基于粗集的决策树构建的探讨

杨宝华
(安徽农业大学 信息与计算机学院,安徽 合肥 230036)

摘 要:决策树是对未知数据进行分类预测的一种方法。自顶向下的决策树生成算法关键是对结点属性值的选择。近似精度是RS中描述信息系统模糊程度的参量,能够准确地刻画粗集。文中在典型的ID3算法的基础上提出了基于RS的算法。该算法基于近似精度大的属性选择根结点,分支由分类产生。该算法计算简单,且分类使决策树和粗集更易理解。

关键词:粗集;决策树;近似精度
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2006)08-0083-02

Discussion of Constructing Decision Tree Based on RS

YANG Bao-hua
(College of Information and Computer, Anhui Agriculture University, Hefei 230036, China)

Abstract: The decision tree is a kind of method to classify to predict for the unknown data. The key of policy-making tree production algorithm which is from top to bottom is the pitch point attribute value. The approximation quality describes parameter of the information system fuzzy degree in RS, and it portrays RS accurately. The algorithm on the basis of RS is proposed in this paper on the basis of typical ID3 algorithm, which chooses root on the basis of approximation quality. Classification produces branch. This algorithm is simple in calculation, and classification makes decision tree and RS easy to be understood.

Key words: rough set; decision tree; approximation quality

0 引 言

决策树的表现形式类似于流程图的树结构,在决策树内部结点进行属性值测试,并根据属性值判断由该结点引出的分支,在决策树的叶子结点得到结论。在生成决策树后,可以方便地提取决策树描述的知识,沿着根结点到叶子结点的每条路径对应一条决策规则,并表示成IF-THEN的形式。沿着决策树的一条路径所形成的属性-值偶对就构成了分类规则条件部分(IF部分)中的合取项,叶子结点所标记的类别就构成了规则的结论内容(THEN部分)[1]。

决策树构建的方法很多,构建的重点是找到良好的结点和分支,产生好的规则。典型的算法是ID3[2]算法,该算法是基于信息熵的决策树分类算法,根据属性集的取值选择实例的类别。它采用自顶向下不可返回的策略,搜出全部空间的一部分。ID3算法存在很多不足,文献[3,4]对ID3算法进行了优化。文中基于ID3算法提出了基于RS(Rough Set)的决策树构建方法。

收稿日期:2005-11-23
基金项目:安徽省教育厅资助项目(2003kj117);高校青年基金资助项目(2003)
作者简介:杨宝华(1974-),女,安徽合肥人,讲师,硕士,研究方向为粗糙集、数据挖掘。

1 基础理论

以下基础理论均引自文献[5]。

定义1 信息系统。粗集理论中的信息系统可用一个四元组来表示:$S=\{U,A,V,f\}$。其中U是一个非空集合,表示数据库中的所有记录(Record);A表示数据库中的全部属性(Attribute),如果该信息表同时为决策表,则在A中的属性可以进一步分为条件属性C和决策属性D,$A=C\cup D$;V是属性值组成的集合;f是属性和记录的函数,$f(a,e)$的值确定记录e关于属性a的取值。

定义2 等价类。对于一个信息系统$S=\{U,A,V,f\}$中的的一个属性集合$B\subseteq A$,如果满足$IND(B)=\{(x,y)\in U\times U\mid a(x)=a(y)$,任意$a\in B\}$,则称等价关系$IND(B)$为不分明关系(Indiscernibility Relation)。$U/IND(B)$表示关系$IND(B)$上所有等价类的集合;$B(x)$表示对象x所在的等价类。

定义3 上近似集和下近似集。每一个不确定概念由一个对称为上近似和下近似的精确概念来表示:设给定知识库$K=(U,R)$,对于每个子集$X\in U$和一个等价关系$R\in IND(K)$,可以根据R的基本集合描述来划分集合X:

$$\underline{R}(X)=\bigcup\{r\in U/IND(R):r\subset X\}$$

$$\overline{R}(X)=\bigcup\{r\in U/IND(R):r\cap X\neq\varnothing\}$$

式中$\underline{R}(X)$和$\overline{R}(X)$分别称为X的R上近似和R下近

似,集合的下近似是包含给定集合中所有基本集的集合,集合的上近似是包含给定集合元素中所有基本集的最小集合。

定义 4 由等价关系 R 定义的集合 X 的近似精度为:$\alpha_R(X)=\frac{|\underline{R}X|}{|\overline{R}X|}$,其中 $X\neq\emptyset$,$|X|$表示集合 X 的基数,显然 $0\leqslant\alpha_R(X)\leqslant1$。

$\alpha_R(X)$反映了利用知识 R 近似表示 X 的完全程度。当 $\alpha_R(X)=1$,X 是 R 精确集合;当 $0<\alpha_R(X)<1$,X 是 R 的粗集合;$\alpha_R(X)=0$,X 是 R 的不确定集合。

定义 5 规则支持度(support)及置信度(confidence)。设规则形式为:IF $[X]_{Rf}$ THEN D。则其支持度 SD 及置信度 CD 可定义[6]为:

$$SD=\frac{|[X]_{Rf}\cap D|}{|D|},CD=\frac{|[X]_{Rf}\cap D}{|[X]_{Rf}|}$$

$|[X]_{Rf}|$表示挖掘数据库中条件属性满足该规则的前提条件的记录数,$|D|$表示数据库结论属性满足该规则结论条件的记录数。由上述公式可以看出,支持度反映了该分类规则在统计意义上的可靠性,置信度反映了使用该规则进行分类的准确程度。

2 算法描述

2.1 算 法

在粗集中表示集合 X 不精确性的数值并不是人为给定的,而是通过现有知识中的精确集合定义的。产生不精确性的原因在于对论域的现有知识有限,随着知识粒度的细化,不确定性会随之降低。刻画粗集的方法很多,常用近似精度的数值表示粗集的数字特征。近似精度是粗集理论中描述信息系统模糊程度的参量,一般是相对条件属性的一个子集。基于粗集的决策树建立算法如下:

(1)对数据表进行约简。

(2)在约简表中选择近似精度大的属性作为根结点,if 各个属性的近似精度相等且不为 0,then 选择属性复合值最小的属性。

(3)选择分支,在余下的对象中对属性进行分类,若决策属性惟一,产生叶子结点;

每个叶子结点就是一条规则。

2.2 实 例

如文献[2]中表 1 所示,条件属性 C = {Outlook, Temperature, Windy, Humidity},决策属性 D = {Class}。对该信息表进行约简结果是 C' = {Outlook, Humidity, Windy},计算上下近似:

$\underline{R}$ (Outlook) = {3,7,12,14},$\underline{R}$ (Humidity) = $\emptyset$,$\underline{R}$ (Wind) = $\emptyset$

$\overline{R}$ (Outlook) = $\overline{R}$ (Humidity) $\overline{R}$ (Windy) = {1,2,3,4,5,6,7,8,9,10,11,12,13,14}

$$\alpha_R(\text{Outlook})=\frac{4}{14}$$

$$\alpha_R(\text{Humidity})=\alpha_R(\text{Windy})=0$$

Outlook = {overcast} 的对象都有明确的分类结果 Class = {play},说明可以用 Outlook 近似表示该决策表的决策的程度最高,并且该决策表是粗集。所以选择属性 Outlook 作为根结点,对于 Outlook = {sunny} 和 Outlook = {rain} 的决策属性不惟一,即不确定,所以该结点继续分支,分别进行分类,直到决策属性惟一。

Outlook = {sunny} = {1,2,8,9,11}

Humidity = {{1,2,8}{9,11}}

Windy = {1,8,9},{2,11}

在 Outlook = {sunny} 时,Humidity 分类后决策属性惟一,作为叶子结点。

Outlook = {rain} = {4,5,6,10,14}

Humidity = {{4,14}{5,6,10}}

Windy = {4,5,10},{6,14}

在 Outlook = {rain} 时,Windy 分类后决策属性惟一,作为叶子结点。构建的决策树如图 1 所示。

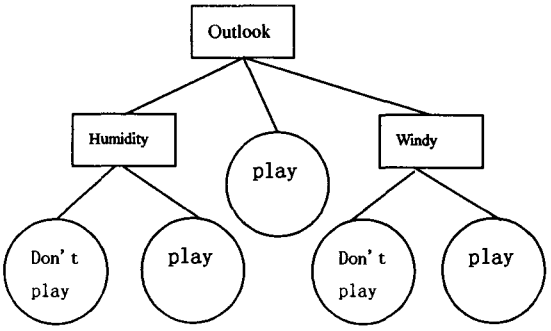

图 1 决策树示意图

由图 1 所示的决策树从根结点到叶子结点共产生 5 条规则如下:

①If Outlook = {overcast} Then Class = {play}

原数据表中有 4 条该规则,所以 CD = 1, SD = 4/14 = 28.6%。

②If Outlook = {sunny} and Humidity = {high} Then Class = {Don't play}

原数据表中有 3 条该规则,所以 CD = 1, SD = 3/14 = 21.4%。

③If Outlook = {sunny} and Humidity = {nomal} Then Class = {play}

原数据表中有 2 条该规则,所以 CD = 1, SD = 2/14 = 14.3%。

④ If Outlook = {rain} and Windy = {true} Then Class = { Don't play}

原数据表中有 2 条该规则,所以 CD = 1, SD = 2/14 = 14.3%

⑤If Outlook = {rain} and Windy = {false} Then Class = {play}

原数据表中有 3 条该规则,所以 CD = 1, SD = 3/14 = 21.4%。

(下转第 87 页)

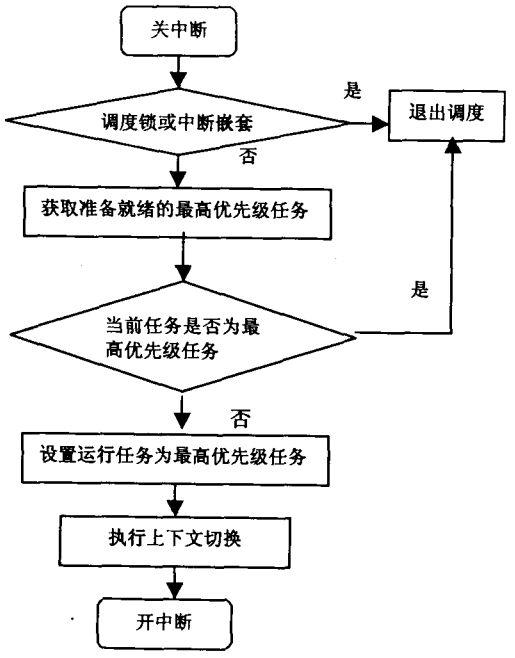

图 2　OSSched()函数流程

对于定时中断函数 OSTickISR(),它主要负责中断进入时保存处理器寄存器内容,完成任务切换退出时恢复处理器寄存器内容并返回,相当于中断服务程序的入口。

中断级的上下文切换是 OSIntExit()通过调用 OSIntCtxSw()来执行切换功能。下面重点谈谈 OSIntExit()函数过程。与任务级切换不同的是中断级切换时,中断返回函数将决定是返回到被中断的任务,还是让优先级最高任务运行。其流程如图 3 所示。

3　结束语

以上是一些经验与体会,笔者曾在研制电力谐波检测仪的项目中,根据嵌入式操作系统 μC/OS 的内核运行机制原理,在 TI DSP F2407 芯片上移植取得很好的效果。现在网站上有很多关于各种类型芯片的移植程序,但不仅要知其然,更要知其所以然,才能做到举一反三、灵活运用,所以有必要掌握 μC/OS 内核的运行基本原理,从而能深入理解移植程序,达到了解嵌入式操作系统原理的目的,为从事各种嵌入式操作系统的研究打下坚实的基础。

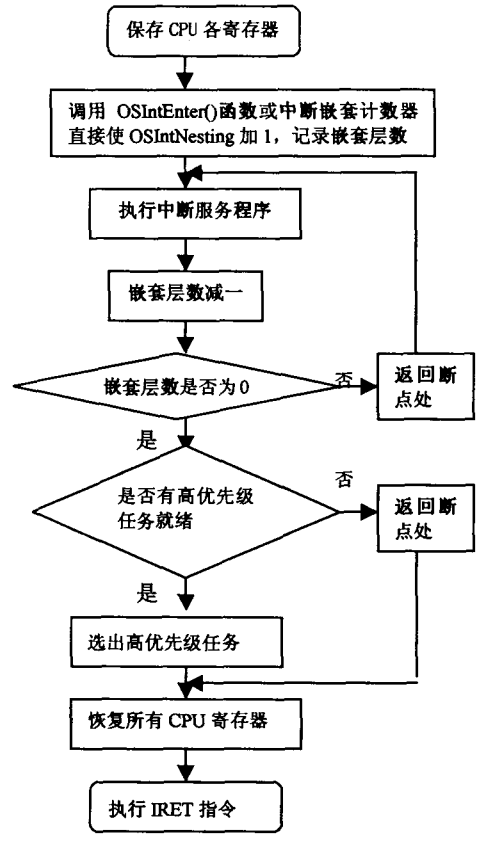

图 3　中断返回函数流程

参考文献:

[1] 邵贝贝.μC /OS 源码公开的实时嵌入式操作系统[M].北京:中国电力出版社,2001.

[2] 崔树林.嵌入式系统通用的应用软件结构研究[J].单片机与嵌入式系统应用,2003(8):9-10.

[3] 王劲松.嵌入式操作系统 uc/os 的内核实现[J].现代电子技术,2003(8):48-49.

[4] 罗　蕾.嵌入式实时操作系统及应用开发[M].北京:北京航空航天出版社,2005.

[5] 王克星.实时多任务操作系统的开发与应用[J].计算机工程与应用,2003(5):132-134.

(上接第 84 页)

3　小　结

决策树的建立可以使数据规则可视化,结构清晰,所以在对知识的分类中常常用决策树表示。利用典型 ID3 算法构造决策树,按照信息增益最大的原则,相对抽象,计算繁琐;利用粗集近似精度选择根结点时计算简单,而且分类可以使决策树和粗集更容易理解。

参考文献:

[1] 朱　明.数据挖掘[M].合肥:中国科学技术大学出版社,2002.

[2] Quinlan J R. Induction of decision trees[J]. Machine Learning,1986(1):81-106.

[3] 王静红,王熙照,邵艳华,等.决策树算法的研究及优化[J].微机发展,2004,14(9):30-32.

[4] 尹阿东,郭秀颖,宫　雨,等.增量决策树算法研究[J].微机发展,2005,15(2):63-66.

[5] 曾黄麟.粗糙集理论及其应用[M].重庆:重庆大学出版社,1998.

[6] 董祥军,宋瀚涛,姜　合,等.时态关联规则的研究[J].计算机工程,2005,15:24-26.