决策树算法及其核心技术

杨学兵,张 俊
(安徽工业大学 计算机学院,安徽 马鞍山 243002)

摘 要:决策树是归纳学习和数据挖掘的重要方法,通常用来形成分类器和预测模型。概述了决策树分类算法,指出了决策树算法的核心技术:测试属性的选择和树枝修剪技术。通过对当前数据挖掘中具有代表性的优秀分类算法进行分析和比较,总结出了各种算法的特性,为使用者选择算法或研究者改进算法提供了依据。最后,通过一个实例说明决策树分类在实际生产中的应用。

关键词:决策树;测试属性;树枝修剪
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2007)01-0043-03

Decision Tree and Its Key Techniques

YANG Xue-bing, ZHANG Jun
(School of Computer Science, Anhui University of Technology, Maanshan 243002, China)

Abstract: Decision tree is an important method in induction learning as well as in data mining, which can be used to form classification and predictive model. Introduces decision tree and points out its key techniques: the choice of testing feature and tree pruning. It summarizes the main features of every algorithm by analyzing and comparing a variety of typical classifiers to provide a basis for selecting or improving the algorithms in data mining. Finally, through an instance, this paper shows the application of decision tree in production.

Key words: decision tree; testing feature; tree pruning

1 决策树基本概念

决策树是一树状结构,它从根节点开始,对数据样本(由实例集组成,实例有若干属性)进行测试,根据不同的结果将数据样本划分成不同的数据样本子集,每个数据样本子集构成一子节点。它是通过一系列规则对数据进行分类的过程。它提供一种在什么条件下会得到什么值的类似规则的方法。

决策树分为分类树和回归树两种[1],分类树对离散变量做决策树,回归树对连续变量做决策树。一般的数据挖掘工具,允许选择分裂条件和修剪规则,以及控制参数(最小节点的大小,最大树的深度等等)来限制决策树。决策树作为一棵树,树的根节点是整个数据集合空间,每个分节点是对一个单一变量的测试,该测试将数据集合空间分割成两个或更多块。每个叶节点是属于单一类别的记录。构造决策树的过程为:首先寻找初始分裂。整个训练集作为产生决策树的集合,训练集每个记录必须是已经分好类的。决定哪个属性域(Field)作为目前最好的分类指标。一般的做法是穷尽所有的属性域,对每个属性域分裂的好坏做出量化,计算出最好的一个分裂。不同的算法的计算属性域分裂的标准也不太相同。其次,重复第一步,直至每个叶节点内的记录都属于同一类,增长到一棵完整的树。构造决策树的目的是找出属性和类别间的关系,用它来预测将来未知类别的记录的类别。这种具有预测功能的系统叫决策树分类器。

2 常用的决策树算法

决策树分类算法从提出以来,出现了很多算法,比较常用的有:1986年 Quinlan 提出了著名的 ID3[2]算法。ID3算法体现了决策树分类的优点:算法的理论清晰,方法简单,学习能力较强。其缺点是:只对比较小的数据集有效,且对噪声比较敏感,当训练数据集加大时,决策树可能会随之改变,并且在测试属性选择时,它倾向于选择取值较多的属性。

在ID3算法的基础上,1993年 Quinlan 又自己提出了改进算法——C4.5[3]算法。为了适应处理大规模数据集的需要,后来又提出了若干改进的算法,其中 SLIQ[4](supervised learning in quest)和 SPRINT[5](scalable parallelizable induction of decision trees)是比较有代表性的两个算法, PUBLIC[6](Pruning and Building Integrated in Classification)算法是一种很典型的在建树的同时进行剪枝的算法。此外,还有很多决策树分类算法。

收稿日期:2006-04-18
基金项目:安徽省教育厅自然科学基金重点资助项目(2004KJ053ZD)
作者简介:杨学兵(1967-),男,安徽和县人,副教授,博士,研究方向为人工智能。

3 决策树算法的核心技术

建立决策树的目标是通过训练样本集,建立目标变量关于各输入变量的分类预测模型,全面实现输入变量和目标变量不同取值下的数据分组,进而用于对新数据对象的分类和预测。当利用所建决策树对一个新数据对象进行分析时,决策树能够依据该数据输入变量的取值,推断出相应目标变量的分类或取值。决策树技术中有各种各样的算法,这些算法都存在各自的优势和不足。目前,从事机器学习的专家学者们仍在潜心对现有算法的改进,或研究更有效的新算法。总结起来,决策树算法主要围绕两大核心问题展开:第一,决策树的生长问题,即利用训练样本集,完成决策树的建立过程。第二,决策树的剪枝问题,即利用检验样本集,对形成的决策树进行优化处理。以下将主要就这两方面的问题进行论述。

3.1 决策树的生长过程

决策树的生产过程本质是对训练样本集的不断分组过程。决策树上的各个分枝是在数据不断分组的过程中逐渐生长出来的。决策树生长的核心技术是测试属性选择问题。

最初的 ID3 算法用一个叫做增益标准来选择需要检验的属性,它基于信息论中熵的概念。设 S 是 s 个数据样本的集合。假定类标号属性具有 m 个不同值,定义 m 个不同类 $C_i(i=1,\cdots,m)$。设 s_i 是类 C_i 中的样本数。对一个给定的样本分类所需的期望信息由下式给出:

$$I(s_1,s_2,\cdots,s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

其中 $p_i = s_i/s$ 是任意样本属于 C_i 的概率。注意,对数函数以 2 为底,其原因是信息用二进制编码。

设属性 A 具有 v 个不同值 $\{a_1,a_2,\cdots,a_v\}$。可以用属性 A 将 S 划分为 v 个子集 $\{S_1,S_2,\cdots,S_v\}$,其中 S_j 中的样本在属性 A 上具有相同的值 $a_j(j=1,2,\cdots,v)$。设 s_{ij} 是子集 S_j 中类 C_i 的样本数。由 A 划分成子集的熵或信息期望由下式给出:

$$E(A) = \sum_{j=1}^{v} ((s_{1j}+s_{2j}+\cdots+s_{mj})/s) * I(s_{1j}+s_{2j}+\cdots+s_{mj})$$

熵值越小,子集划分的纯度越高。对于给定的子集 S_j,其信息期望为

$$I(s_{1j}+s_{2j}+\cdots+s_{mj}) = -\sum_{i=1}^{m} p_{ij}\log_2(p_{ij})$$

其中 $p_{ij} = s_{ij}/s_j$ 是 S_j 中样本属于 C_i 的概率。

在属性 A 上分枝将获得的信息增益是:

$$\text{Gain}(A) = I(s_1, s_2,\cdots,s_m) - E(A)$$

C4.5 算法在决策树各级结点上选择属性时,用增益比率(gain ratio)作为属性的选择标准。

$$\text{SplitInformation}(A,S) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|}\log_2\frac{|S_i|}{|S|}$$

$$\text{GainRatio}(A,S) = \frac{\text{Gain}(S,A)}{\text{SplitInformation}(S,A)}$$

SLIQ,SPRINT,PUBLIC 算法中,使用 gini 指标(gini index)代替信息量(Information)作为属性选择的标准。gini 指标比信息量性能更好,且计算方便。对数据集包含 n 个类的数据集 S,gini(S) 定义为:

$$\text{gini}(S) = 1 - \sum p_j * p_j$$

式中,p_j 是 S 中第 j 类数据的频率。gini 越小,Information Gain 越大。

3.2 决策树的修剪

决策树的修剪是针对训练数据过分近似问题而提出来的,修剪方法通常利用统计方法删去最不可靠的分支(树枝),以提高今后分类识别的速度和分类识别新数据的能力。其实质是消除训练集中的异常和噪声。

通常采用两种方法进行树枝的修剪,它们分别是[7]:

(1)事前修剪(prepruning)方法。该方法通过提前停止分支生成过程,即通过在当前节点上就判断是否需要继续划分该节点所含训练集来实现。一旦停止分支,当前节点就成为一个叶节点,该叶节点中可能包含多个不同类别的训练样本。在建造一个决策树时,可以利用统计上的重要性检测 χ^2 或信息增益等来对分支生成情况进行评估。如果在一个节点上划分样本集时,会导致节点中样本数少于指定的阈值,则要停止继续分解样本集合。但确定这样一个合理的阈值常常也比较困难,阈值过大会导致决策树过于简单化,而阈值过小又会导致多余树枝无法修剪。

(2)事后修剪(postpruning)方法。该方法从一个"充分生长"树中,修剪掉多余的树枝。基于代价成本的修剪算法就是一个事后修剪方法,被修剪的节点就成为一个叶节点,并将其标记为它所包含样本中类别个数最多的类别。而对于树中每个非叶节点,计算出若该节点被修剪后所发生的预期分类错误率;同时根据每个分支的分类错误率,以及每个分支的权重,计算若该节点不被修剪时的预期分类错误率;如果修剪导致预期分类错误率变大,则放弃修剪,保留相应节点的各个分支,否则就将相应节点分支修剪删去。在产生一系列经过修剪的决策树候选之后,利用一个独立的测试数据集,对这些经过修剪的决策树的分类准确性进行评价,保留下预期分类错误率最小的决策树。除了利用预期分类错误率进行决策树修剪之外,还可以利用决策树的编码长度来进行决策树的修剪。所谓最佳修剪树就是编码长度最短的决策树,这种修剪方法利用最短描述长度(Minimum Description Length ,简称 MDL)原则来进行决策树的修剪。该原则的基本思想就是:最简单的就是最好的。与基于代价成本方法相比,利用 MDL 进行决策树修剪时无需额外的独立测试数据集。当然事前修剪可以与事后修剪相结合,从而构成一个混合的修剪方法。事后修剪比事前修剪需要更多的计算时间,从而可以获得一个更可靠的决策树。

4 决策树算法的应用

决策树分类可应用在钢铁厂轧辊选择中,用于决策是否更换某一轧辊的情况,在钢铁厂中,轧辊是易磨损的,且

价格比较高,需要经常更换,使用成本比较高。而且一旦轧辊出了问题,可能会造成很大的损失。正确的决策是否更换某一轧辊,使得公司的效益最大化,具有重要的现实意义。把以往的更换情况整理的数据作为训练集,然后对影响是否更换的相关特征进行数据挖掘,从而可得到对轧辊的选择的决策进行指导的有意义的知识。在进行挖掘前,首先对数据进行清理,可采用平滑技术消除或减少噪声,用该属性最常用的值处理空缺值。用决策树算法进行分类,要求处理连续属性和离散属性。在选择中,轧辊受役龄、价格、是否关键部件和磨损程度等多种因素影响,通过分类算法得到决策树,用来决策正在生产线上的某一轧辊是否需要更换。

表 1 中的数据是从钢铁厂的轧辊更换的数据库中抽取出的部分数据,含有 5 个属性:役龄、价格、是否关键部件、磨损程度和是否更换。利用这样的少量的数据来说明决策树分类在钢铁厂轧辊选择中的应用。

表 1　轧辊更换情况数据库训练数据

役龄	价格	是否关键部件	磨损程度	是否更换
<=5	高	否	一般	否
<=5	高	否	好	否
5~10	高	否	一般	否
>=10	中	否	一般	是
>=10	低	是	一般	是
5~10	中	否	好	否
5~10	高	是	一般	是
<=5	中	否	一般	否
5~10	中	否	好	否
>=10	高	是	好	是
5~10	低	是	一般	是
<=5	中	是	一般	是
<=5	低	是	一般	是
>=10	中	是	好	是

使用信息增益进行属性选择:

更新的备件数为 s_1,不更新的备件数为 s_2。

$I(s_1, s_2) = I(9,5) = 0.94$

$E($ 役龄$) = 0.694$

Gain(役龄) = 0.245

同理,Gain(价格) = 0.21,Gain(是否关键部件) = 0.15,Gain(磨损程度) = 0.10

因为 Gain(磨损程度)<Gain(关键部件)<Gain(价格)<Gain(役龄),可以看出以"役龄"这个属性进行训练集分类的信息赢取值最大,于是"役龄"就被选为用于划分的属性,以此类推,可以得到决策树如图 1 所示。

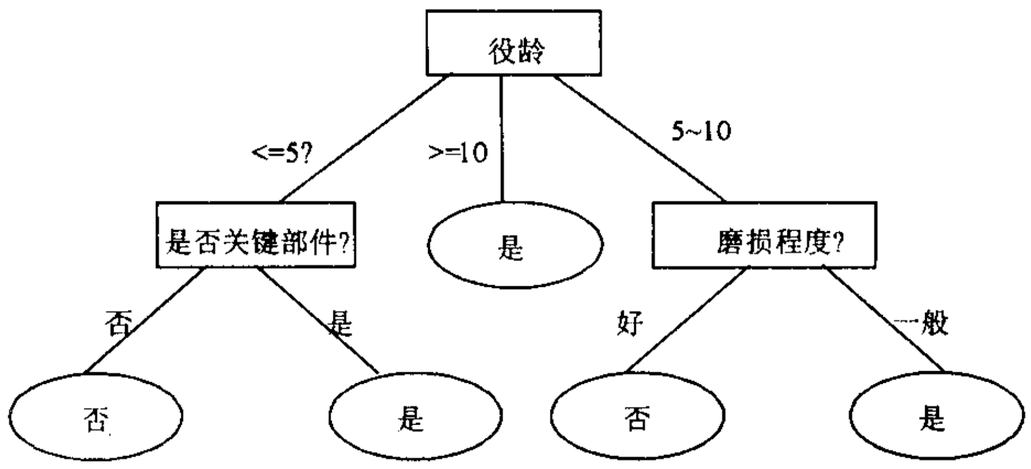

图 1　使用 ID3 算法得到轧辊是否更换问题的决策树

这样的通过训练集得到的决策树分类模型就可以用来对新数据进行分类了,即可以判断生产线上的轧辊是否需要更换了。

参考文献:

[1] Mitchell T M. 机器学习[M]. 北京:机械工业出版社,2004.

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

[3] Quinlan J R. C4.5: Programs for Machine Learning[M]. [s. l.]:Morgan Kaufman,1993.

[4] Mehta M, Agrawal R, Rissanen J. SLIQ: A Fast and Scalable Classifier for Data Mining[M]. US: IBM Almaden Research Center,1996.

[5] Shafer J C, Agrawal R, Mehta M. SPRINT: A Scalable Parallel Classifier for Data Mining[C]//Proc of the 22nd Int Conf on Very Large Databases. Mumbai(Bombay), India: [s. n.], 1996.

[6] Rastogi R, Shim K. PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning[R]. Murray Hill: Bell Laboratories,1998.

[7] Han Jiawei, Kamber M. DATA MINING Concepts and techniques[M]. 北京:高等教育出版社,2001.

(上接第 42 页)

2005.

[3] Soliman H, Castelluccia C, Malki K E, et al. RFC4140 Hierarchical Mobile IPv6 Mobility Management (HMIPv6)[S]. 2005.

[4] Braden R, Clark D, Shenker S. RFC1633 Integrated Services in the Internet Architecture: an Overview[S]. 1994.

[5] Blake S, Black D, Carlson M, et al. RFC2475 An Architecture for Differentiated Service[S]. 1998.

[6] Rosen E, Viswanathan A, Callon R. RFC3031 Multiprotocol Label Switching Architecture[S]. 2001.

[7] Li T, Rekhter Y. RFC2430 A Provider Architecture for Differentiated Services and Traffic Engineering[S]. 1998.

[8] Deering S, Hinden R. RFC 2460 Internet Protocol, Version 6 (IPv6) Specification[S]. 1998.

[9] 刘　沙,杨寿保. 基于 F－HMIPv6 服务质量保证[C]//中国计算机用户协会网络分会 2003 年年会. 海南:[出版者不详],2003:252－258.

[10] Talukdar A K, Badrinath B R, Acharya A. MRSVP: A Reservation Protocol for Integrated Services Packet Networks with Mobile Hosts[J]. Wireless Networks,2001(7):5－19.

[11] Tseng Chien－Chao, Lee Gwo－Chuan, Liu Ren－shiou. HMRSVP: A Hierarchical Mobile RSVP Protocol[J]. ACM Wireless Networks,2003(2):95－102.