

多 Agent 信息系统知识发现研究^{*}

王 勇^{1,2} 黄国兴¹ 王 雨³ 彭道刚²

(华东师范大学 上海 200062)¹ (上海电力学院计算机科学与技术系 上海 200090)²

(中国国际人才交流协会 北京 100873)³

摘要 本文根据信息系统的定义,提出了采用集合方法描述的多 Agent 信息系统,研究了在多 Agent 交互中规则的形成过程。对规则表示和存储方法进行了研究。鉴于各 Agent 的规则可存储在数据库中,采用 ID3 算法对这些规则进行挖掘是可行的。对该方法用实例进行了验证。

关键词 Agent, 知识发现, 信息系统

Knowledge Discovery Research on Multi-agent Information System

WANG Yong^{1,2} HUANG Guo-Xing¹ WANG Yu³ PENG Dao-Gang²

(East China Normal University, Shanghai 200062)¹

(Dept. of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090)²

(China Association for International Exchange of Personnel, Beijing 100873)³

Abstract This paper presents a Multi-Agent information system of collection set according to information system definition. We research into the procedure of generating rules during the Multi-Agent system. Logical structure and physical structure of rules are discussed in the paper. In respect that Agent rules can be stored in database, we mine the rules using ID3 algorithm and verify the algorithm by the instance.

Keywords Agent, Knowledge discovery, Information system

1 引言

在 AI 领域,许多学者对 Agent 的认知系统的逻辑结构进行了研究,较有影响的是 BDI-Agent 模型^[1~3]。在 AI 领域对知识的复杂性、时间序列相关运动理论和协议的逻辑结构方面进行了探索,此方向的研究拓宽了 Agent 的逻辑结构研究领域^[4~6]。智能理论应用到了模糊识别系统、决策支持系统中^[7,8],决策树算法在决策支持领域有着广泛的应用,Agent 的计算方法的理论研究成果可以应用到决策支持领域^[8,9]。

Quinlan 的 ID3 算法是有影响的决策树算法,它是基于信息熵,根据属性集的取值进行分类的。由于 Agent 的 BDI 的逻辑结构包括属性特征,故出现了采用 ID3 算法挖掘 Agent 知识方面的研究^[10~13]。

Agent 的知识可以采用链表的方法表示,为每一个链表上的连接赋予相应的权值,构建认知网络。在此链式存储结构下,采用基于信息熵的 ID3 算法进行挖掘,其效率会比较低。Agent 知识也可以采用关系数据库的方式表示,Agent 的内部状态信息、外部状态信息和知识库可存储在数据库中,采用决策树方法可以非常方便地对关系数据进行挖掘。由于关系数据库的集合表示性质,Agent 信息系统采用集合表示方法。

2 多 Agent 信息系统的集合表示

2.1 信息系统定义

在集合概念的基础上,采用三元组表示信息系统,其定义为:称 $(UA, Attr, Func)$ 是一个信息系统,其中 UA 为对象集合,即 $UA = \{x_1, x_2, \dots, x_n\}$, UA 中的每一个元素 $x_i (i \leq n)$ 被称为一个对象。 $Attr$ 为属性集合,即 $Attr = \{a_1, a_2, \dots, a_m\}$, $Attr$ 属性集合中的每一个元素为 $a_l (l \leq m)$ 被称为一个属性。 $Func$ 为 UA 和 $Attr$ 之间的关系集合,即 $Func = \{f_l : UA \rightarrow V_l (l \leq m)\}$, 其中 V_l 为 $a_l (l \leq m)$ 的值域。

2.2 多 Agent 信息系统的集合表示

在信息系统定义的基础上,根据 Agent 的特征,把三元组扩展成为五元组,给出多 Agent 信息系统的集合定义:称 (U, S, E, R, T) 是一个信息系统,其中 U 为 Agent 集合, U 中的每一个元素 $Agent_i (i \leq n)$ 称为一个 Agent,即

$$U = \{Agent_1, Agent_2, \dots, Agent_n\}$$

S 是 $Agent_i (i \leq n)$ 内部状态属性,是 $Agent_i (i \leq n)$ 的局部变量,可以采用集合的方法表示这些状态属性,即

$$S = \{InternalSA_1, InternalSA_2, \dots, InternalSA_m\}$$

E 是 $Agent_i (i \leq n)$ 感知外部环境状态的属性,是感知器的输入接口参数的类型,可以采用集合表示,即

$$E = \{ExternalSA_1, ExternalSA_2, \dots, ExternalSA_m\}$$

R 是 $Agent_i (i \leq n)$ 返回的状态属性,是输出参数的类型,这些属性作为其他 Agent 的输入,集合表示为

$$R = \{ReturnSA_1, ReturnSA_2, \dots, ReturnSA_m\}$$

T 是 $Agent_i (i \leq n)$ 的任务集合, $Agent_i (i \leq n)$ 个体具备改变外界环境和自身的能力,其不仅可以改变内部状态属性 S 的值,而且可以改变返回的状态属性 R 的值, F_s 表示内部状

^{*} 本文受到华东师范大学 211 重点项目基金(521B0108)、上海市重点学科建设项目基金(P1303)资助。王 勇 博士生,从事 Agent 的研究;黄国兴 教授,博士生导师。

态属性所对应的函数, F_R 表示返回状态属性对应的函数, V_i 表示值, 即

$$T = \{F_S; InternalSA_i \rightarrow V_i; F_R; ReturnA_i \rightarrow V_i\}$$

2.3 多 Agent 信息系统实例

假设 (U, S, E, R, T) 是一个 Agent 的信息系统, 其中 $U = \{Agent_1, Agent_2, \dots, Agent_8\}$, S 是 $Agent_i (i \leq 8)$ 内部状态属性, 可以采用集合的表示方法表示其内部状态, $S = \{InternalSA_1, InternalSA_2, InternalSA_3\}$ 。

E 是 $Agent_i (i \leq 8)$ 感知外部环境状态的属性, 集合表示为 $E = \{ExternalSA_1, ExternalSA_2\}$ 。

R 是 $Agent_i (i \leq 8)$ 返回的状态属性, 用集合可以表示为 $R = \{ReturnSA_1\}$ 。

T 是 $Agent_i (i \leq 10)$ 的任务标识集合, 仅修改 R 的返回值。初始状态值如表 1 所示。

表 1 $Agent_i (i \leq 8)$ 的信息系统

| U | S1 | S2 | S3 | E1 | E2 | R1 | T |
|--------|----|----|----|----|----|----|---|
| Agent1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Agent2 | 0 | 0 | 1 | 0 | 1 | 1 | |
| Agent3 | 0 | 1 | 0 | 1 | 0 | 0 | |
| Agent4 | 0 | 1 | 1 | 1 | 1 | 1 | |
| Agent5 | 1 | 0 | 0 | 0 | 0 | 0 | |
| Agent6 | 1 | 0 | 1 | 0 | 1 | 1 | |
| Agent7 | 1 | 1 | 0 | 1 | 0 | 0 | |
| Agent8 | 1 | 1 | 1 | 1 | 1 | 1 | |

Agent1 的规则: $Agent1Rule1 = S\{0,0,0\}E\{0,0\}R\{0\}$;
 Agent2 的规则: $Agent2Rule1 = S\{0,0,1\}E\{0,1\}R\{1\}$;
 Agent3 的规则: $Agent3Rule1 = S\{0,1,0\}E\{1,0\}R\{0\}$;
 Agent4 的规则: $Agent4Rule1 = S\{0,1,1\}E\{1,1\}R\{1\}$;
 Agent5 的规则: $Agent5Rule1 = S\{1,0,0\}E\{0,0\}R\{0\}$;
 Agent6 的规则: $Agent6Rule1 = S\{1,0,1\}E\{0,1\}R\{1\}$;
 Agent7 的规则: $Agent7Rule1 = S\{1,1,0\}E\{1,0\}R\{0\}$;
 Agent8 的规则: $Agent8Rule1 = S\{1,1,1\}E\{1,1\}R\{1\}$ 。
 假设 $Agent_i (i \leq 8)$ 之间信息交互的结构如图 1 所示。

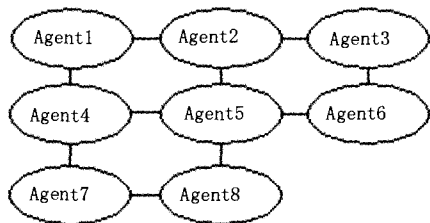


图 1 $Agent_i (i \leq 8)$ 的邻接图

规定相邻的 Agent 可以信息交互, 由此得到邻接矩阵, 如表 2 所示。

表 2 $Agent_i (i \leq 8)$ 的邻接矩阵

| | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
|----|----|----|----|----|----|----|----|----|
| A1 | | 1 | | 1 | | | | |
| A2 | 1 | | 1 | | 1 | | | |
| A3 | | 1 | | | | 1 | | |
| A4 | 1 | | | | 1 | | 1 | |
| A5 | | 1 | | 1 | | 1 | | 1 |
| A6 | | | 1 | | 1 | | | |
| A7 | | | | 1 | 1 | | | |
| A8 | | | | | 1 | | 1 | |

假设 $Agent_i (i \leq 8)$ 的内部环境变量 $S1, S2, S3$ 均保持不变, 初始状态下 $E2=1$ 。把目标 $T\{R1=0\}$ 作为 Agent2 的 $E1$ 环境输入值。Agent1 会影响到其邻接的 Agent2 和 Agent4, 若输入符合 Agent 规则要求, 就有相应的返回值, 否则无法影响到该 Agent。

查找邻接矩阵得到 A1 的邻接为 A2, A4。已知 $\{E1=0, E2=1\}$, 根据 Agent2Rule1 其返回值 $R1=1$; $\{E1=0, E2=1\}$ 根据 Agent4Rule1, 由于 $E2=1$, 不符合规则要求, 则 Agent4 对应 Agent1 的输入信息没有反应。Agent2 的返回值为 1, $E2=1$, 与其相邻的 $\{Agent1, Agent3, Agent5\}$ 的规则中没有匹配值, 所以 Agent1, Agent3, Agent5 均不受到影响, 不受影响的集合为 $\{Agent1, Agent3, Agent4, Agent5\}$, 该集合的邻接 Agent 均不受到影响, 最终只有 Agent2 受到了影响, 如表 3 所示。

表 3 $Agent_i (i \leq 8)$ 的信息交互表

| U | S1 | S2 | S3 | E1 | E2=1 | R1 | T |
|--------|----|----|----|----|------|----|-----------------------------------|
| Agent1 | 0 | 0 | 0 | 0 | 0 | 0 | $R1=0$, 未变化 |
| Agent2 | 0 | 0 | 1 | 0 | 1 | 1 | $\{E1=0, E2=1\} \rightarrow R1=1$ |
| Agent3 | 0 | 1 | 0 | 1 | 0 | 0 | 未变化 |
| Agent4 | 0 | 1 | 1 | 1 | 1 | 1 | 未变化 |
| Agent5 | 1 | 0 | 0 | 0 | 0 | 0 | 未变化 |
| Agent6 | 1 | 0 | 1 | 0 | 1 | 1 | 未变化 |
| Agent7 | 1 | 1 | 0 | 1 | 0 | 0 | 未变化 |
| Agent8 | 1 | 1 | 1 | 1 | 1 | 1 | 未变化 |

3 多 Agent 规则的存储结构

3.1 多 Agent 规则的顺序储存结构

多 Agent 规则依次记录在规则表中, 存储在关系数据库中。除首记录外, 每一个记录具有唯一的直接前驱。除尾记录外, 每一个记录具有唯一的直接后继, 这种存储结构就称为顺序储存。顺序存储结构具备操作简单, 通用性强的特点。

假设 (U, S, E, R, T) 是一个 Agent 的信息系统, 其中 $U = (Agent_1)$, S 是 $Agent_1$ 内部状态属性, 用集合可以表示为 $S = \{InternalSA_1, InternalSA_2, InternalSA_3\}$; R 是 $Agent_1$ 感知外部环境状态的属性, 采用集合可以表示为 $E = \{ReturnSA_1\}$ 。 R 是 $Agent_1$ 返回的状态属性, $R = \{ReturnSA_1\}$ 。表 4 列出了 8 条规则, 根据 $RuleNumber = 2^3 * 2^2 * 2^1 = 64$, 还有 56 条规则没有列出。根据规则表所列出的规则, 我们可以采用逻辑函数表示该表。

表 4 $Agent_1$ 的规则表

| U | S1 | S2 | S3 | E1 | E2=1 | R1 |
|-------|----|----|----|----|------|----|
| Rule1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Rule2 | 0 | 0 | 1 | 0 | 1 | 1 |
| Rule3 | | 1 | 0 | 1 | 0 | 0 |
| Rule4 | 0 | 1 | 1 | 1 | 1 | 1 |
| Rule5 | 1 | 0 | 0 | 0 | 0 | 0 |
| Rule6 | 1 | 0 | 1 | 0 | 1 | 1 |
| Rule7 | 1 | 1 | 0 | 1 | 0 | 0 |
| Rule8 | 1 | 1 | 1 | 1 | 1 | 1 |

在 $Agent_1$ 的规则表中, 挑选那些使 $R1$ 为 1 的规则, 对应选中的规则, 若其内部状态属性和可感知的外部状态属性值为 1 的, 写出其属性名, 否则写出其反属性名。把这一规则的属性名相乘, $R1=1$ 的规则相加, 就得到了规则函数。规则函

数如下所示:

$$R = \overline{S1S2S3E1E2} + \overline{S1S2S3E1E2} + S1 \overline{S2S3E1E2} + S1S2S3E1E2$$

3.2 多 Agent 规则的链式存储结构

Agent 规则采用顺序存储或者逻辑函数进行表示时,规则包括内部状态属性、感受的外部环境、输出的结果。若取值范围是{0,1}, RuleNumber = 2^E * 2^E * 2^R。当取值范围增多时,规则按照指数级别增长,规则表内记录快速增加。链式存储的方法是把规则函数的属性值直接写入到结点中,如图 2 所示。

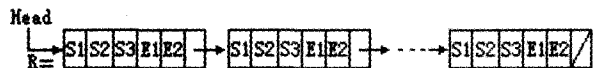


图 2 Agent 规则的链式存储

在此链式存储结构下编程,需要对链表的结点进行定义,实现 Agent 规则的结构体定义形式如下所示:

```
#define BLOCKSIZE 5
typedef struct LogicFunction{
    int Value[BLOCKSIZE];
    struct LogicFunction * next;
}LinkLogicFunction;
typedef struct{
    LinkLogicFunction * head;
    int curLength;
}AgentRule;
```

由于 ID3 算法是基于信息熵,根据属性集的取值进行分类的,因此构建的 Agent 信息系统中,选用顺序存储结构比较利于 ID3 算法发现 Agent 信息系统中的知识。

4 采用 ID3 算法发现多 Agent 信息系统中的知识

4.1 ID3 算法

决策树(Decision Tree)是运用于分类的一种树结构。其中的每个内部节点代表对某个属性的一次测试,一条边代表一个测试结果,叶子代表某个类或者类的分布,最上面的节点是根结点。

J. R. Quinlan 的 ID3 算法是有影响的决策树算法,它基于信息熵,根据属性集的取值进行分类。ID3 采用自顶向下不回溯的策略搜索全部的属性空间,该算法简单,深度小,分类速度快。但是 ID3 对于大的属性集则执行效率下降快,准确性降低,并且学习能力低下。

ID3 算法的工作过程:首先找出具有判别力的属性,把数据集分成若干个子集,每个集合然后再选出具有判别力的属性进行划分,一直进行到所有子集仅仅包含同一类型的数据为止,最后根据得到的决策树对实例进行分类。

ID3 算法原理:设 $E = \{V_1, V_2, \dots, V_m\}$ 是 m 维有穷向量空间,其中 V_i 是有穷离散符号集, E 中的元素 $e = \langle E_1, E_2, \dots, E_n \rangle$ 称为实例。其中 $E_i \in F_i, i=1, 2, \dots, n$ 。设 Pe 和 Ne 是 E 的 2 个实例集,分别称为正例集和反例集。

假设向量空间 E 中的正例集 Pe 和反例集 Ne 的大小分别为 p, n , ID3 基于如下两种假设:在空间 E 上的一颗正确决策树对任意实例的分类概率同正反实例的概率。一棵决策树对一实例做出正确判断所需的信息量为:

$$I(p, n) = -(p/(p+n)) \log_2(p/(p+n)) * \log_2(p/(p+n)) - (n/(p+n)) \log_2(n/(p+n)) * \log_2(n/(p+n))$$

如果以某属性 A 作为决策树的根,则 A 具有 m 个值 $\{V_1, V_2, \dots, V_m\}$,它将 E 分成 m 个子集 $\{E_1, E_2, \dots, E_m\}$ 。假设 E_t 中含有 P_t 个正例和 N_t 个反例,那么子集 E_t 所需的期

望信息是 $H(P_t, N_t)$,以属性 A 为根所需的期望熵是

$$E(A) = \sum((P_t + N_t)/(P + N)) I(P_t, N_t)$$

以 A 为根的信息熵增益是: $Gain(A) = I(P, N) - E(A)$ 。ID3 选择使 $Gain(A)$ 具有最大的属性 A^* 作为根节点,对 A^* 的不同取值对应的 E 的 V 个子集 E_t 递归调用上述生成过程生成子节点。

4.2 多 Agent 信息系统中规则库实例

为了生成决策树,对规则库数据进行挖掘,需要建立一个规则库实例。假设一个 Agent 有 3 个内部属性,2 个可以感知的外部属性,还要 1 个返回的属性值。属性的取值范围如下所示,规则库如表 5 所示。

- 血压 = {低, 正常, 高}
- 体温 = {低, 正常, 高}
- 心跳 = {快, 正常, 慢}
- 气温 = {低, 正常, 高}
- 天气 = {好, 坏}
- 心情 = {好, 坏}

表 5 Agent 规则库

| 规则 | 血压 | 心跳 | 体温 | 气温 | 天气 | 心情 |
|-------|----|----|----|----|----|----|
| Rule1 | 低 | 快 | 低 | 低 | 好 | 坏 |
| Rule2 | 正常 | 快 | 正常 | 正常 | 好 | 坏 |
| Rule3 | 高 | 慢 | 高 | 正常 | 坏 | 坏 |
| Rule4 | 正常 | 正常 | 正常 | 高 | 好 | 坏 |
| Rule5 | 低 | 快 | 正常 | 高 | 坏 | 坏 |
| Rule6 | 高 | 慢 | 低 | 正常 | 好 | 坏 |
| Rule7 | 高 | 快 | 正常 | 低 | 坏 | 坏 |
| Rule8 | 低 | 慢 | 低 | 正常 | 好 | 坏 |

4.3 采用 ID3 算法发现 Agent 规则库里的知识

(1)信息熵计算

$$H(U) = -\sum_i P(u_i) * \log_2 P(u_i)$$

$$P(\text{好心情}) = 3/8; P(\text{坏心情}) = 5/8$$

$$H(U) = -((3/8) * \log_2(3/8) + (5/8) * \log_2(5/8)) = 0$$

(2)条件熵计算

$$H(U/V) = -\sum_j P(v_j) \sum_i P(u_i/v_j) * \log_2 P(u_i/v_j)$$

血压 = {低, 正常, 高}; 已知条件如下所示:

$$P(\text{血压低}) = 3/8;$$

$$P(\text{血压正常}) = 2/8;$$

$$P(\text{血压高}) = 3/8;$$

$$P(\text{心情好/血压低}) = 1/3;$$

$$P(\text{心情坏/血压低}) = 2/3;$$

$$P(\text{心情好/血压正常}) = 1/2;$$

$$P(\text{心情坏/血压正常}) = 1/2;$$

$$P(\text{心情好/血压高}) = 1/3;$$

$$P(\text{心情坏/血压高}) = 2/3;$$

$$H(\text{心情/血压}) =$$

$$-(3/8)((1/3) * \log_2(1/3) + (2/3) * \log_2(2/3)) -$$

$$-(2/8)((1/2) * \log_2(1/2) + (0/2) * \log_2(1/2)) -$$

$$-(3/8)((1/3) * \log_2(1/3) + (2/3) * \log_2(2/3)) = 0.94$$

$$H(\text{心情/心跳}) =$$

$$-(3/8)((1/3) * \log_2(1/3) + (2/3) * \log_2(2/3)) -$$

$$-(1/8)((1/1) * \log_2(1/1) + (0/1) * \log_2(0/1)) -$$

$$-(4/8)((1/4) * \log_2(1/3) + (2/3) * \log_2(3/4)) = 0.8$$

(下转第 192 页)

提出了一种基于 MDS 的奇异值探测算法,算法赋予每一数据点相应的置信度,以其来进行奇异值的判定,算法稳定,比以往的奇异值挖掘算法效率也有提高。

参考文献

- 1 Han Jiawei, Kamber M. Data Mining: Concepts and techniques [M]. Morgan Kaufmann Publishers, 2001
- 2 Grubbs F E. Procedures for detecting outlying observations in samples [J]. Technometrics, 1969, 11: 1~21
- 3 Barnett V, Lewis T. Outliers in Statistical Data [M]. John Wiley & Sons, 1994
- 4 Knorr E, Ng R. A unified notion of outliers: Properties and computation. In: Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining (KDD'97), 1997. 219~222

- 5 Knorr E, Ng R. Algorithms for mining distancebased outliers in large datasets. In: Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98), 1998. 392~403
- 6 Arning A, Agrawal R, Raghavan P. A linear method for deviation detection in large databases. In: Proc. 1996 Int. Conf. Data Mining and Knowledge Discovery (KDD'96), 1996. 164~169
- 7 Breunig M, Kriegel H P, Ng R, Sander J. LOF: Identifying Density-Based Local Outliers. In: Proc. ACM SIGMOD 2000 Int. Conf. on Management of Data, 2000. 93~104
- 8 Chang Hong, Yeung D Y. Robust locally linear embedding [J]. Pattern Recognition, 2006, 39: 1053~1065
- 9 Cox T, Cox M. Multidimensional Scaling [M]. London Chapman & Hall, 1994
- 10 Huber P J. Robust regression: asymptotics, conjectures, and Monte Carlo. Ann. Statist, 1973, 1(5): 799~821

(上接第 186 页)

$$\begin{aligned}
 H(\text{心情/体温}) &= \\
 &-(2/8)((1/2) * \log_2(1/2) + (1/2) * \log_2(1/2)) - \\
 &-(4/8)((1/4) * \log_2(3/4) + (1/4) * \log_2(3/4)) - \\
 &-(2/8)((1/2) * \log_2(1/2) + (1/2) * \log_2(1/2)) = 0.91 \\
 H(\text{心情/气温}) &= \\
 &-(2/8)((1/2) * \log_2(1/2) + (1/2) * \log_2(1/2)) - \\
 &-(4/8)((1/4) * \log_2(3/4) + (1/4) * \log_2(3/4)) - \\
 &-(2/8)((1/2) * \log_2(1/2) + (1/2) * \log_2(1/2)) = 0.91 \\
 H(\text{心情/天气}) &= \\
 &-(4/8)((2/4) * \log_2(2/4) + (2/4) * \log_2(2/4)) - \\
 &-(4/8)((1/4) * \log_2(1/4) + (3/4) * \log_2(3/4)) = 0.905
 \end{aligned}$$

(3)互信息计算

$$I(U) = H(U) - H(U|V)$$

$$\begin{aligned}
 I(\text{血压}) &= 0.954 - 0.94 = 0.014; \\
 I(\text{心跳}) &= 0.954 - 0.8 = 0.154; \\
 I(\text{体温}) &= 0.954 - 0.91 = 0.044; \\
 I(\text{气温}) &= 0.954 - 0.91 = 0.044; \\
 I(\text{天气}) &= 0.954 - 0.905 = 0.049;
 \end{aligned}$$

(4)建立决策树的树根和分支

ID3 选择互信息最大的心跳作为树根,在 8 个规则中,对心跳进行分支,3 个分支所对应的子集为

F 心跳慢 = {3, 6, 8}; F 心跳正常 = {4}; F 心跳快 = {3, 6, 8}

F 心跳正常 = {4} 中的例子全部属于“心情好”的类,其余两个集合继续采用该建树方法。

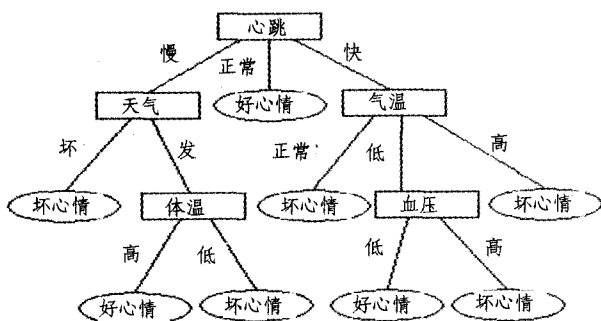


图 3 Agent 的决策树

(5)递归建树

F 心跳慢 = {3, 6, 8}, 所有的值都是心跳慢,所以 $H(U) = H(U/V)$, 故 $I(U) = 0$ 。在剩余的 4 个属性中, $I(\text{天气}) = 0.049$ 互信息最大,所以作为该分支的根节点。再向下分支, F

天气坏 = {3}, 其全部属于“心情”好的类; F 天气好 = {6, 8}; 然后在建立子树。I(体温) = 0.044, 作为其子树的根节点, F 体温高 = {6}, 属于“好心情”类; F 体温低 = {8}, 属于“坏心情”类。

F 心跳快 = {1, 2, 5, 7}, 所有的值都是心跳慢, 所以 $H(U) = H(U|V)$, 故 $I(U) = 0$ 。在剩余的 2 个属性中, $I(\text{气温}) = 0.044$ 的互信息最大, 故选择其为子树的根节点。F 气温低 = {1, 7}; F 气温正常 = {2}, 属于坏心情类; F 气温高 = {5}, 属于坏心情类; 对于 F 气温低 = {1, 7}, 选择 $I(\text{血压}) = 0.014$ 作为子树的根节点。F 血压低 = {1}, 属于“好心情”; F 血压高 = {7}, 属于“坏心情”。根据计算结果画出的 Agent 决策树, 如图 3 所示。

结论 本文根据信息系统的定义, 提出了采用集合方法描述的多 Agent 信息系统, 研究了在多 Agent 交互中规则的形成过程。用实例验证了采用 ID3 算法挖掘 Agent 规则, 发现知识过程的可行性。在后续的研究中, 将把知识发现功能融入到 Agent 系统中, 使其具备更强的认知能力。

参考文献

- 1 Wooldridge M, Jennings N R. Intelligent agents: theory and practice. The Knowledge Engineering Review, 1995, 10(2): 115~152
- 2 Cohen P R, Levesque H J. Intention is choice with commitment. Artificial Intelligence, 1990, 42(2): 213~261
- 3 Konolige K, Pollack M E. A representation a list theory of intention. In: Bajcsy, R., ed. Proceedings of the 13th International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1993. 390~395
- 4 Halpern J Y, Moses Y. A guide to completeness and complexity for modal logics of knowledge and belief. Artificial Intelligence, 1992, 54: 319~379
- 5 Kraus S, Sucara K, Evenchik A. Reaching agreements through argumentation: a logical model and implementation. Artificial Intelligence, 1998, 104~169
- 6 Allen J F. Towards a general theory of action and time. Artificial Intelligence, 1984, 23(2): 123~154
- 7 Zhang Wenran, Chen Sushinn. Pool2: A generic system for cognitive map development and decision analysis. IEEE Transactions on Systems, Man and Cybernetics, 1989, 19(1): 31~39
- 8 Tomohiro T, Michio S. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics, 1985, 15(1): 116~132
- 9 Quinlan J R. Induction of decision trees [J]. Machine Learning, 1986, 1: 81~106
- 10 Sandholm T W, Lesser V R. Coalition among computationally bounded agents. Artificial Intelligence, 1997, 94: 99~137
- 11 刘东升. 基于 MobileAgent 的分布式 ID3 挖掘模型. 计算机应用与软件 [J], 2005, 22(10): 49~51
- 12 王熙照, 谢竞博. 基于属性间交互信息的模糊 ID3 算法的扩展. 复以学报(自然科学版) [J], 2004, 43(5): 777~780
- 13 王澜, 何华灿. 基于广义相关系数的 Agent 行为决策模型. 计算机科学 [J], 2005, 32(2): 175~177