# 一种基于蚁群算法的分类规则挖掘算法

常晓磊,闫仁武

(江苏科技大学 电子信息学院,江苏 镇江 212003)

**摘 要**:Parepinelli等提出了基于ACO的分类算法。文中提出了一种基于自适应蚁群算法的分类规则挖掘算法,该算法采用了与Parepinelli算法不同的启发式函数及信息素改变方法,引入了自适应机制与变异策略,从而达到缩短蚁群算法计算时间、加快算法收敛速度、提高预测准确率的目的。实验结果验证了该算法的有效性。

**关键词**:蚁群算法;分类规则;自适应机制;变异策略

**中图分类号**:TP311;TP18 **文献标识码**:A **文章编号**:1673-629X(2007)07-0114-03

# An Improved Classification Rule Mining Based on Ant Colony Algorithm

CHANG Xiao-lei, YAN Ren-wu

(College of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

**Abstract**: Parepinelli proposed ACO classification algorithm. The paper proposes an improved classification rule mining based on ant colony algorithm. This algorithm uses new heuristic computation and pheromone update methods. Otherwise, an adaptive mechanism and a mutation strategy are applied to the algorithm for the purpose of shortening the computing time and improving the the accurate rate of prediction. The experiment result shows the validity of it.

**Key words**: ant colony algorithm; classification rule; adaptive mechanism; mutation strategy

## 0 引 言

蚁群算法是20世纪90年代初由意大利学者M. Dorigo等人提出并逐步引起研究者的注意,被用于求解TSP问题、调度问题、二次指派问题、组合优化等问题等,取得了较好的仿真试验结果。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[1]。从树的根节点到叶节点的每一条路径对应一条分类规则,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得以提高。文中提出了一种改进的基于蚁群算法的分类规则挖掘算法。采用了与文献[2]不同的启发式函数及信息素改变方法,引入了自适应机制与变异策略,从而大大缩短了算法计算时间,加快了算法收敛速度,提高了算法的预测准确率。

**收稿日期**:2006-10-25

**作者简介**:常晓磊(1983-),男,江苏镇江人,硕士研究生,研究方向为智能信息处理;闫仁武,副教授,研究方向为智能信息处理、数据挖掘。

## 1 改进的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。

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,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。

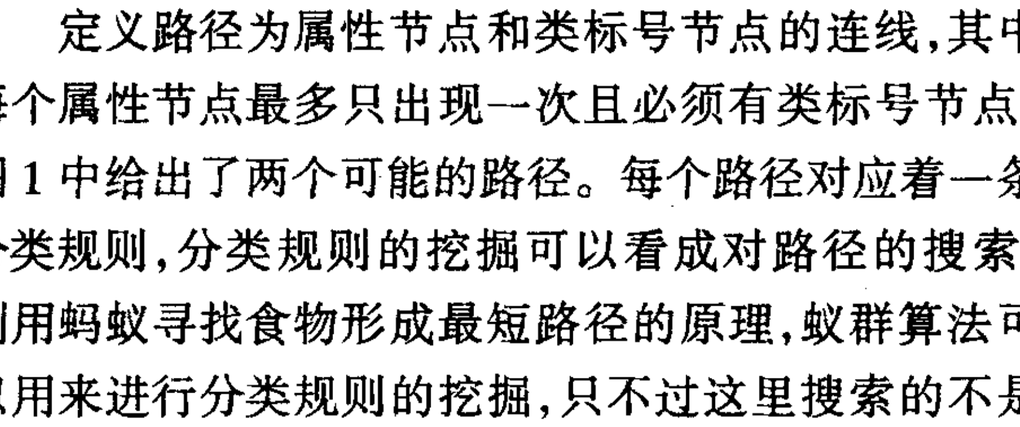


图1 对应着分类规则的路径[3]

### 1.1　规则构建

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。

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 $\mathrm{Term}_{ij}$ 的启发式函数值 $\eta_{ij}$ 为:

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

其中,$|\mathrm{Trainingset}|$ 为训练集样例数;$|\mathrm{Term}_{ij}|$ 为训练集中属性 $\mathrm{Term}_{ij}$ 取值为 $j$ 的样例数。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按(2)式被初始化为相同的值。

$$\tau_{ij}(t=0) = \frac{1}{\sum_{i=1}^{a} b_i} \tag{2}$$

式中,$a$ 表示数据库中属性总数;$b_i$ 表示属性 $i$ 所有可能取值的数目。

属性节点的选择根据赌轮选择机制进行,对于那些还没有出现在路径中的属性来说,其中属性节点 $\mathrm{Term}_{ij}$ 被添加到当前规则中的概率按(3)式计算。

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

当所有的属性都包含在路经当中,或者任一条属性节点的增加都将使得这个路径不能覆盖更多的样例时,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,形成一条完整的规则,并且该规则的有效性最大。规则的有效性按(4)式计算。

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

其中,$tp$ 为规则前件后件都符合的样例数;$fp$ 为符合规则前件但不符合规则后件的样例数;$fn$ 为符合规则后件但不符合规则前件的样例数;$tn$ 为规则前件后件都不符合的样例数。

### 1.2　规则修剪

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,故在规则产生之后对其进行规则剪枝。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。

### 1.3　信息素更新

当每只蚂蚁都构造一条规则后,这些规则中最好的规则被认为是一条分类规则而被保留,其它的分类规则被丢弃。然后,从所有训练集中移走符合此条规则条件的实例。这样,将得到新的训练集,开始蚂蚁的下一轮搜索。此时,所有条件项的信息素均被更新。对于包含在规则中的任一属性节点 $\mathrm{Term}_{ij}$,信息素浓度按(5)式[4]更新。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \tau_{ij}(t) * Q \tag{5}$$

其中,$p \in [0,1]$ 表示信息素挥发度,$1-p$ 表示信息素的挥发度残留度。对于没有被包含在规则中的属性节点 $\mathrm{Term}_{ij}$,信息素浓度按(6)式更新。

$$\tau_{ij}(t+1) = \frac{\tau_{ij}(t)}{\sum_{i}^{a}\sum_{j}^{b_i}\tau_{ij}(t)} \tag{6}$$

(6)式其实是一规范化过程,即对没有被包含在规则中的任一属性节点的 $\tau_{ij}(t)$,均除以所有 $\tau_{ij}(\forall i, j)$ 的和。

### 1.4　算法自适应机制及变异策略

为了加快进化过程,减少计算时间,有人将算法自适应机制及变异策略应用到蚁群算法求解问题当中,取得了较好的效果。受其启发,笔者也在蚁群算法中引入自适应机制及变异策略。其中,自适应机制包括两方面内容[5],即动态调节变异概率 $p$ 以及信息素残留度 $\rho$。该算法采取的自适应策略为:在 $n$ 只蚂蚁分别构建一条规则的过程中,如果连续有 $C$ 只蚂蚁搜索到同一条路径时,则以后的蚂蚁开始进行自适应调整,即增加 $p$ 和 $\rho$ 的值。

$$p(t) = p(t-1) + C_p, p_{\min} < p(t-1) < p_{\max}$$

其中 $p_{\min}, p_{\max}$ 是变异概率 $p$ 允许的极小、极大值;

$$\rho(t) = \rho(t-1) + C_\rho, \rho_{\min} < \rho(t-1) < \rho_{\max}$$

其中 $\rho_{\min}, \rho_{\max}$ 是信息素残留度的极小、极大值。其变异思想为:对于修剪后的规则,根据变异概率 $p$ 进行单点变异,即随机选择一个变异位置(即某个属性节点),利用这属性节点对应的属性的另外一个属性节点来取代原有的属性节点,从而构成新的规则。如果新规则的有效性大于原来的规则,则进行变异,否则不进行变异。此过程操作简单,只需要很短的时间,但却能扩大搜索范围,提高修剪后的规则质量,从而可以加快算法收敛速度。

### 1.5　算法描述

定义:

max_uncovered_case:终止条件,即训练集中最多剩下的实例数;

Number_of_Agents:Agent 的数目;

Number_Rules_Coverg:允许规则收敛值,即收敛到同一条规则的蚂蚁最大数目;

Arule:一条规则;

PreviousRule:前一个 agent 搜索到的规则;

Bestrule:最好规则;

算法:

```
TrainingSet = {all training cases}
DiscoveredRuleList = [];//用空集合初始化规则集
WHILE(Training set > max_uncovered_case)
Begin
    初始化信息素;
    计算启发式函数值;
    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); //规则修剪;
    If mutation //是否变异取决于变异概率 p;
      Mutation(Arule);
    If Arule=PreviousRule j=j+1;
    Else {PreviousRule=Arule;j=0;}
    If Arule is better than Bestrule
      Bestrule=Arule;
    If j>=C then //进行局部自适应动态调整信息素及变异概率
      Begin //一般取 C< Number_Rules_Coverg
        If pmin<p(t-1)<pmax then p(t)=p(t-1)+Cp,
        If ρmin<ρ(t-1)<ρmax then ρ(t)=ρ(t-1)+Cρ,
      End if
    UpdatePheromone(Arule);//进行全局信息素更新
  Until((i> Number_of_Agents) or (j> Number_Rules_Coverg))
  Remove cases coverd by Bestrule from TrainingSet;
  Add BestRule to DiscoveredRuleList;//规则有序存储
End;
```

## 2 实 验

从 UCI 公共数据库中选择两组数据集(Breast cancer 和 Tic_tac_toe)如表 1 所示,每个数据集被分为 10 份,选择 1 份作为测试集,而另外 9 份作为训练集。实验参数取值为:

max_uncovered_case = 10; Number_of_Agents = 3000; Number_Rules_Coverg = 10; $p(0) = 0.2$; $\rho(0) = 0.2$; $C = 7$。

表 1 公关数据集[4]

| 数据集 | 样例数 | 属性数 | 类别数 |
|---|---|---|---|
| Breast cancer | 358 | 33 | 6 |
| Tic-tac-toe | 958 | 9 | 2 |

模拟实验通过使用训练集产生的规则对测试集进行分类,实验结果如表 2 所示:Breast cancer 数据集的分类准确率为 93%,Tic_tac_toe 数据集的分类准确率为 85%,试验结果明显优于文献[2]与 C4.5。表 3 比较了有/无变异策略的计算时间,实验结果充分说明变异策略有效地节省了计算时间。当数据集属性较多时,效果会更明显。

表 2 预测准确率

| | 预测准确率 | | |
|---|---|---|---|
| 数据集 | 文献[2]算法 | C4.5 | 文中算法 |
| Breast cancer | 85.67% | 89.05 | 93.50% |
| Tic-tac-toe | 73.04% | 83.18% | 85.36% |

表 3 有/无变异策略的计算时间(s)

| 数据集 | 进行变异 | 不进行变异 |
|---|---|---|
| Breast cancer | 125.58 | 298.28 |
| Tic-tac-toe | 17.35 | 23.76 |

## 3 结 论

蚁群优化是人工智能领域中群体智能分支之一,已经成功应用于 TSP、作业调度、路由选择等问题上,但用它解决数据挖掘问题还是一个新的研究课题。文中将蚁群优化算法应用到数据挖掘当中,提出了一种改进的基于蚁群算法的分类规则挖掘算法。该算法采用了与文献[2]不同的启发式函数及信息素改变方法,引入了自适应机制与变异策略,从而达到缩短蚁群算法计算时间、加快算法收敛速度、提高预测准确率的目的。

**参考文献:**


[1] 张惟皎,刘春煌,尹晓峰.蚁群算法在数据挖掘中的应用研究[J].计算机工程与应用,2004(28):171-173.

[2] Parepinelli R S, Lopes H S, Freitas A. An Ant Colony Algorithm for Classification Rule Discovery[C]//In H. A. a. R. S. a. C. Newton. Data Mining Heuristic Approach. [s. l.]: Idea Group publishing, 2002.

[3] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computing, 1997, 1(1): 53-56.

[4] 朱庆保,杨志军.基于变异和动态信息素更新的蚁群优化算法[J].软件学报,2004,15(2):185-192.

[5] 李 薇,张凤鸣.多 Agent 技术研究与应用[J].微计算机信息,2006,8(3):293-295.