山东农业大学学报 JOURNAL OF SHANDONG AGRICULTURAL UNIVERSITY
(自然科学版)，2008，39(3)：461-464

C4.5数据挖掘算法的改进及其应用

袁爱香
(山东警察学院现代教育中心，山东 济南 250014)

摘要：简要介绍数据挖掘算法中的C4.5算法的基本思想，并在分析传统行业税负测算方法的基础上，结合税收行业领域的应用实际，对C4.5算法进行改进和应用。通过验证表明，改进后的算法运行结果可靠,运行效率提高。
关键词：C4.5算法；数据挖掘；改进应用
中图分类号：TP 311 文献标识码：A 文章编号：1000－2324（2008）03－0461－04
收稿日期：2007-03-17
作者简介：袁爱香（1966-），女，高级工程师,主要研究方向为数据库应用、数据仓库建设等。

A THE IMPROVEMENT AND APPLICATION OF C4.5 ALGORITHM FOR DATA MINING

YUAN AI-xiang
(Modern Education Center of Shandong Police College,Shandong jinan 250014, China)

Abstract:The basic principles of C 4.5 algorithm for data mining are simply introduced in this paper. Then, based on analyzing the calculation methods of tax burden in traditional industrial companies,the paper puts forward with an improvement and application of C 4.5 algorithm combining with the Tax field's practice. Finally ,it is proved that the improved algorithm brings more reliable and more effective results by means of an example.
Key words: C 4.5 algorithm; data mining ;improvement and application

1 引言

决策树和决策规则是解决实际应用中分类问题的数据挖掘方法。一般来说，分类是把数据项映射到其中一个事先定义的类中的学习函数的过程。利用统计方法，通过总结样本集的统计属性，可以实现对分类问题建模，如：贝页斯定理、方差分析、预测分析等都是这种分类方法的典型代表。另一种建模方法则是基于逻辑的，逻辑模型不用加法和乘法这样的算术运算，而是通过对属性值运用布尔型和比较运算进行真或假的评价来表达。和其他非逻辑方法比较起来，这种建模方法的分类结果更为精确，它们包含范围更广的解释性属性。决策树和决策规则是典型的以逻辑模型方式输出分类方法的数据挖掘技术。

所谓决策树，就是在对数据进行决策分类时利用树的结构将数据记录进行分类,其中树的一个叶结点代表符合某个条件的属性集,根据属性的不同取值建立决策树的各个分支;随后递归的构造每个子节点的子树。由于决策树结构简单，便于人们认识理解以及决策树不需要额外的数据训练，因此决策树是数据挖掘中常用的一种分类方法，而现在最常用的是基于信息熵的算法。Quinlan的ID3算法是国际上公认的最具有影响的基于信息熵的决策树算法，它根据属性集的取值进行分类。

2 C4.5数据挖掘算法简介

C4.5算法是ID3算法的改进，它采用了一种归纳学习的机制。该算法的属性选择是基于一个假设，即：决策树的复杂度和所给属性值表达的信息量是密切相关的。C4.5把分类范围扩展到了数字属性，这个度量标准倾向于能把数据分区成有低类熵的子集的属性，即大部分样本都属于一个单独的类。C4.5算法的基本过程包括决策树生成、修剪决策树、生成决策规则、等几个部分，一般来说，决策树算法为了满足基于归纳学习的方法，必须满足几个关键性的要求：

(1)属性值描述——要分析的数据必须是平面文件形式，一个对象或样本的所有信息必须以属性的固定集来表达。每个属性可能有离散型值或数值型值，但是用于描述样本的属性，必须是彼此不同。这样可以消除样本内有内在的可变结构的区域。

(2)预先定义的类——要分配给样本的类必须事先建立，用机器学习的术语来说就是有指导的学习。

(3)离散的类——这些类必须被严格的描述：一个对象要么属于一个具体的类，要么不属于。样本的个数要远远超过类的个数。

(4)充足的数据——以决策树形式给出的归纳总结从识别数据中的模式开始，若能从偶然重合中辨别出足够多的健壮的模式，这个方法是有效的，由于该辨别通常是依赖于统计检验，因此，必须有足够的样本才能使这些检验生效。

(5)“逻辑”分类模型——这些方法仅构造能够用于决策树或决策规则表达的分类器。

这些形式实质上限制了类的描述必须是关于个别属性值的陈述的逻辑表达。

C4.5算法的生成决策树的基本思想如下：

C4.5算法的属性选择的基础是基于使生成的决策树中节点所含的信息熵最小。所谓熵在系统学上是表示事物的无序度。不难理解熵越小则记录集合的无序性越小，也就是说记录集合内的属性越有顺序，越具有一致性，这也正是我们所追求的目标。集合S的熵的计算公式如下：[1]

$$Info(S)=-\sum((freq(Ci,S)/|S|).(log2(freq(Ci,S)/|S|))$$

其中freq(Ci,S)代表集合S中属于类Ci(k个可能类中的一个)的样本数量，|S|表示集合S中的样本数量。上面的

公式仅仅给出了一个子集的熵的计算,如果我们按照某个属性进行分区后就涉及到若干个子集,我们需要对这些子集进行熵的加权和的计算,公式如下所示:

Infox(T) =－Σ((|Ti|/|T|).Info(Ti)) 其中T是按照属性x进行分区的集合。为了更加明显的比较不同集合的熵的大小，我们计算分区前的集合的熵和分区后的熵的差(我们把这个差叫做增益),增益大的就是我们要选取的节点。公式如下：

Gain(X) = Info(T)－Infox(T)

C4.5算法包含3种类型的检验结构：

(1)离散值得标准检验，对属性的每个可能值有一个分枝和输出；

(2)如果属性Y有连续的数值，通过该值和阀值Z进行比较，用输出Y<=Z和Y>=Z定义二元检验；

(3)基于离散值的更复杂的检验，该检验中属性的每个可能值被分配到许多易变的组中，每个组都有一个输出和分枝。

对于数值型的属性检验，C4.5算法有一个计算最优阀值的算法：

首先根据正在考虑的属性Y对训练样本进行分类。这些值的数量是有限的，因此我们按分类顺序用{v1，v2，...,vm}表示。位于vi和vi+1之间的阀值与把样本按属性Y分成{v1，v2，...,vi}和{vi+1,vi+2,...,vm}有同样的效果，因此，对Y仅有m-1个分区，要系统地检查所有的分区以求得最优分区。通常选择每个分区中间点(vi+vi+1)/2作为代表阀值，而C4.5算法的不同之处在于它选择每一个区间{vi,vi+1}的最小值vi作为阀值，而不是中间点，这确保了出现在最终决策树或规则中的阀值都实际存在于数据库中。

决策书修剪的主要任务是抛弃一个或更多的子树，并用叶替换这些子树，使决策树简单化。在具备可用的训练样本和检验样本的情况下,决策树修剪的基本思想是去掉那些对未知检验样本的分类精度没有帮助的部分树，生成一个更简单、更容易理解的树。主要有两种改进的递归分区方法，C4.5遵循后修剪方法。

大型的决策树理解起来有困难，为了使决策树模型更容易读，可以把到达每个叶的路径转换成IF-THEN生成规则。IF部分包括一条路径的所有检验，THEN部分是最终分类。C4.5用改进方法并仅仅选择包含最多训练样本而不是被任何规则包含的类来作为默认类。这个过程是迭代的，从每个值表示一个组的初始分布开始，然后每次再进行新的迭代，分析把两个以前的组合并成一个组的概率。如果信息增益率是非递减的，接受该合并。最终的结果可能是两个和更多的组，这样就在决策书和决策规则的基础上简化了分类模型。

基于信息熵的ID3算法、C4.5算法虽然能有效地生成决策树，但是随着训练样本集中样本个数的不断增多，即样本集规模的不断扩大，训练样本集在主存中的换进换出消耗了大量的时间，严重影响了算法的效率。如何使算法能够有效处理大规模的训练样本集，成为决策树算法研究的一个重要问题。下面以行业税负率的测算为例，介绍一种结合领域知识对C4.5算法进行改进应用的方法。

3 算法改进

传统的行业税负测算方法一般建立在对抽取样本的平均值的计算上，通过这种方法进行计算，效率高、易理解，容易被使用人员所接受。但其缺点是抽样带有很大的偶然性，即使是将所有样本纳入计算也很难获得一个接近于实际的理论值，而只能是不同样本的平均值。如果参与计算的样本中存有未照章纳税的企业有漏税行为，则计算得到的平均值就会低于实际值。因此，采用科学的方法选取正确的样本是税负计算中的关键。为了找到一类能合理代表税负率的样本，我们将数据挖掘中的决策树方法引入到了增值税税负分析中。

挖掘算法的确定基于以下两个前提：

(1)按照税制理论，增值税税收负担不受生产结构变化和生产经营环节多少的影响。就一项货物或劳务而言，只要最后销售价相同，不论它经过多少道生产经营环节，也不论是一个或是多个单位生产经营，该货物或劳务负担的增值税额是相同的。[3] 因此，就同一个行业来说，由于市场竞争日趋激烈价格日益透明，从而导致了同类商品的售价越来越接近，故其税负率具有一致性趋势。

(2)任何企业在未经合理评估和检查之前都无法确定其是否照章纳税，因此不宜直接选用其为测算税负率的样本。

基于信息论的方法坚持对数据库中一个样本进行分类时所做检验的数量最小。由于全省有将近10万户增值税一般纳税人共分布在93个行业大类中，如果对每个行业m个样本进行m-1个阀值反复比较最高信息增益，其计算量很大，难以满足实时查询的需要。因此，如何提高算法的计算效率是一个难点问题。

理论上，每个行业的真实税负率水平应该高于所有样本的平均税负率水平，除非该行业的所有样本无一有偷漏税行为发生（实际上是不可能的），这就给我们改进算法提供了可能。算法改进的主要一点就是要充分利用所有样本的平均税负率这一个值，具体处理流程描述如下：

(1)选择某一行业所有样本；

(2)根据是否查补、出口企业将样本分为两类；

(3)计算所选样本税负率的平均值；

(4)计算以(3)中税负率平均值为阀值进行分类所得到集合的熵；

(5)除掉低于税负率平均值的样本，对高于平均税负率的样本重复步骤(3)、(4)并计算信息增益；

(6)直到信息增益大于0止

算法的改进在于步骤（5），考虑到行业实际税负率应该高于平均税负，因此可以对低于平均税负率的样本进行大胆舍弃，这使得整个算法的执行不是按照一个线形顺序进行，而是一个跳跃性的过程。这种处理方式大大减少了算法的样本检测量，在后面的迭代中加快了样本聚集于某特定熵值的速度，计算性能明显提高。

下面是某市15316个样本中样本数量最多的一个行业——批发业的税负率测算过程：

(1)批发业样本数量为3367；

(2)其中属于查补或出口企业的样本数为522，非查补或出口企业的样本数为2845；

(3)所选样本的平均税负率为0.9733；

(4)以“是否为查补出口企业”属性分类的初始熵计算如下：

Info(查) =－522/3367.log2(522/3367)－ 2845/3367.log2(2845/3367= 0.6222比特

再以平均税负率为阀值分类后初始熵计算如下：

Info(税负率) = 522/3367.(－221/522.log2(221/522)－ 301/522.log2(301/522))+2845/3367.(-1211/2845.log2(1211/2845)-1634/2845.log2(1634/2845))=0.9836比特

(5)算法对高于平均税负0.9733的样本进行一次迭代后得到了增益值为0.2061，循环终止，此时的税率阀值为2.9008

(6)为了查找是否还有更高增益，我们又进行了一次迭代，结果增益值为0.0181，小于上一次增益，故取上一次迭代的税负率阀值2.9008作为批发行业最终判断企业是否照章纳税的阀值。计算的具体数据见下表：

表 1　税负率阀值计算

Table 1　Tax bearing tate value calculation

迭代次数 Iteration number	阀值 Thresholol level	查补出口企业（样本数量） Mending export enterprise(sample capacity)		非查补出口企业（样本数量） Unmending export enterprise(sample capacity)		熵 Entropy		增益值 Yield value	
		税负率〉阀值 Tax bearing rate〉thres hold level	税负率〈=阀值 Tax bearing rate <=thre shold level	税负率〉阀值 Tax bearing rate〉threshold level	税负率〈=阀值 Tax bearing rate <=threshold level	查补出口属性 Mending export property	税负率属性 Tax bearing rate property	查补出口属性 Mending export property	税负率属性 Tax bearing rate property
0	0.9733	221	301	1211	1634	0.6222	0.9836		
1	2.9008	72	150	262	949	0.6218	0.7775	0.0004	0.2061
2	3.2061	27	46	104	157	0.7573	0.9655	-0.1351	0.0181

4　算法验证

验证分为算法结果验证和算法计算性能验证两部分，验证基于下列考虑：

(1)算法改进不应该影响最终的迭代结果，因此，通过改进后的算法所取得的最终结果应与算法改进前的结果接近或一致。

(2)在保证结果的基础，计算效率应有所提高，否则改进是无必要的。

我们首先对算法的准确性进行了验证。仍以上述批发行业的样本选择为例，按照所有样本进行检测计算,算法经过了3024次迭代之后终止，此时的税负率阀值为2.8990，结果集中的样本数量为340个，其中查补出口企业样本量为73个，非查补样本量为267个。由此可见，改进后算法产生的结果集（72+262=334个）比改进前的算法减少了6个样本，税负率阀值2.9008比改进前的算法所得到的阀值2.8990高出0.0018。差异产生的主要原因是，改进前的算法是以每个实有样本的税负率作为阀值进行熵计算和比较的，而改进后的算法是用样本的平均值进行比较计算的，因此，平均值不一定恰好等于集合中某个样本的实际税负，从而导致了样本数量及税负率阀值的略微差异。但从税负率阀值的差异率来看，0.0018/2.8990结果近似为0，因此，改进后算法所产生的结果是可靠的。

再看算法计算效率的测试，改进前的算法共进行了3024次迭代，共耗时5 s；改进后的算法共进行了2次迭代和3次平均值的计算，共耗时2秒，效率提高了1.5倍。

5　结语

需要说明的是，由于改进前的算法效率与样本量的多少有关，因此随着样本量的增加，使用改进后的算法效率提高的幅度会增大。尤其是对样本数量超过百万的大型数据的计算，上述算法的计算性能可以提高10倍以上。通过类似的算法不但可以找出代表行业实际税负的样本集合，还可以找出低于行业实际税负的不同类型的子集，因为通过这种方法所选取的样本集合具有最大程度的内在有序性和相似性，可能暗示着一些同一的税负低的原因。可以看出结合不同的行业领域知识对算法进行改进，不仅可以提高运算效率，还可以收到意外的效果。

参考文献

[1] 闪四清.数据挖掘概念、模型、方法和算法[M].北京：清华大学出版社，2003

[2] 汤贡亮.中国税收制度与管理[M].北京：清华大学出版，2005.1

[3] 马秀红，宋建设，董宴飞.数据挖掘中决策树的探讨[J].计算机工程与应用，2004，1：59-62

返回