基于蚁群算法的分类规则挖掘算法

吴正龙[1,2] 王儒敬[2] 滕明贵[2] 许梅生[1]
[1](解放军炮兵学院,合肥 230021)
[2](中国科学院合肥智能机械研究所,合肥 230021)
E-mail:zhenglongw@yahoo.com.cn

摘 要 提出了一种基于蚁群算法的分类规则挖掘算法。算法实质上是一种序列覆盖算法:蚁群搜索一个规则,移去它覆盖的样例,再重复这一过程,从而得到共同覆盖样例的一组规则。针对蚁群算法计算时间长的缺点,提出了一种变异算子。对两个公用数据的实验及其与 C4.5 和 Ant-Miner 的对比表明,算法能够发现更好的分类规则,包括预测能力更强,有更少规则的规则集,以及形式更简单的规则。实验同时显示变异算子有效节省了计算时间。

关键词 蚁群算法 分类规则 变异算子

文章编号 1002-8331-(2004)20-0030-04 **文献标识码** A **中图分类号** TP181;TP182

Mining Glassification Rule Based on Colony Algorithm

Wu Zhenglong[1,2] Wang Rujing[2] Teng Minggui[2] Xu Meisheng[1]
[1](Artillery Academy of PLA,Hefei 230021)
[2](Institute of Intelligent Machines,CAS,Hefei 230021)

Abstract: The paper proposes an algorithm,which is based on ant colony algorithm,for mining classification rule from categorical database.A mutation operator is applied to the algorithm for the purpose of shortening the computing time, which is usually too long in simple ant colony algorithm.Experiments on two public data set show that compared with C4.5,a famous decision tree learning algorithm,and Ant-Miner,the algorithm can discover better classification rule,including rule set with better predictive accuracy rate and fewer rule,simpler rule with fewer terms.Experiments also show that the mutation operator saves the computing time effectively.

Keywords: ant colony algorithm,classification rule,mutation operator

1 前言

分类是一项重要的数据挖掘任务。决策树学习能很好地发现分类规则。从树的根节点到叶节点的每一条路径表示一条分类规则,路径中的每个节点代表了对样例的某个属性的测试。在构造决策树的过程中,属性的选择至关重要。属性按照其单独分类样例集的能力大小被选择,而增加测试后产生的新样例集的纯度(purity)得以提高。

蚁群算法(Ant Colony Algorithm)是 20 世纪 90 年代初由意大利学者 M.Dorigo、V.Maniezzo、A.Colorni 等人提出并逐渐引起研究者的注意,被用于 TSP 问题求解、合作求解、网络路由[3]、分配问题[2]、数据聚类[6]、组合优化[5]等问题,取得了一系列较好的实验和应用结果。文献[10]提出基于蚁群算法的数据挖掘系统 Ant-Miner,它可以挖掘分类规则。虽然蚁群算法的研究刚刚起步,蚁群现象本身很多问题还缺乏严格的理论基础,但上述研究显示其在求解复杂优化问题,特别是离散优化问题上具有独特优越性,是一种很有发展前景的方法。

文章提出了一种基于蚁群算法的分类规则挖掘算法。算法模仿蚂蚁寻找食物的方式来构造分类规则。采用不同的启发式函数得到了比 C4.5 和 Ant-Miner 更好的结果,并且设计了变异算子,有效节省了计算时间。文章组织如下,第 2 节介绍了算法,第 3 节给出实验结果及与 C4.5 和 Ant-Miner 的详细对比,第 4 节总结并提出下一步的研究方向。

2 基于蚁群算法的分类规则挖掘算法

定义路径为属性节点和类标号节点的连线,其中每个属性节点最多只出现一次且必须有类标号节点。图 1 中给出了两个可能的路径。每个路径对应着一条分类规则,分类规则的挖掘可以看成对路径的搜索。利用蚂蚁寻找食物形成最短路径的原理,蚁群算法可以用来进行分类规则的挖掘,只不过这里搜索的不是最短路径,而是最优路径。最优路径表示了最优的分类规则。可以用路径对应规则的分类能力(有效性)和长短(简洁性)来衡量路径的优劣。

蚂蚁构造规则的过程体现为构造一条路径,分为三个阶段。首先从一条空路径开始重复选择路径节点增加到路径上(模仿蚂蚁的爬行过程)直到得到一条完整路径,也即一条分类规则;然后进行规则的剪枝,以考虑分类规则对样例的过度拟合问题;最后更新所有路径上的外激素浓度,对下一只蚂蚁构造规则施加影响(模拟蚂蚁间的信息交流)。

基金项目:国家自然科学基金重点项目(编号:69835001);国家 863 高技术研究发展计划重点项目(编号:2001AA115170)资助

作者简介:吴正龙(1976-),男,博士,研究方向为军事决策支持系统。王儒敬,男,副研究员,研究方向为知识工程与智能系统。滕明贵,男,博士生,研究方向为知识发现与智能系统。许梅生,男,教授,研究方向为军事决策支持系统。

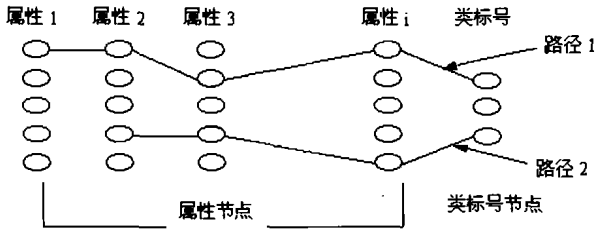

图 1 对应着分类规则的路径

2.1 构造规则

构造规则模仿了蚂蚁的爬行行为。蚂蚁重复选择节点直到构造一条完整路径。理论上节点的选择可以是完全随机的,但这可能需要漫长的计算时间作为代价。通常可以设计一个与问题相关的启发式函数,来引导蚁群的搜索。在这里,启发式函数被用来与外激素浓度一起,决定属性节点的选择。

Parpinelli,R.S 等[10]采用一种基于信息论的方法来构造启发式函数,每一个属性节点的启发式函数值与其分类样例的能力成正比。基于信息论的属性选择已经被证明倾向于选出取值多但每个取值内的例子不多的属性,这被称为算法的偏见(bias)。而在其后的增益比率属性选择[9]则可能会对属性选择造成误导,即选择被属性的信息熵决定,而不是被信息增益决定。这往往是错误的。为此 Mantaras 于 1991 年提出基于距离的属性选择法,有效克服以上问题,并且证明了基于距离的属性选择法,可以产生出比增益比率属性选择法更小的决策树。

这里根据基于距离的属性选择法来构造启发式函数,定义每个属性节点 $Term_{ij}$ 的启发式函数值 η_{ij} 为:

$$\eta_{ij}=\frac{|Term_{ij}|}{|Trainingset|}D_N(P_{A_i},P_C) \tag{1}$$

其中 $|Trainingset|$ 为训练集样例数;$|Term_{ij}|$ 为训练集中属性 i 取值为 $Term_{ij}$ 的样例数。对于一个训练集来说,在蚁群进化过程中,η_{ij} 并不改变。因此为了节省计算时间,这个值可以在蚁群搜索前进行计算。

当第一只蚂蚁开始构造路径时,所有路径节点的外激素浓度被初始化为相同的值:

$$\tau_{ij}=\frac{1}{\sum_{i=1}^{a}b_i} \tag{2}$$

其中 k 为类别数;a 为属性数(不包括类别属性);b_i 为属性 i 取值数。

赌轮选择机制[7]被用来选择属性节点。对于那些还没有出现在路径中的属性来说,其中属性节点 $Term_{ij}$ 被选择的概率由下式计算:

$$P_{ij}(t)=\frac{\tau_{ij}(t)\eta_{ij}}{\sum_{i}^{a}\sum_{j}^{b}\tau_{ij}(t)\eta_{ij}} \tag{3}$$

选择出的属性节点将被加入到路径中去。

当所有的属性都已包括在路径中,或者任一属性节点的增加将使得这个路径不能覆盖足够多的样例时,某个蚂蚁重复选择属性节点的过程结束。然后它将选择一个类标号节点,以形成一条完整路径也即一条分类规则,并且该规则的有效性最大。规则的有效性 Q 可以用下式来计算:

$$Q=\left(\frac{tp}{tp+fn}\right)\times\left(\frac{tn}{fp+tn}\right) \tag{4}$$

其中 tp 为规则前件后件都符合的样例数;fp 为符合规则前件不符合规则后件的样例数;fn 为不符合规则前件符合规则后件的样例数;tn 为规则前件后件都不符合的样例数。

注意上式中可能有未定义或不合理的情况出现,例如当 $fp+fn+tn=0$ 或 $fp+tn=0$ 时,上式未定义;而当 $tn+fn=0$ 时,根据上式计算,Q 为 0,而这显然不合理。因此对上式做如下扩展:

$$\begin{cases} Q=1; & when\ fp+fn+tn=0; \\ Q=\dfrac{tp}{tp+fn}; & when\ fp+tn=0; \\ Q=\dfrac{tp}{tp+fp}; & when\ tn+fn=0; \\ Q=\left(\dfrac{tp}{tp+fn}\right)\times\left(\dfrac{tn}{fp+tn}\right); & else \end{cases} \tag{5}$$

2.2 规则后剪枝

路径节点的重复选择可能会带来分类规则对样例的过度拟合,所以在规则产生之后对其进行规则剪枝。通过删除任何能导致规则精度提高的前件来修剪(泛化)规则。剪枝后的规则就是这个蚂蚁搜索到的规则。一种简单的剪枝策略就是,依次移去能使规则有效性得到最大提高的属性节点,直到任一属性节点的移去将降低规则有效性。当从规则中移去属性节点而使规则改变时,可能需要重新给它赋予一个类标号节点,以使规则的有效性仍然为最大。

```
procedure PruneRule;
begin
    Do{
        float max=0;
        int index=-1;
        for each term i in ruleA while term, is not the consequent{
            ruleB=ruleA|-term i;
            if ruleB.Q-ruleA.Q>=max
                index=i;
        }
        if index≠-1
            ruleA=ruleA|-term index;
        //maybe need to re-assign a class label to ruleA;
    }while(index=-1);
    return ruleA;
end;
```

其中 $ruleB=ruleA|_{-term\ i}$ 表示规则 B 由规则 A 移去第 i 个前件得到。

2.3 外激素浓度更新

当路径构造结束并经过规则剪枝得出一条分类规则后,所有路径节点的外激素浓度都将依据这条分类规则的效率被更新。规则中包含的路径节点的外激素浓度将被增加(外激素累积),而没有被包含的路径节点的外激素浓度将减小(外激素散发)。

规则的简洁性是衡量规则效率的重要指标。通常简洁的规则更易于理解和使用。将规则长度考虑到外激素浓度的更新中去,从而使得蚁群可以搜索出更简洁的规则。更简洁的规则对应着更短的路径。设规则长度为 n,则对包含在规则中的任一路径节点 $Term_{ij}$,有:

$$\tau_{ij}=\tau_{ij}+\tau_{ij}\times\left(1+\frac{1}{n\times Q}\right) \tag{6}$$

其中 Q 为规则的有效性。对于没有包含在规则中的节点

$Term_{ij}$,有:

$$\tau_{ij}=\frac{\tau_{ij}(t)}{\sum_{i}^{a}\sum_{j}^{b_i}\tau_{ij}} \tag{7}$$

这其实是一个归一化的过程。从式可以看出,包含在规则中的路径节点的外激素浓度增加了,因此,通过上式,没有包含在规则中的路径节点的外激素浓度减小了。

所有属性节点的外激素浓度将被更新,在此基础上,下一个蚂蚁开始它的搜索。当连续若干只蚂蚁搜索到同一条路径时,就认为搜索收敛;否则,这个重复的过程直到所有的蚂蚁搜索完。

每一个蚂蚁都构造出一条规则。这些规则当中最好的规则被最终认为是一条分类规则而保留,其他的规则被丢弃。训练集将根据这条规则进行修改。那些符合这条规则描述的样例被从训练集中除去,这实质上是一种序列覆盖算法(文献[10]同样如此)。这样得到新的训练集,开始蚁群的下一轮搜索。所有属性节点的外激素浓度此时被重新初始化。直到训练集样例数小于一个给定的值。对于每一个训练集都将得到一条最好规则,所有最好规则的集合就是最后对于初始训练集的分类规则集。

2.4 变异算子

蚁群算法具有较强的发现最优解的能力,不容易陷入局部最优。这是因为该算法不仅利用了正反馈原理,在一定程度上可以加快进化过程,而且是一种本质并行的算法,个体之间不断进行信息交流和传递,有利于发现较好解。但是该算法也存在一些缺陷。文献[2][10]指出其需要较长的搜索时间,但没有提出改善方法。主要原因是各个蚂蚁的运动是随机的,在进化的初始阶段,各个路径上的信息量相差不明显,当群体规模较大时,很难从较短的时间内从大量杂乱无章的路径中找到较好的路径,数据属性多时更是如此。只有经过较长一段时间,才能使得较好路径上的信息量明显高于其他路径上的信息量,随着这一过程的进行,差别愈来愈明显,从而得到较好的路径。这个过程一般需要较长的时间。

变异特征被引入到蚁群算法求解 TSP 问题中,并取得很好的效果[11]。受其启发,笔者在蚁群算法中设计变异算子,以提高某一代解(某一个蚂蚁搜索到的解)的质量,从而最终改善群体性能,减少计算时间。对于剪枝后的规则,进行单点变异,即随机选择一个变异位置,即某个属性节点,用这个属性节点对应属性的另一个属性节点取代原有属性节点从而构成新的规则,如果新规则有效性大于原有规则,则执行变异,否则取消变异。这个过程操作简单,只需很短时间,但却能够扩大搜索范围,提高单个解(修剪后的规则)的质量,从而加快算法收敛速度。变异算子的形式化描述如下:

```
procedure Mutation(rule A);
output:rule B;
begin
    Randomize a Term_ij in the rule A;
    ruleB=ruleA_{Term_ij ← Term_ik}  where k≠j;
    If B.Q>A.Q then mutation
        return rule B;
    else
        return rule A;
end;
```

其中 $ruleB=ruleA_{Term_{ij}\leftarrow Term_{ik}}$ 表示规则 B 由用 $Term_{ik}$ 取代规则 A 中的 $Term_{ij}$ 而得。

2.5 算法描述

下面是基于蚁群算法的分类规则挖掘算法形式化描述。

CRMBACA(Classification rule mining based on ant colony algorithm):

定义:

Uncovered-cases-threshold:允许剩余训练集样例数;

Rule-convergence-threshold:规则收敛值;

Min-cover-cases-num:规则最小覆盖值;

Number of ants:蚁群蚂蚁数;

Mutation rate:变异率;

Arule:一条规则;

PreviousRule:前一个蚂蚁搜索到的规则;

BestRule:最好规则;

RuleList:针对初始训练集的分类规则集。

算法:

```
While the number of cases in Training set> Uncovered-cases-threshold do
Begin
    int i=0;
    Repeat
        i=i+1;
        int j=0;
    Arule=ConstructRule(i);//第 i 只蚂蚁构造一条路径;
    PruneRule(Arule);//规则剪枝
    if mutation //controlled by mutation rate;
        Mutation(Arule);
    If Arule=PreviousRule j=j+1;
    else { PreviousRule=Arule;j=0;}
    if Arule is better than Bestrule
      Bestrule=Arule;
    UpdatePheromone(Arule);//依据规则更新外激素
  Until((i>Number of ants)or(j> Rule-convergence-threshold))
  remove cases coverd by Bestrule from training set;
  Add Bestrule to RuleList;//规则有序存储
end;
```

挖掘出的规则被有序的保存。规则将按照其被发现的次

表 1 公共数据集

数据集	样例数	属性数	类别数
Tic-tac-toe	958	9	2
Dermatology	366*	33	6

*:文献[10]中是 358,但这应该不会影响结果的对比

表 2 预测准确率

	预测准确率(%)		
数据集	C4.5	Ant-Miner	ACOBAFC
Tic-tac-toe	83.18±1.71	73.04±7.60	90.89±1.94
Dermatology	89.05±0.62	86.55±6.13	95.66±2.90

表 3 规则集及规则简洁性

	规则集规则数			规则集项数		
Data Set	C4.5	Ant-Miner	ACOBAFC	C4.5	Ant-Miner	ACOBAFC*
Tic-tac-toe	83.0±14.1	8.50±1.86	7.40±1.26	384.2±73.4	10.0±6.42	18±2.31
Dermatology	23.2±1.99	7.00±0.00	6.00±0.00	91.7±10.64	81.0±2.45	28±1.83

*:规则集所有规则前后件项数总和

表4 全部数据计算得出的规则集

Data set	Rule set
Tic-tac-toe	规则0:IF mm=x THEN C=p
	规则1:IF br=o THEN C=n
	规则2:IF bl=x THEN C=p
	规则3:IF tr=x THEN C=p
	规则4:IF br=x THEN C=n
	规则5:IF br=b THEN C=n (拟合率100%)
Dermatology	规则0:IF f_h=0 & spongiosis=0 & s_a_o_r=0 & p_p=0 & f_o_t_p_d=0 THEN Class=1
	规则1:IF f_o_t_p_d=0 & p_p=0 & v_a_d_o_b_l=0 & d_o_t_g_l=0 & k_p=0 THEN Class=2
	规则2:IF m_i=0 & o_m_i=0 & p_p=0 & f_o_t_p_d=0 THEN Class=4
	规则3:IF P_i=0 & s_a_o_r=0 & m_i=0 & p_p=0 THEN Class=5
	规则4:IF P_i=0 & p_p=0 THEN Class=3
	规则5:IF Age=young & k_p=0 THEN Class=6 (拟合率99.45%)

序,依次应用于新样例。一旦有规则可以对新样例作出预测,则预测过程结束。

3 实验

采用VC6.0实现算法。实验条件为Pentium Ⅳ 1.7G,128M内存,windows 2000 professional。由于算法目前只处理离散值数据,连续值的数据需要先进行离散化,所以只选择了两个离散值公共数据进行了实验,并与C4.5和Ant-Miner[10]进行了对比。数据集被随机分为10份,其中一份作为测试集,其余为训练集。每次取不同的测试集,计算10次,然后用平均值±标准误差的形式作为结果。算法参数设置分别为:Uncovered-cases-threshold=10;Rule-convergence-threshold=10;Min-cover-cases-num=10;Number of *ants*=3000;Mutation *rate*=0.2。

从表1、表2、表3可以看出,与C4.5和Ant-Miner相比,算法不仅能发现预测能力更强的规则集,而且在规则形式上也更简单,包括更少规则的规则集和更短的规则。这与启发式函数的选取有关系,同时也与将规则长度考虑到规则性能中去有关。而文献[10]由于简单选取属性熵信息作为启发式函数,不能克服基于信息论的属性选择的种种不足。因此,其预测准确率和C4.5相比,没有太大的提高,而其简单的规则集似乎也只是由于序列覆盖算法本身带来的。为了充分说明该算法可以得出简洁的规则集和规则,将所有数据作为训练集进行计算,分别得出如表4的结果。

可以看出,对于两个类别的Tic-tac-toe数据,算法得出六条规则,每条规则都只需要一个属性的测试;而对于六个类别的Dermatology数据,算法得出六条规则,每条规则对应一个类别的判定,而且最多也只需要5个属性的合取测试,这对于有着33个属性的数据来说,实在是相当简单的规则了。可以认为,在规则评价中考虑规则长度确实使得蚁群可以搜索到更短规则。

表5比较了有/无变异算子的计算时间。实验结果充分说明变异算子的设计有效的节省了计算时间。当数据集属性较多时,效果更为明显。

表5 有/无变异算子的计算时间(seconds)

Data Set	进行变异操作	不进行变异操作
Tic-tac-toe	20.7±3.59	24.3±2.16
Dermatology	158.23±69.68	748.4±337.37

4 结论与展望

该文提出了一种基于蚁群算法的分类规则挖掘算法。算法实质上是一种序列覆盖算法:蚁群搜索一个规则,移去它覆盖的样例,再重复这一过程。简单蚁群算法具有收敛慢、计算时间长的缺点,为此提出了一种变异算子。在规则评价中考虑了规则长度,从而使得蚁群可以搜索到更短的规则。对两个公用数据的实验及其与C4.5和Ant-Miner的对比表明,算法能够发现更好的分类规则,包括更强的预测能力,有更少规则的规则集以及更简单的规则;实验还显示变异算子有效的节省了计算时间。目前的算法只能处理离散值的数据,如何不需离散化就能直接处理连续值的数据,是下一步的研究目标。

(收稿日期:2004年4月)

参考文献

1.Colorni A,Dorigo M,Maniezzo V.Distributed optimization by ant colonies[C].In:Proc 1st European Conf Artificial Life,Pans,France:Elsevier,1991:134~142

2.Dorigo M,Maniezzo V,Colorni A.Ant System:An autocatalytic Optimizing process[R].Tech report 91-016,1991

3.Kwang Mong Sim,Weng Hong Sun.Multiple ant-colony optimization for network routing[C].In:2002 Proceedings,First International Symposium on Cyber Worlds,2002:277~281

4.R Lopez de Mantaras.A Distance--Based Attribute Selection Measure for Decision Tree Induction[J].Machine Learning,1991;6(1):81~92

5.刘道海,方毅,黄樟灿.一种求解组合优化问题的演化算法[J].武汉大学学报(理学版),2002;48(3):315~318

6.Monmarché N.On data clustering with artificial ants[C].In:AA Freitas ed.AAAI-99 & GECCO-99 Workshop on Data Mining with Evolutionary Algorithms:Research Directions,Orlando,Florida,1999:23~26

7.Z Michalewicz.Genetic Algorithms+Data Structures=Evolution Programs[M].2nd Edition,New York:Springer Verlag,USA,1994

8.Quinlan J R.Discovering rules by induction from large collections of examples[C].In:D Michie Ed.Expert systems in the micro electronic age,Edinburgh University Press

9.Quinlan J R.Induction of Decision Trees[J].Machine Learning,1986;1(1):81~106

10.Parpinelli R S,Lopes H S,Freitas A.A Data mining with an ant colony optimization algorithm[J].IEEE Transactions on Evolutionary Computation,2002;6(4):321~332

11.吴庆洪,张记会,徐心和.具有变异特征的蚁群算法[J].计算机研究与发展,1999;36(10):1240~1245