基于知识粗糙度的混合变量决策树生成方法

路红梅[1,2],胡学钢[1]
(1.合肥工业大学 计算机与信息学院,安徽 合肥 230009;
2.宿州学院 计算机系,安徽 宿州 234000)

摘 要:单变量决策树难以反映信息系统属性间的关联作用,构造的决策树往往规模较大。多变量决策树能较好地反映属性间的关系,得到非常简单的决策树,但使构造的决策树难以理解。针对以上两种决策树特点,提出了基于知识粗糙度的混合变量决策树的构造方法,选择知识粗糙度较小的分类属性来构造决策树。实验结果表明,这是一种操作简单、效率很高的决策树生成方法。

关键词:粗糙集;知识粗糙度;单变量决策树;多变量决策树;混合变量决策树
中图分类号:TP18 **文献标识码**:A **文章编号**:1673-629X(2008)01-0056-03

Construction of Hybrid Decision Tree Based on Knowledge Roughness

LU Hong-mei[1,2], HU Xue-gang[1]
(1. School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China;
2. Department of Computer Science, Suzhou College, Suzhou 234000, China)

Abstract: It is difficult for univariate decision tree to reflect the relationship of attributes, multivariate decision tree can resolve this problem preferably, the former produces big tree, the latter gains simple tree but difficult to explain. Aim to upwards points, in this paper, advance a knowledge roughness based approach to hybrid decision tree, select less knowledge roughness as tested attribute to construct decision tree. As a result, find this is a good approach with simple operation and higher efficiency.

Key words: rough sets; knowledge roughness; univariate decision tree; multivariate decision tree; hybrid decision tree

0 引 言

决策树分类方法由于具有速度快、精度高、生成模式简单等优点[1]而得到了广泛应用。大多数决策树构造方法在每个节点上只检验单个属性,这种单变量决策树忽视了信息系统中广泛存在的属性间的关联作用,而且修剪时往往代价很大[2]。苗夺谦等人利用粗糙集合理论中条件属性相对于决策属性的核,用相对泛化的概念构造多变量检验,提出了一种构造多变量决策树的方法[3],该方法可以构造非常简单的决策树,但当核中属性较多时,决策树结点中的属性过多,因而对结点分裂条件的描述比较困难,使所构造的决策树难以理解。研究表明,单变量决策树的复杂度主要由树的结点个数决定,而多变量决策树的复杂度主要由结点中属性的个数决定[4]。针对以上两种决策树特点,胡学钢等提出了混合变量决策树结构[5],文中在此基础上提出了基于知识粗糙度的混合变量决策树的构造方法,选择知识粗糙度较小的分类属性来构造决策树。实验结果表明,这是一种操作简单、效率很高的决策树生成方法。

收稿日期:2007-03-26
基金项目:安徽省自然科学基金项目(2006kj091B)
作者简介:路红梅(1965-),女,山东聊城人,副教授,硕士研究生,研究方向为数据挖掘、计算机控制等;胡学钢,博士,教授,硕士生导师,研究方向为人工智能、数据挖掘等。

1 基本概念

定义1 一个信息表知识表达系统 $S=<U,R,V,f>$,这里 U 是对象的集合,也称为论域,R 是属性的集合,V 是属性值的集合,$f:U\times R\rightarrow V$ 是一个信息函数。若属性集合 R 可以划分为两个不相交的集合:条件属性集 C 和决策属性集 D,满足 $R=C\cup D$ 且 $C\cap D=\varnothing$,这样的信息系统也称为决策系统。

关于粗糙集的基本概念如等价关系、等价类以及集合的上逼近、下逼近、边界的论述可参考文献[6,7]。

定义2 设 $A\subseteq C, B\subseteq D, U/A=\{X_1,X_2,\cdots,X_n\}$ 和 $U/B=\{Y_1,Y_2,\cdots,Y_m\}$ 分别是由等价关系 A 和 B 对 U 的划分,则集合 U/B 的子集 Y_i 是论域 U 上的一个关于知识 A 的 Rough 集,定义其 A 精度(在不

发生混淆的情况下,也简称精度)为:

$$\begin{cases} d_A(Y_i) = |A_-(Y_i)|/|A^-(Y_i)| & Y_i \neq \varnothing \\ d_A(Y_i) = 1 & Y_i = \varnothing \end{cases}$$

定义其 A 粗糙度(在不发生混淆的情况下,也简称粗糙度)为:

$$P_A(Y_i) = 1 - d_A(Y_i)$$

定义 3　定义 $\mathrm{Min}(P_A(Y_i),(i=1,\cdots,m))$ 作为集合 U/B 关于知识 A 的粗糙度 $P_A(B)$。

定义 4　设 P 和 Q 是 U 上的 2 个等价关系族,令:

$$U/\mathrm{IND}(P) = \{X_1, X_2, \cdots, X_n\}$$

$$U/\mathrm{IND}(Q) = \{Y_1, Y_2, \cdots, Y_m\}$$

$$Z_i = \bigcup_{X_j \in \mathrm{IND}(P)} \{X_j : X_j \subset Y_i\} \quad i = 1,2,\cdots,m$$

$$Z_{m+1} = \bigcup_{X_j \in \mathrm{IND}(P)} \{X_j : X_j \notin Y_i, \forall i\}$$

则称 $\{Z_1, Z_2, \cdots, Z_{m+1}\}$ 在 U 上确定的等价关系为 P 相对于 Q 的泛化[3],记做 $\mathrm{GEN}_Q(P)$。

用等价关系的泛化代替原有的属性值的合取作为多变量的检验,可以避免对数据的过分拟合,同时也简化了决策树。

定义 5　决策树学习算法在每个结点根据具体的数据集,选择尽可能少的属性明确分类尽可能多的实例,以此决定当前结点分裂属性的个数,因而在某个结点可能采用单变量,也可能采用多变量分裂,故称之为混合变量决策树。

2　构造决策树的算法

文中提出的基于知识粗糙度的混合变量决策树构造算法,其输入是一个待分类的数据集合,输出是一棵决策树。具体算法如下所示:

(1) 对条件属性集 C 中每一个未被单选过的属性(假定有 k 个)计算其知识粗糙度,若 $P_{ci}(D) = 1$ $(i = 1,2,\cdots,k)$,转(5);

(2) 若某条件属性形成的划分的知识粗糙度为 0,则选择该单变量属性为根结点,标志该属性为已选;转(1);

(3) 否则选择条件划分粗糙度最小及次小的(如果有的话)两条件属性(粗糙度为 1 的属性不能被选)利用相对泛化划分论域 $U = \{Z_1, Z_2, \cdots, Z_{m+1}\}$,形成多变量分裂属性,标记 $Z_1, Z_2, \cdots, Z_m$ 为叶结点;

(4) 若 $Z_{m+1} = \varnothing$,标记为叶结点,算法结束;否则,令 Temp U 为 Z_{m+1} 中所包含实例的集合,U = Temp U,转(1);

(5) 计算 C 中每个未被单选过的属性的信息熵,选择熵最小的属性作为单变量分裂属性,标志该属性为已选,转(1)。

3　实例分析

以一个例子说明算法的工作流程,将得到的决策树与传统的基于信息增益的 ID3 算法构造的决策树比较,说明算法的可行性和优越性。

表 1 所示数据集选自文献[4],表中共有 24 条记录,每条记录对应 5 个属性,其中前 4 个属性 Outlook, Temperature, Humidity, Windy 为条件属性,最后一个属性 Class 为决策属性。

表 1　The decision table

ID	Outlook	Temperature	Humidity	Windy	Class
1	Overcast	Hot	High	Not	N
2	Overcast	Hot	High	Very	N
3	Overcast	Hot	High	Medium	N
4	Sunny	Hot	High	Not	P
5	Sunny	Hot	High	Medium	P
6	Rain	Mild	High	Not	N
7	Rain	Mild	High	Medium	N
8	Rain	Hot	Normal	Not	P
9	Rain	Cool	Normal	Medium	N
10	Rain	Hot	Normal	Very	N
11	Sunny	Cool	Normal	Very	P
12	Sunny	Cool	Normal	Medium	P
13	Overcast	Mild	High	Not	N
14	Overcast	Mild	High	Medium	N
15	Overcast	Cool	Normal	Not	P
16	Overcast	Cool	Normal	Medium	P
17	Rain	Mild	Normal	Not	N
18	Rain	Mild	Normal	Medium	N
19	Overcast	Mild	Normal	Medium	P
20	Overcast	Mild	Normal	Very	P
21	Sunny	Mild	High	Very	P
22	Sunny	Mild	High	Medium	P
23	Sunny	Hot	Normal	Not	P
24	Rain	Mild	High	Very	N

利用第 1 节有关定义以及第 2 节给出的算法可知:

U = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24}

U/Outlook = {{1,2,3,13,14,15,16,19,20},{4,5,11,12,21,22,23},{6,7,8,9,10,17,18,24}};

U/Temperature = {{1,2,3,4,5,8,10,23},{6,7,13,14,17,18,19,20,21,22,24},{9,11,12,15,16}};

U/Humidity = {{1,2,3,4,5,6,7,13,14,21,22,24},{8,9,10,11,12,15,16,17,18,19,20,23}};

U/Windy = {{1,4,6,8,13,15,17,23}, {2,10,11,20,21,24},{3,5,7,9,12,14,16,18,19,22}};

U/Class = {{1,2,3,6,7,9,10,13,14,17,18,24},{4,5,8,11,12, 15,16,19,20,21,22,23}}。

依次计算决策属性相对于每个条件属性的知识粗糙度 $P_{ci}(D)$:

$P_{Outlook}(D) = 17/24$, $P_{Temperature}(D) = 1$,

$P_{Humidity}(D) = 1$, $P_{Windy}(D) = 1$

所以选取 Outlook 单变量作为决策树根结点，由于属性 Outlook 有三个取值，原始数据集将被分为三个子集 $U1, U2, U3$，在每个子集中，所有的训练实例都具有相同的 Outlook 的值，$U2$ 子集中所有实例均已明确分类，标记为叶结点。

在 $U1$ 分支上持续进行相同的工作：

$U1$/Temperature = {{1,2,3}, {13,14,19,20}, {15,16}}

$U1$/Humidity = {{1,2,3,13,14},{15,16,19,20}}

$U1$/ Windy = {{1,13,15},{2,20},{3,14,16,19}}

$U1$/Class = {{1,2;3,13,14},{15,16,19,20}}

$P_{Temperature}(D) = 4/7$, $P_{Humidity}(D) = 0$, $P_{Windy}(D) = 1$

由于 $P_{Humidity}(D) = 0$，选择属性 Humidity 作为测试属性，此时两个分支中所有的训练实例都分别具有相同的类，该分支的划分结束。

在 $U3$ 分支上持续进行相同的工作：

$U3$/Temperature = {{8,10}, {6,7,17,18,24}, {9}}

$U3$/Humidity = {{6,7,24},{8,9,10,17,18}}

$U3$/Windy = {{6,8,17},{10,24},{7,9,18}}

$U3$/Class = {{6,7,9,10,17,18,24},{8}}

$P_{Temperature}(D) = 2/8$, $P_{Humidity}(D) = 5/8$, $P_{Windy}(D) = 3/8$

由于 $P_{Temperature}(D) < P_{Windy}(D) < P_{Humidity}(D)$，选择 Temperature、Windy 两属性的组合进行测试。

设 P = Temperature ∧ Windy,

$U3/P$ = {{6,17},{7,18},{8},{9},{10},{24}}

$U3$/Class = {{6,7,9,10,17,18,24},{8}}

计算在 $U3$ 上确定的等价关系 P 相对于 Class 的泛化，记作 $GEN_{class}(P)$。

$Z_1 = \{6,7,9,10,17,18,24\}$, $Z_2 = \{8\}$

由于 $Z_3 = \emptyset$，此时所有的训练实例都能明确地分类，该分支的划分结束。

由该实例构造的基于知识粗糙度的混合变量决策树 T^H，如图 1 所示，图 1 中括号里的数字代表每个叶结点含有的实例个数。图 2 给出的是基于信息增益的 ID3 方法建立的决策树 T^E，显然，T^H 比 T^E 简单的多，得到的规则也比 T^E 的简练。

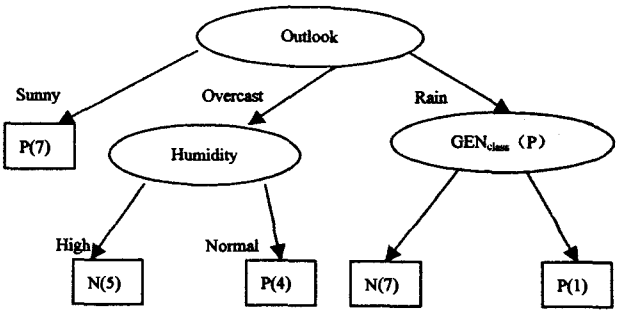

图 1 基于知识粗糙度的混合变量决策树 T^H

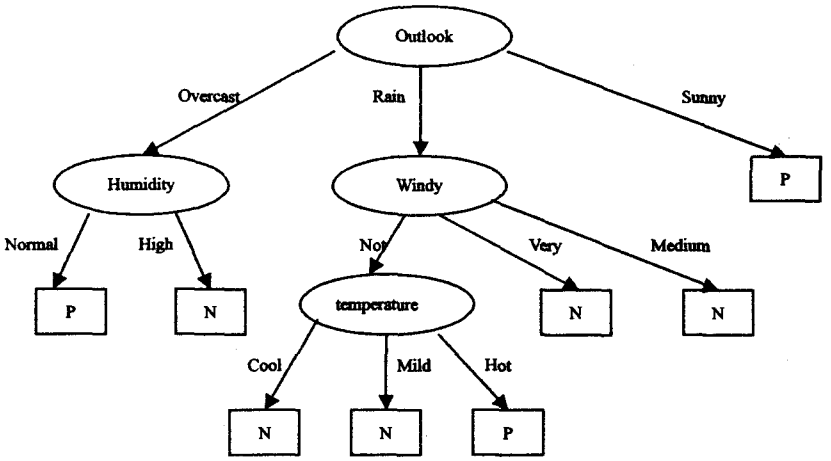

图 2 基于信息增益的决策树 T^E

4 结束语

决策树是多属性归纳学习的重要形式，单变量决策树强调知识内划分的细度，忽视了信息系统中广泛存在的属性间的关联作用，在选择属性时具有一定的缺陷，难以得到最佳决策树。苗夺谦等人提出的构造多变量决策树的方法，可以构造非常简单的决策树，较好地反映了属性间的关联作用，但当核中属性较多时，决策树结点中的属性过多，造成对结点分裂条件的描述困难，使所构造的决策树难以理解。基于知识粗糙度的混合变量决策树可以较好地弥补两者的不足，因为知识粗糙度反映的是无法进行正确分类的程度，它是由 X 在 B 下的上近似和下近似产生的，它描述了知识的内划分和外划分对集合 X 的逼近程度[8]。特别是当属性的各取值的关联度较强，且无矛盾数据时，文中提出的方法，与基于信息熵的方法相比，能得到更优的决策树，同时也减小了算法的计算量。

参考文献：

[1] Breiman L, Friedman J H, Olshen R A. Classification and Regression Trees[M]. Belmont: Wadsworth International

（下转第 62 页）

回报函数 r 的值满足：$|f(d_{og}, d_{oa}, v_o)| < 0.005$，动作集 Move(dir, power)离散为 24 * 5 个值。如果在 500 个仿真周期内，进攻 Agent 没有成功带球推进到禁区或者防守 Agent 成功断球，则认为防守成功，否则防守失败。经过 30000 个场景的训练，得到了基本稳定的防守策略。

将学习后的防守策略应用到 3D 仿真球队 Fantasia 中，并与 2005 年全国比赛的冠军浙大求是队、亚军科大蓝鹰队以及 2005 年世界杯冠军 Aria 分别进行了 100 场比赛(每场比赛 6000 个仿真周期)，每场比赛对手带球推进到禁区的次数和失球数如表 1 和 2 所示。

表 1 对手带球推进到禁区的平均次数

	随机策略	手工代码	学习代码
浙大求是队	16.62	12.37	4.18
科大蓝鹰队	14.35	9.28	2.49
Aria	18.21	12.56	4.75

表 2 平均失球数

	随机策略	手工代码	学习代码
浙大求是队	3.13	1.21	0.33
科大蓝鹰队	2.51	0.82	0.15
Aria	3.86	1.24	0.38

应用该防守策略的 3D 仿真球队 Fantasia 参加了 2006 年 6 月份在德国不来梅举行的 Robocup 世界杯和 2006 年 10 月份在苏州举行的 Robocup 全国比赛暨中国公开赛 3D 仿真组的比赛。在不来梅进行的 10 多场世界杯比赛中，Fantasia 只丢了一个球，输了一场比赛，并最终获得世界杯第 11 名；在苏州参加的近 20 场比赛中，Fantasia 一球未失，最终以不败战绩囊括全国比赛和中国公开赛的冠军。

5 结论与展望

针对 Robocup 半场防守这一对抗性问题，将多对多的半场防守任务分解为多个一对一防守的子任务，并用 Markov 零和对策模型描述该任务，研究了基于 Markov 对策的强化学习方法在一对一防守中的应用。实验和比赛结果表明，该方法取得了优于手工代码的应用效果。由于篇幅有限，在防守任务分配问题上未作详细讨论，实际上该环节还有改进的余地，例如可以采用“匈牙利算法”[8]或者基于阵型的角色分配[9]等解决进攻 Agent 和防守 Agent 的一对一匹配问题。

参考文献：

[1] Kitano H, Tambe M, Stone P, et al. The RoboCup Synthetic agent challenge97[C]// In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence. Nagoya: [s. n.], 1997: 24 - 29.

[2] Stone P. Layered Learning in Multi - Agent Systems[D]. Pittsburgh, PA, USA: Computer Science Department, Carnegie Mellon University, 1998.

[3] Yao Jinyi, Chen Jiang, Cai Yunpeng, et al. Architecture of Tsinghua Aeolus[C]// In: Birk A, Coradeschi S, Tadokoro S eds. Robocup 2001: Robot Soccer World Cup V. Heidelberg: Springer - Verlag, 2002.

[4] Riedmiller M, Braun H. A direct adaptive method for faster back - propagation learning: The RPROP algorithm[C]// In Ruspini H ed. Proceedings of the IEEE International Conference on Neural Networks (ICNN). San Francisco: [s. n.], 1993: 586 - 591.

[5] 张汝波. 强化学习理论及应用[M]. 哈尔滨：哈尔滨工程大学出版社，2001.

[6] Owen G. Game Theory[M]. 2nd Edition. Orlando, FL, USA: Academic Press, 1982.

[7] Littman M L. Markov games as a framework for multi - agent reinforcement learning [C] // In Proceedings of the Eleventh International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann, 1994: 157 - 163.

[8] 常庭懋，韩中庚. 用“匈牙利算法”求解一类最优化问题[J]. 信息工程大学学报，2004，5(1)：60 - 62.

[9] Stone P, Veloso M. Task decomposition. dynamic role assignment, and low - bandwidth communication for real - time strategic teamwork[J]. Artificial Intelligence, 1999, 110(2): 241 - 273.

(上接第 58 页)

Group, 1984.

[2] Yin De - Sheng, Wang Guo - Yin, Wu Yu. A Self - learning Algorithm for Decision Tree Pre - Pruning[C]// Proceedings of the Third International Conference on Machine Learning and Cybernetics. Shanghai: [s. n.], 2004.

[3] 苗夺谦，王 珏. 基于粗糙集的多变量决策树的构造方法[J]. 软件学报，1997，8(6)：425 - 431.

[4] 史忠植. 知识发现[M]. 北京：清华大学出版社，2002.

[5] 胡学钢，张冬艳. 一种新的基于粗集的决策树构造算法[J]. 计算机科学，2005，32(8A)：7 - 9.

[6] Pawlak Z. Rough Set Approach to Multi - Attribute Decision Analysis[J]. European Journal of Operational Research, 1994, 72(3): 443 - 459.

[7] 曾黄麟. 粗集理论及其应用[M]. 重庆：重庆大学出版社，1998.

[8] 沙慧新. 基于知识粗糙度和拓展属性约简的若干智能挖掘算法的研究[D]. 福州：福州大学，2004：13 - 17.