合应该为 $\{x_1,x_2,x_3,x_4,\ldots,x_n\}$, x_1 到 x_n 是随着蚂蚁的转移规则而排序, 显然 x_1 为蚂蚁的初始位置. 在原始的基于蚁群的属性约简算法中, 在 t+1 次迭代过程中, 蚂蚁的移动集合将会仅保留 x_1, 这就类似删除绝大多数蚂蚁的走过的集合, 但若仅删除 1–2 个移动集合元素, 而让蚂蚁从当前的最新步骤开始继续访问, 将可以缩减蚂蚁的转移次数. 由于是从蚂蚁的最新步骤开始继续访问, 这就类似蚂蚁是在不断的在寻找, 而不是从最初始的起点开始, 为此将该改进的算法命名为 O-AntRSAR. O-AntRSAR 在每一轮访问结束后删除第二个和第三个访问过的位置, 这样子可以保证每只蚂蚁开始于不同的属性并且保留大部分的转移属性集合(在 t+1 次迭代时蚂蚁的子集将为 $\{x1,x4,\ldots,xn\}$), 既保持着蚁群搜索的多样性能力, 又降低了转移开销.

O-AntRSAR 的流程图如图 1 所示.

3 实验

本章采用 UCI 数据集和银行客户信息验证基于快速蚁群算法的属性约简算法的可行性.

3.1 UCI 测试数据集

首先, 本节采用 11 个 UCI 数据集来测试提出的搜索机制的可行性. 具体的数据集信息如表 1 所示.

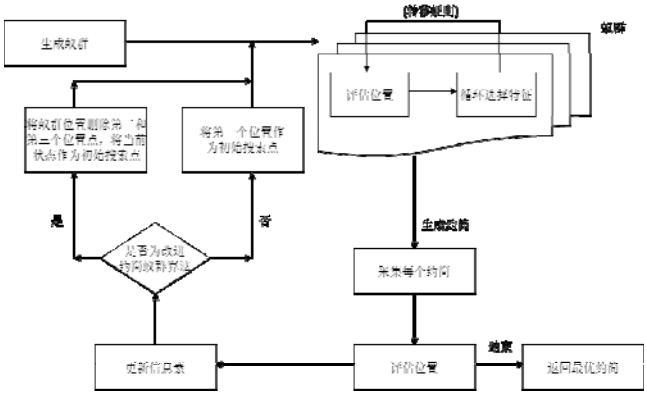

图 1 O-AntRSAR 算法流程图

表 1 UCI 数据集

数据集 ID	数据集	样本个数	条件属性个数
1	Bupa	345	6
2	Corral	32	6
3	Breast_cancer	191	9
4	DNA_nominal	2000	60
5	Led24	200	24
6	Mushroom	8124	22
7	Soybean_small	47	35
8	Soybean_large	307	35
9	Splice	2126	60
10	Vote	300	16
11	Balloon4	16	4

实验过程中, 每个算法在数据集上以不同的初始值计算 20 次. 本文比较 O-AntRSARAR 和其它 7 种属性约简算法(基于蚁群算法的属性约简算法(AntRSAR)[18]、基于粒子群(映射粒子速度)的属性约简算法(PSORSFS)[19]、基于标准二进制粒子群属性约简算法(PSOAR)[20]、基于禁忌的属性约简算法(TSAR)[21]、基于遗传优化的属性约简算法(GeneAR)和基于自适应变异的属性约简算法(AMPSOAR)[22]、利用 Hu[23]的算法求解的属性约简的长度)的求解质量和计算开销.

如下表 2 显示了 AntRSAR 和 O-AntRSAR 在 20 次迭代计算中的平均迭代次数和时间开销; 从这张表可以看出 O-AntRSAR 的搜索结果与 AntRSAR 相当, 在 11 个数据集上计算平均耗时均少于 AntRSAR, 在 DNA 数据集上耗时缩减 30%而搜索质量相当; 同时可以看出在 Led24 数据集上虽然 O-AntRSAR 的迭代次数比 AntRSAR 多, 但是整体耗时比 AntRSAR 低, 这也说明了 O-AntRSAR 的每次迭代搜索速度比 AntRSAR 快. 因此 O-AntRSAR 的计算开销远低于 AntRSAR 算法.

表 2 AntRSAR 和 O–AntRSAR 的计算开销比较

数据集 ID	AntRSAR			One-Generate AntRSAR		
	结果	平均迭代次数	平均耗时/s	结果	平均迭代次数	平均耗时/s
1	3^{20}	42.0	1.4002	3^{20}	42.05	1.1343
2	4^{20}	44.3	1.3198	4^{20}	43.35	1.1272
3	8^{20}	42.1	5.5982	8^{20}	42.05	3.4750
4	9^{20}	44.75	331.1909	9^{20}	44.65	255.5992
5	$11^{5}12^{15}$	53.65	40.8183	$11^{4}12^{16}$	62.75	27.2560
6	$4^{19}5^{1}$	55.1	329.1338	4^{20}	54.7	282.6530
7	2^{20}	44.95	13.2929	2^{20}	65	8.843
8	$9^{17}11^{3}$	75.25	108.1459	$9^{17}10^{1}12^{2}$	96.6	78.3955
9	9^{20}	47.75	397.1646	9^{20}	47.1	266.6779
10	8^{20}	65.25	26.8509	$8^{18}9^{2}$	79.55	18.5781
11	4^{20}	41.0	0.4574	4^{20}	41.0	0.4150

如下表 3 列出了比较的 9 种算法在 20 次实验过程中获得的最小约简的长度和数据集最小属性约简('Best')长度.

从这张表可以看出来 O–AntRSAR 在测试的 11 个数据集上都可以搜索到最优约简, 而 TSAAR、SPSOAR 在 Led24 上都无法搜索到最优约简, 因此 O–AntRSAR 搜索能力优于 TSAR、SPSOAR 和 Hu, 与 GameAR、PSORSFS、GeneAR、AMPSOAR 搜索效率相当. 因此本文提出的算法从计算开销和搜索效率上都能够满足实际要求.

3.2 银行客户数据属性约简

银行客户信息主要体现在客户的公共信息、财务

信息、电话信息、地址信息及历史财务信息等, 对于银行客户信息营销主要目标是在企业级的客户单一信息视图基础上, 着眼于富裕客户, 实现对公、对私业务的整体联动营销以及高端客户的获取机制, 提高理财经理对个人客户的管理能力, 提供差异化服务, 从而推动个人理财业务的发展; 实现对公客户营销方式由“粗放式”个体关系营销模式向“精细化”团队营销模式的转变, 提高对重点行业、重点客户的销售过程(商机)管理水平, 加强对客户经理的业绩统计与评估. 因此银行的客户信息维度是极其丰富的, 对于银行信息客户的细分就需要提取出来关键的属性.

表 3　八种属性约简算法约简结果比较

ID	约简长度		最小约简长度						
	Best	Hu	GameAR	PSORSFS	TSAR	SPSOAR	GeneAR	AMPSOAR	O-AntRSAR
Bupa	3	3	3	3	3	3	3	3	3
Corral	4	4	4	4	4	4	4	4	4
Breast_cancer	8	8	8	8	8	8	8	8	8
DNA_nominal	9	10	9	9	9	9	9	9	9
Led24	11	12	11	11	12	12	11	11	11
Mushroom	4	5	4	4	4	4	4	4	4
Soybean_small	2	2	2	2	2	2	2	2	2
Soybean_large	9	10	9	9	9	9	9	9	9
Splice	9	9	9	9	10	9	9	9	9
Vote	8	8	8	8	8	8	8	8	8
Balloon4	4	4	4	4	4	4	4	4	4

现对某农商行的客户信息建立条件属性和决策属性(共计 26 个条件属性), 如下表所示.

表 4　银行客户信息属性集合

	属性	备注
条件属性	客户类别	1– 个人 2– 对公 3– 金融(同业) 4–联名
	证件类别	
	国家代码	
	开户日期	
	社员(股东)标志	0–不是 1–是
	授信额度	
	公司类别	对公客户
	所有者性质	对公客户
	经济组织形式	对公客户
	经济类型	对公客户
	单位性质	对公客户
	免税标志	对公客户
	银企合作标志	对公客户
	注册币种	对公客户
	内部评估级别	级别分为 AAA、AA、A、 B、待评级
	贷款卡年审标志	0–未审 1–已审
	月均存款状态	0–高于同类 1–低于同类
	年均存款状态	0–高于同类 1–低于同类
	第一季度存款额度状态	0–高于同类 1–低于同类
	第二季度存款额度状态	0–高于同类 1–低于同类
	第三季度存款额度状态	0–高于同类 1–低于同类
	第四季度存款额度状态	0–高于同类 1–低于同类
决策属性	客户分级	1–高价值客户 2–一般客户 3–低价值客户 (该评定数据根据前期的客户评级结果导出)

采用 O-AntRSAR 对脱敏的部分存量客户数据(3.5 万条数, 对公和对私客户各抽取一半)进行属性约简, 得到属于约简集合 R 为“客户类别、国家代码、社员(股东)标志、公司类别、银企合作标志、内部评估级别、月均存款状态、年均存款状态”, 利用该 8 个条件属性就可以完成后续的客户分类规则计算. 同时, 这 8 个条件属性也从直观上也说明了属性约简的可行性, 例如股东标识为 1 并且客户类别是 2 时, 客户分级为 1–高价值客户.

4　结语

目前包括蚁群算法在内的诸多群智能算法都用于解决属性约简问题. 基于蚁群算法的属性约简算法具有很好的约简结果, 但是计算开销较大. 本文提出的

改进的基于蚁群算法的属性约简算法, 在蚁群的转移策略上优化了初始节点的集合, 降低了蚁群算法的搜索开销. 通过 11 个 UCI 数据集验证了本文提出的算法的可用性, 并且在银行客户数据约简上也做了应用上的验证. 然而, 基于蚁群算法仍然存在着不足: 在数据集具有非常大的条件属性条件下, 蚂蚁的群体个数将会变大.

参考文献

1 邵兵家.客户关系管理:理论与实践.北京:清华大学出版社, 2004

2 汤亚玲,黄华,程泽凯.基于自适应遗传神经网络的银行客户分类研究.计算机技术与发展,2014,24(7):192–195.

3 Jensen R, Shen Q. Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. IEEE Trans. on Knowledge and Data Engineering, 2004, 16(12): 1457–1471.

4 Wroblewski, J. Finding minimal reducts using genetic algorithms. Proc. of the Second Annual Join Conference on Information Sciences. Wrightsville Beach. NC. 1995. 186–189.

5 Jensen R, Shen Q. Finding rough set reducts with ant colony optimization. Proc. of the 2003 UK Workshop on Computational Intelligence. 2003. 15–22.

6 Hedar AR, Wang J, Fukushima M. Tabu search for attribute reduction in rough set theory. Springer-Verlag. Soft Comput, 2008, 12(9): 909–918.

7 Wang XY, Yang J, Peng NS, et al. Finding minimal rough set reducts with particle swarm optimization. Springer-Verlag Berlin Heidelberg. 2005. 451–460.

8 马胜蓝,叶东毅.一种带禁忌搜索的粒子并行子群最小约简算法.智能系统学报,2011,6(2):132–141.

9 马胜蓝,叶东毅.一种基于博弈策略的群智能属性约简算法.计算机工程与应用,2012,48(1):145–149.

10 马胜蓝,叶东毅.信息熵最小约简问题的若干随机优化算法研究.模式识别与人工智能,2012,25(1):96–104.

11 程美英,倪志伟,朱旭辉.基于生命周期的二元蚁群优化算法.模式识别与人工智能, 2014,27(11):1006–1014.

12 滕书华,鲁敏,杨阿锋,等.基于一般二元关系的粗糙集加权不确定性度量.计算机学报,2014,3:649–665.

13 韩素青,阴桂梅.一种面向用户需求的属性约简算法. 模式识别与人工智能,2014,27(3):281–288.

14 Maniezzo V, Colorni A. The ant system applied to the quadratic assignment problem. Knowledge and Data Engineering, 1999, 11(5): 769–778.

15 Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. IEEE Trans. on Systems, Man, and Cybernetics, Part B, 1996, 26(1): 29–41.

16 王国胤.Rough 集理论与知识获取.西安:西安交通大学出版社, 2001.

17 张文修,梁怡,吴伟志.信息系统与知识发现.北京:科学出版社,2003.

18 Deng TQ, Yang CD, Zhang YT, Wang XX. An Improved Ant Colony Optimization Applied to Attributes Reduction. German. Springer-Verlag Berlin Heidelberg. 2009. 1–6

19 Wang XY, Yang J, Peng NS, et al. Finding Minimal Rough Set Reducts with Particle Swarm Optimization. Springer-Verlag Berlin Heidelberg, 2005:451–460.

20 Kennedy J, Eberhart RC. A discrete binary version of the particle swarm algorithm. Proc. of the IEEE International Conference on Systems, Man and Cybernetics. Piscataway, USA. 1997. 4104–4109.

21 Hedar AR, Wang J, Fukushima M. Tabu search for attribute reduction in rough set theory. Springer-Verlag. Soft Comput, 2008, 12(9): 909–918.

22 吕振肃,候志荣.自适应变异的粒子群优化算法.电子学报, 2004,3:416–411.

23 Hu XH, Cercone N. Learning in relational databases: a rough set approach. Computational Intelligence, 1995, 11(2): 323–337.