增量式的多变量决策树构造算法研究

常志玲[1],张晓玲[2]
(1. 洛阳师范学院 信息技术学院,河南 洛阳 471022;
2. 河南科技大学 电子信息工程学院,河南 洛阳 473000)

摘 要:针对增量数据集,结合粗糙集理论和多变量决策树的优点,给出了增量式的多变量决策树构造算法。该算法针对新增样本与已有规则集产生矛盾,即条件属性相匹配,而决策属性不匹配的情况,计算条件属性相对于决策属性的核,如果核不为空,则计算核相对于决策属性的相对泛化,根据不同的结果形成不同的子集,最终形成不同的决策树分支。该算法很好地避免了在处理增量数据集时,不断重构决策树。实例证明该算法的正确性,对处理小增量数据集具有良好的性能。

关键词:增量式学习;多变量决策树;粗糙集;相对泛化
中图分类号:TP18 **文献标识码**:A **文章编号**:1673-629X(2011)02-0090-04

Study of Building Incremental Multivariate Decision Tree

CHANG Zhi-ling[1], ZHANG Xiao-ling[2]
(1. Academy of Information Technology, Luoyang Normal University, Luoyang 471022, China;
2. Electronic & Information Engineering College of Henan University of Science and Technology, Luoyang 473000, China)

Abstract: In this paper, a new algorithm to build incremental multivariate decision tree is proposed. The advantages of the rough set theory and the multivariate decision tree are combined in this method. Aiming at the inconsistency between the new sample and the old sample, the core is computed. If the core is empty, the generalization between core and decision attribute will be computed, the different results will be the different branches of decision tree at last. The decision tree rebuilding is avoided in the algorithm and the validity of the algorithm is proved by the example.

Key words: incremental learning; multivariate decision tree; rough set; generalization

0 引 言

所谓增量式学习[1],就是针对一个数据集,当增加新样本时,仅仅在原数据集的基础上作由新样本引起的更新,而不需要重建所有的数据集。这样数据集随着样本数据的增加就处于时常更新状态,就能够在原有知识的基础上进行快速的学习,进而节省了大量的时间。而现实生活中,数据集就是不断增加的,例如超市、银行等行业数据一天就有上万条记录增加,因此增量式的学习更符合人们的思维。

从20世纪80年代中期开始,一些学者对决策树的增量学习能力进行了研究,主要研究成果有:ID_4增量学习算法[2]、ID_5R算法[3]及ITI算法,还有其他的一些算法[4,5]等。另外还有关于增量决策树的一些应用研究[6],但这些算法构造出来的都是增量式的单变量决策树。决策树有单变量和多变量之分,单变量决策树就是在每个节点上只检验单个属性,不考虑属性间的相关性,这一限制使得有些属性在一棵决策树中某一路径上被多次检验。多变量决策树在树的节点上可以同时检验多个属性,其优点是叶节点数和深度比较小。

文中针对上述问题,应用粗糙集理论[7],针对动态增长的数据集,提出了增量式的多变量决策树构造算法,实例表明随着样本的增加,本算法并不需要对决策树进行重新构造,而只需要重构与样本相关的子树,大大降低了建树的复杂性,并且获得很好的分类能力。

收稿日期:2010-06-04;**修回日期**:2010-09-28
基金项目:河南省自然科学研究计划项目(2010A520030)
作者简介:常志玲(1976-),女,河南濮阳人,硕士研究生,讲师,主要研究方向为粗糙集理论、数据挖掘;张晓玲,硕士,讲师,研究方向为数据挖掘。

1 相关概念介绍

1.1 决策树

决策树[8,9]是指用树形结构来表示决策集合,是一种直观的知识表达方法,同时也是高效的分类器,可

以非常容易地产生关联规则。其中每个内部节点表示在一个属性上的测试,每个分枝代表一个测试输出,而每个树叶节点代表类或类分布。树的最顶层节点是根节点。构造决策树的主要思想是以信息论[9]为工具,在各非叶节点选择重要的属性或属性组,自上而下分割训练实例集,直到满足某种终止条件,即节点中的实例属于同一类。

理想的决策树分为3种[10]:(1)叶节点数最少;(2)叶子节点深度最小;(3)叶节点数最少且叶子节点深度最小。但是最优决策树已经被证明是一个NP-hard问题。

1.2 粗糙集

粗糙集(Rough sets)理论是由波兰科学家Pawlak[7]于20世纪80年代提出的一种处理不确定问题的方法,它的观点就是[11]:知识(即人的智能)就是一种对对象进行分类的能力,可以用等价类形式化表示分类,可以这样理解:知识是用等价类(记为R)对离散空间的一种划分,记为$U/R=\{X_1,X_2,\cdots,X_n\}$,其中$X_i$就是$U/R$的一个等价类。

1.3 决策表

一个决策表可以形式化定义为[7]:$S=<U,C\cup D,V,f>$,其中$U=\{u_1,u_2,\ldots,u_n\}$是所感兴趣对象的有限集合,$C\cup D$是属性的有限集,其中C为条件属性集,D为决策属性集,并且$C\cap D=\varnothing$,V为属性集$C\cup D$的值域,$f:U\times(C\cup D)\rightarrow V$为一个信息函数,表示任一对象的属性在$V$上的取值,即$f(x,r)\in V_r$,它指定了$U$中每一对象$x$的属性值。

为表达语言中的决策规则,其中θ和ψ分别称为$\theta\rightarrow\psi$的因和果。对于一个决策表S,当所有规则$\theta\rightarrow\psi$为真时,则称决策表S是相容的,否则称不相容。

1.4 核

对于任何子集$X\subseteq U$,称为一个概念。对于每个概念X可以定义上、下近似为[7]:

$$R\_X=\cup\{X_i\in U:X_i\subseteq X\}\qquad R^{-}X=\cup\{X_i\in U:X_i\cap X\neq\varnothing\}$$

其中R_X是由U上在现有知识R下肯定属于X的元素组成的集合;$R^{-}X$是可能属于X的元素组成的集合。设P和Q是U上的两个等价关系,那么Q的P-正域定义为:

$$\mathrm{POS}_P(Q)=\bigcup_{X\in U/Q}P\_X \tag{1}$$

$\mathrm{POS}_P(Q)$是U中所有那些通过知识P被肯定属于U/Q的元素组成的集合。如果

$$\mathrm{pos}_p(Q)=\mathrm{pos}_{(P-\{r\})}(Q) \tag{2}$$

成立,则称$r\in P$是Q-不必要的,否则r在P中是Q-必要的。P中所有Q-必要的等价关系组成的集合称为P的Q-核,记为$\mathrm{core}_Q(P)$[7]。对于整个决策表来说,核属性是非常重要的,去掉核中属性将改变整个决策表的决策。

1.5 相对泛化的定义

相对泛化是定义在两个等价关系之间的,那么一个等价关系相对于另外一个等价关系的泛化定义为[12]:

设P和Q是U上的两个等价关系簇,且

$$U/P=\{X_1,X_2,\cdots,X_n\}\qquad U/Q=\{Y_1,Y_2,\cdots,Y_m\}$$

令 $$Z_i=\bigcup_{X_j\in U/P}\{X_j:X_j\subseteq Y_i\}\quad i=1,2,\cdots,m \tag{3}$$

$$Z_{m+1}=\bigcup_{X_j\in U/P}\{X_j:X_j\not\subset Y_i,\forall i\} \tag{4}$$

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,记为$\mathrm{GEN}_Q(P)$。

2 增量式的多变量决策树构造算法

2.1 算法描述

针对决策表$S=(U,C\cup D,V,f)$,其中$C=\{a_1,a_2,\cdots,a_n\}$是条件属性集,$D=\{d_1,d_2,\cdots,d_n\}$是决策属性集,假定决策表中样本是动态增长的。那么新增一个样本,存在三种情形:

情形1:新增样本与已有规则集相容。

情形2:新增样本与已有规则集相容,但不包含。

情形3:新增样本与已有规则集产生矛盾,即条件属性相匹配,而决策属性不匹配。

针对这三种情形,结合核相对于决策类的泛化对该决策表进行多变量决策树的构造。其算法描述如下:

算法:增量式多变量决策树构造(IMDT)

输入:动态增长的决策表$S=(U,C\cup D,V,f)$

输出:增量式的多变量决策树

(1)如果根节点为空,则把样本放入根节点的样本集,任选一属性a_i作为根节点的分裂属性;

(2)否则,将样本沿树进行匹配,直到到达一个叶节点。如果新增样本与已有规则集相容,则决策树无需任何修改转(9);如果新增样本与已有规则集相容,但不包含,则需要增加新的分支转(9);如果新增样本与已有规则集产生矛盾,则转(3);

(3)对开始不匹配的节点所包含的子集,计算C相对于D的核,即$\mathrm{core}_D(C)$。若$\mathrm{core}_D(C)=\varnothing$则转(4);否则,不妨设$\mathrm{core}_D(C)=\{a_1,a_2,\cdots,a_k\}$,如果$\mathrm{core}_D(C)$与作为子树节点的分裂属性组不相同则转(5),否则,转(6);

(4)用ID_3的方法选择一个最佳属性,作为根节点,根据属性的不同取值将S分裂为$S_1,S_2,\cdots S_{|N|}$,针对子集$S_i(i=1,2,\cdots,|N|)$,如果S_i中的所有样本

都在同一决策类则转(7),否则如果用于划分的属性不为空则令 $C = C_i$, $D = D_i$ 转(3);

(5) $P = a_1 \wedge a_2 \wedge \cdots \wedge a_k$,作为子树节点,计算 P 相对于 D 的泛化 $GEN_D(P)$,根据不同的结果形成不同的子集,记为 S_1 , S_2 ,… $S_{|N|}$,针对子集 $S_i(i = 1,2,\cdots,|N|)$,如果 S_i 中的所有样本都在同一决策类则转(7),否则如果用于划分的属性不为空则令 $C = C_i$, $D = D_i$ 转(3);

(6)针对与新增样本产生不相容规则的子树所有样本和新增样本合为一新的子集,计算该子集中 P 相对于 D 的泛化 $GEN'_D(P)$,根据不同的结果形成不同的子集,记为 S'_1 , S'_2 ,… $S'_{|n|}$,针对子集 $S'_i(i = 1,2,\cdots,|N|)$,如果 S'_i 中的所有样本都在同一决策类则转(7),否则如果用于划分的属性不为空则令 $C = C'_i$, $D = D'_i$ 转(3);

(7)返回 N 为叶节点,以类 C 标记;

(8)如果多个分支包含了分类属性组的所有取值,则合并该多个分支为一个分支;

(9)返回一棵增量式的多变量决策树。

2.2 实例分析

利用文献[12]中一个相容决策表如表1所示,属性集 $C = \{a_1,a_2,a_3,a_4\}$ 是条件属性集,属性集 $D = \{d\}$ 是决策属性集。决策树的内部节点(又名分裂节点)用椭圆形表示;决策树的叶节点用它的决策类代表,并用矩形表示,同时为了清楚起见,在矩形框中标示出所包含子集。利用文中给出的算法构造增量式多变量决策树的执行过程如下:

U	Outlook(a1)	Temperature(a2)	Humidity(a3)	Windy(a4)	Class(d)
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	P
4	rain	mild	high	false	P
5	rain	cool	normal	false	P
6	rain	cool	normal	true	N
7	overcast	cool	normal	true	P
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	P
10	rain	mild	normal	false	P
11	sunny	mild	normal	true	P
12	overcast	mild	high	true	P
13	overcast	hot	normal	false	P
14	rain	mild	high	true	N

1. 输入样本1(sunny,hot,high,false,N),由于根节点为空,任选 a_1 作为分裂属性开始建立决策树如图1所示。

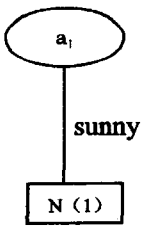

图1 样本1生成的决策树

2. 输入样本2,新增样本与已有规则集相容,则决策树无需任何修改,只把样本2放入样本1所在的子集即可。

3. 输入样本3,样本4,新增样本与已有规则集相容,但不包含,则需要增加新的分支,如图2所示。

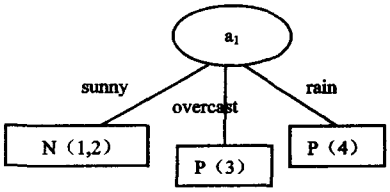

图2 样本2生成的决策树

4. 输入样本5,新增样本与已有规则集相容,则决策树无需任何修改,只把样本5放入样本4所在的子集即可。

5. 输入样本6,新增样本与已有规则集产生矛盾,并且是从根节点开始就不匹配,此时计算包括6个样本在内的决策表的核,通过简单计算可得:

$U/C = \{\{1\}\{2\}\{3\}\{4\}\{5\}\{6\}\}$

$U/D = \{\{12,6\}\{3,4,5\}\}$

由公式(1)可得:

$POS_C(D) = \{1,2,3,4,5,6\}$

考察 $a_i(i = 1,2,3,4)$,在 C 中相对于 D 来说是否必要。为此,从 C 中去掉 a_1 ,可得:

$POS_{(C-\{a_1\})}(D) = \{2,4,5,6\} \neq POS_C(D)$

由公式(2)可得 a_1 在 C 中是 D -必要的。同理可以计算 a_4 在 C 中是 D -必要的,而 a_2 和 a_3 在 C 中是 D -不必要的,由此可得 $CORE_D(C) = \{a_1,a_4\}$ 。由于核和根节分裂属性 a_1 不一致,因此要计算核相对于决策类的泛化:

令 $P = a_1 \wedge a_4$,下面计算 P 相对于 D 的泛化在 U 上导出的划分:

$U/P = \{\{1\}\{2\}\{3\}\{4,5\}\{6\}\}$

由公式(3)和(4)可以计算出:

$GEN_D(P) = \{\{1,2,6\}\{3,4,5\}\}$

由算法可知以 $GEN_D(P)$ 为决策树的根节点,根据所求泛化结果,把决策表中的样本分成不同的对象集。其中子集 $\{1,2,6\}$ 都在同一决策类 N 中,因返回叶节点并以 N 作为标记,同理子集 $\{3,4,5\}$ 都在同一决策类 P 中,返回叶节点并以 N 作为标记,返回决策树如图3所示。

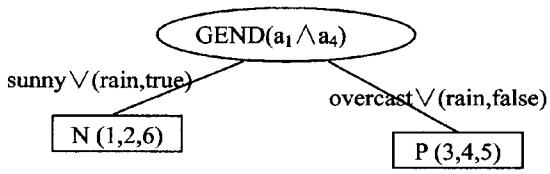

图 3　输入样本 6 后生成的决策树

6. 输入样本 7 和样本 8,新增样本与已有规则集相容,则决策树无需任何修改,只把样本 7 和样本 8 分别放入所在的子集即可。

7. 输入样本 9,新增样本与已有规则集产生矛盾,并且是从根节点开始就不匹配,此时计算包括 9 个样本在内的决策表的核,计算得出其核和根节点分裂属性组相同,即: $CORE_D(C)=\{a_1,a_4\}$,所以只修改产生矛盾的分支即可,即在核不变的情况下,重新计算子集(1,2,6,8,9) 核相对于决策类的泛化,并以泛化为基础进行重建子树,在子树重建过程中子集 (1,8,9) 决策类不一致,并且还存在未用于划分的属性 $\{a_2,a_3\}$,则对此子集重新调用本算法选择 a_3 为分裂属性,如图 4 所示。

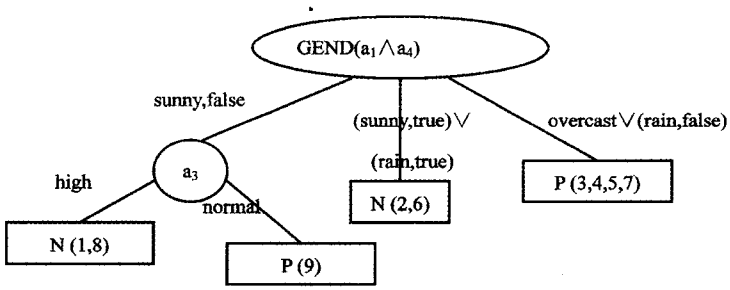

图 4　输入样本 9 之后的决策树

8. 输入样本 10,新增样本与已有规则集相容,则决策树无需任何修改,只需把样本分别放入所在的子集即可;

9. 输入样本 11,新增样本与已有规则集产生矛盾,其情况和步骤 7 相同,采用同样的处理方法,构造的决策树如图 5 所示(只是在图 5 的基础上去除还未输入的样本 12,13,14)。

10. 分别输入样本 12,13,14 后,新增样本与已有规则集相容,则决策树无需任何修改,只需把样本分别放入所在的子集即可;最终获得的决策树如图 5 所示。

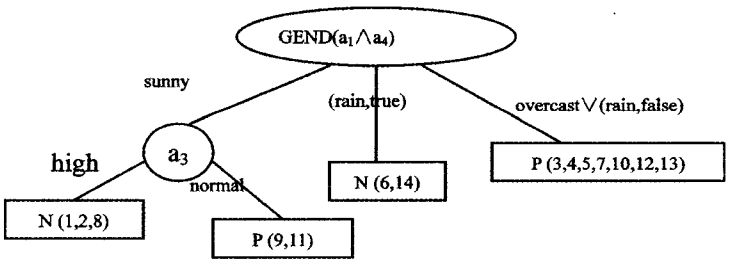

图 5　输入样本 14 之后的决策树

2.3　结果分析

从决策树的构造过程来看,随着样本的增加,并不需要对决策树进行重新构造,而只需要重构与样本相关的子树,大大降低了建树的复杂性,从实例可以看出,文中构造的增量式的多变量决策树最终结果和文献[12]算法所构造的静态多变量决策树相同,因此具有相同的分类能力,为考察本算法的有效性,又对文献[12]等多个经典数据集进行增量式的多变量决策树构造,结果表明都能够构造出分类能力相同的决策树。

3　结束语

增量式的多变量决策树算法结合粗糙集理论和多变量决策树的优点,处理增量数据集的多变量决策树构造问题,解决了传统的多变量决策树构造算法不能处理增量数据集的缺点。通过实例分析,利用增量算法可以一次完成决策树的构造,避免了对数据集的重复扫描和决策树的不断重构问题,而且可以构造出与静态多变量决策树相同的分类能力的决策树。

参考文献:

[1] 王　利,张喜平,郭　林. 增量式知识获取算法综述[J]. 重庆邮电大学学报,2007,7(增刊):99-102.

[2] Utgof P E. An improved algorithm for incremental induction of decision trees[C]// In: Proceedings of the Eleventh Int. Conference on Machine Learning. New Jersey: IEEE,1994:318-423.

[3] Utgof P E. Incremental Induction of Decision Trees[J]. Machine Learning,1989(4):161-186.

[4] Yin D S,Wang G Y,Yu Y. Data-driven Decision Tree Learning Algorithm Base On Rough Set Theory[C]// In: Proceeding of the third International Conference on MLC. Shanghai: IEEE,2005:2140-2145.

[5] 蔡　晨,李凡长. 动态模糊决策树学习算法研究[J]. 计算机技术与发展,2007,17(7):73-76.

[6] 刘　波,梁活民. 基于增量决策树的快速 IDS 研究与实现[J]. 计算机工程与应用,2008,44(7):141-143.

[7] Pawlak Z. W. Rough Sets [J]. International Journal of information and Computer Science, 1982,11(5):314 -356.

[8] 常志玲,周庆敏. 基于变精度粗糙集的决策树优化算法[J]. 计算机工程与设计,2005,27(17):3175-3177.

[9] Han Jiawei,Kamber M. Data Mining Concepts and Techniques[M]. 北京:机械工业出版社,2001.

[10] 洪家荣,丁明峰,李星原,等 . 一种新的决策树归纳学习算法[J]. 计算机学报,1995,18(6):470-474.

[11] 苗夺谦, 李道国. 粗糙集理论、算法与应用[M]. 北京:清华大学出版社,2008.

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