决策树分类方法研究

The Study of Decision Tree Classifying Method in Datamining

魏晓云　（四川行政学院计算机系 四川 成都 610072）

摘要：分类知识的获取是数据挖掘要实现的重要任务之一，其核心问题是解决分类模型的构造和分类算法实现。本文以决策树分类方法中有代表性的方法 C 4.5 为例，介绍数据挖掘中一种分类方法—决策树分类方法及其构建和应用研究。

关键词：决策树 数据挖掘 C4.5

1 引言

从数据中生成分类器的有效方法是建立决策树（Decision Tree）。决策树表示方法是应用最广泛的逻辑方法之一，它从一组无次序、无规则的事例中推理决策树表示形式的分类规则。决策树分类方法采用至顶向下的递归方式，在决策树的内部结点进行属性值的比较，并根据不同的属性值判断从该结点向下的分支，在决策树的叶结点得到结论。从决策树的根到叶结点的一条路径就对应着一条合取规则，整棵决策树就对应着一组析取表达式规则。

2 决策树的基本概念

决策树分为分类树和回归树两种，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。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。

在分类分析中，决策树模型是最受欢迎的模型，最主要的原因在于它能非常方便地用图形化（树型结构）的方式表现挖掘的结果，适应于企业管理部门做出决定。1986 年 J. Ross Quinlan 在 Machine Learning Journal 发表了题为“Induction of Decision Trees”的论文，引入了一种新的 ID3 算法。在此基础上，他又对 ID3 算法进行了补充和改进，提出了更先进的 C4.5 算法。本文讨论如何应用 C4.5 算法构造客户分类决策树。

3 决策树生成的过程

决策树是一个类似于流程图的数结构，其中每个内部结点表示在一个属性上的测试，每个分支代表一个测试输出，而每个树叶点代表类或类分布。

（1）收集客户信息，对数据信息进行合并，形成结构统一的客户信息数据源。

（2）对数据源进行数据预处理，去掉与决策无关的属性和高分支属性、将数值型属性进行概化以及处理含空缺值的属性，形成决策树的训练集。

（3）对训练集进行训练，计算每个属性的信息增益和获取率，选择获取率最大的但同时获取的信息增益又不低于所有属性平均值的属性，作为当前的主属

性节点，为该属性的每一个可能的取值构建一个分支。对该子结点所包含的样本子集递归地执行上述过程，直到子集中的数据记录在主属性上取值都相同，或没有属性可再供划分使用，生成初始的决策树。

（4）对初始决策树进行树剪枝。主要采用后剪枝算法对生成的初始决策树进行剪枝，并在剪枝过程中使用一种悲观估计来补偿树生成时的乐观偏差。

（5）由所得到的决策树提取分类规则。对从根到树叶的每一条路径创建一个规则，形成规则集。将规则集显示给用户，把用户筛选过认为可行的规则存入规则数据库。

（6）当新客户在公司进行注册时，系统运用决策树所得规则对新客户的数据信息进行分析，预测该客户的行为属于哪一等级，从而为公司的客户营销策略提供辅助决策。

4 相关算法

描述 1：设信源 X 的符号取值集合为 $A=\{a_1,a_2,\cdots a_n\}$，其中信号 $a_i\in A$ 的出现概率为 $p_i=P[X=a_i]$ $(i=1,2,\cdots,n)$，称 $I(a_i)=\log\frac{1}{p_i}=-\log p_i$ 为 a_i 的信息量。信息量的数学期望值为信源的平均信息量或信息熵，简称为熵（entropy），记为 $H(X)$ 或 $H_n(p_1,p_2,\cdots,p_n)$，则有：

$$H(X)=\sum_{i=1}^{n}p_i\log\frac{1}{p_i}=-\sum_{i=1}^{n}p_i\log p_i$$，其中 $0\leqslant p_i\leqslant 1,\sum_{i=1}^{n}p_i=1$

描述 2：假设训练集 T 包含 n 个样本，这些样本分别属于 m 个类，其中第 i 个类在 T 中出现的比例为 p_i，那么的信息熵为：

$$I(T)=\sum_{i=1}^{n}-p_i\log_2 p_i$$

如果 $m=1$，也就是 T 的样本都属于一类，那么 $I(T)=0$，达到最小值；如果 $p_1=p_2=\cdots=p_m$，也就是每类样本的个数相同，那么 $I(T)=\log_2 m$，达到最大值。

假设属性 A 把集合 T 划分成 V 个子集 $\{T_1,T_2,\cdots T_v\}$，其中 T_i 所包含的样本数为 n_i，那么划分后的熵就是：

$$E(A)=\sum_{i=1}^{v}\frac{n_i}{n}I(T_i)$$ 那么，分裂后的信息增益为：

$$Gain(A)=I(T)-E(A)$$

描述 3：设属性 A 有 m 个不同的值 $\{a_1,a_2,\cdots a_m\}$，可以用属性 A 将 S 划分为 m 个子集 $\{s_1,s_2,\cdots s_m\}$，其中 s_i 包含 S 中这样一些样本：它们在 A 上具有 a_i。假如我们以属性 A 的值为基准对样本进行分割，Splitl(A) 就是初始信息量：

$$Splitl(A)=-\sum_{i=1}^{m}p_i\log_2(p_i)$$

信息增益比定义为信息增益与初始信息量的比值：

$$GR(A)=Gain(A)/Splil(A)$$

5 归纳决策树

在数据预处理后，进行归纳决策树。此过程使用数据预处理所得的训练集，每次选取选择信息增益比最大的但同时获取的信息增益又不低于所有属性平均值的属性，作为树的结点，将每一个可能的取值作为此节点的一个分支，递归地形成决策树。递归的结束条件是子集中的数据记录在主属性上取值都相同，或没有属性可再供划分使用。

之所以选取获取率大而信息增益不低于平均值的属性，是因为高获取率保证了高分支属性不会被选取，从而决策树的树型不会因某节点分支太多而过于松散。过多的分支会使得决策树过分地依赖某一属性。而信息增益不低于平均值保证了该属性的信息量，使得有利于分类的属性更早地出现。

6 决策树剪枝

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：

（1）事前修剪（pre－pruning）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 或信息增益比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（post－pruning）方法。该方法从一个“充分生长”树中，修剪掉多余的树枝。，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。该原则的基本思想即是：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。

7 一个实例

银行在处理一批客户的信贷活动中，确定是否对具备贷款担保条件的客户发放贷款，存在决策分析问题。在客户信息表中去一些不必要的属性，保留收入、客户年龄、信誉度、贷款期限和准予贷款 5 个属性，做成判断准予客户贷款的训练样本集到表 1。

表 1 训练样本集

年收入	客户年龄	信誉度	贷款期限	准予贷款
≤4 万	50－60	一般	中期	否
≤4 万	50－60	一般	短期	否
≥6 万	50－60	一般	中期	是
4 万－6 万	20－30	一般	中期	是
4 万－6 万	30－50	良好	中期	是
4 万－6 万	30－50	良好	短期	否
≥6 万	30－50	良好	短期	是
≤4 万	20－30	一般	中期	否
≤4 万	30－50	良好	中期	是
4 万－6 万	20－30	良好	中期	是
≥6 万	20－30	良好	短期	是
≥6 万	20－30	一般	短期	是
≥6 万	50－60	良好	中期	是
4 万－6 万	20－30	一般	短期	否

7.1 计算

（1）计算训练样本的信息量

$$I(T) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.9403$$

（2）计算每个属性的信息增益

① 属性收入有 3 个取值，把样本集分为 3 个子集 $\{T_1, T_2, T_3\}$，每个子集的信息量计算过程如下：

$$I(T_1) = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} = 0.8113$$

$$I(T_2) = 0$$

$$I(T_3) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5} = 0.9710$$

属性收入的熵为：$E(\text{收入}) = \frac{4}{14}I(T_1) + \frac{5}{14}I(T_2) + \frac{5}{14}I(T_3) = 0.5786$

② 同理可算出属性为客户年龄、信誉等级、贷款期限的熵

E（客户年龄）=0.9111，E（信誉等级）=0.7885，E（贷款期限）=0.89225

③ 计算各属性的信息增益。

Gain（收入）=I（T）—E（收入）= 0.3617，Gain（客户年龄）=0.0292，Gain（信誉等级）=0.1518，

Gain(贷款期限) =0.0481

(3) 计算各属性的信息增益比

$$SplitI(\text{收入}) = -\frac{4}{14}\log_2\frac{4}{14} - \frac{5}{14}\log_2\frac{5}{14} - \frac{5}{14}\log_2\frac{5}{14} = 1.5774$$

GR(收入) = Gain(收入)/SplitI(收入) =0.3617/1.5774 =0.2293

GR(客户年龄) =0.0188 ,GR(信誉等级) =0.1518 ,GR(贷款期限) =0.0488 ,

得到:GR(收入) > GR(信誉等级) > GR(贷款期限) > GR(客户年龄),收入属性作为决策树的根结点。

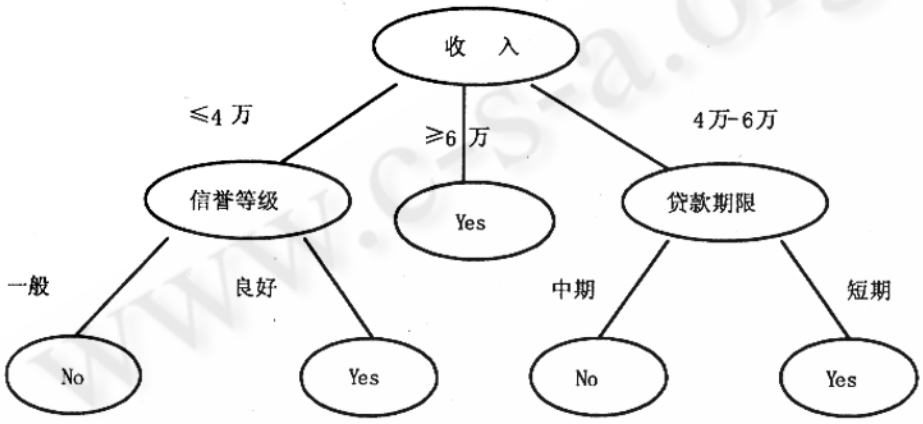

图 1　生成决策树

7.2 预剪枝和构造决策树

由于噪声数据的影响,以及某些规则仅基于少量数据,与客观事实不符,造成通过学习训练集来构造的决策树可能无法达到最好的泛化性能。可以通过对决策树进行剪枝达到预期的目的。前文提到的有预剪枝和后剪枝,预剪枝在实现上比较简单,并且在系统性能上有很大的改进,这里采用预剪枝。但其弱点是进行修剪的阈值要人工定义,这样容易使修剪过粗或过细,进行多次试验确定的阈值最佳点可弥补其弱点。

经过试验,笔者找到一种较好的方法:当构造决策树的算法进行到要对当前节点进行分裂时,此时计算当前节点的父节点增益比率和此节点的增益比率的比值,如果比值小于给定的阈值,则认为此节点趋于纯净,停止节点分裂。经过反复试验,针对本例的属性,这个阈值定在 0. 5 生成的决策树比较合适。选择贷款期限作为决策属性,得到两个叶结点。图 1 给出了最终生成决策树。

在这个决策树模型终于将此有关的只有收入、信誉等级和贷款期限三个属性,剪掉的客户年龄对目标属性的取值没有影响。从模型的结构看,年收入超过 6 万就可直接决定贷款。收入在 4 万 -6 万则要根据贷款期限来判断;而收入在 4 万以下则要根据信誉等级来判断。根据这个模型,可以迅速地对新样本做出判断。例如,如果有一条新记录,"收入≤4 万,年龄 20 -30,信誉等级一般,贷款期限中期",根据模型可知准予贷款的值应该是"否"。

8 结束语

C4.5 算法是在 ID3 的基础上改进而成的,它更好地修正了 ID3 的剪枝算法,并对高分支属性、数值型属性和含空值属性的整理有了系统的描述。决策树是数据挖掘中一个常用的算法工具,它易于转化为图像显示的特点,使得它深受使用者欢迎。除了在客户分析上的应用外,决策树算法在市场划分、金融风险、产品开发等 CRM 应用中也有着广阔的应用前景。

参考文献

1 Jiawei Han, Micheline Kamber. 数据挖掘:概念与技术[M],北京:机械工业出版社, 2001. 188 -196.

2 史忠植,知识发现[M],北京:清华大学出版社, 2002. 2 -28.

3 朱迪茨, 实用数据挖掘[M],北京:电子工业出版社, 2004. 7 -9.

4 汉德,数据挖掘原理[M],北京:机械工业出版社, 2003. 1 -2.

5 王晓国等,应用 C415 算法构造客户分类决策树的方法[J],计算机工程, 2003, (14) .

6 栾丽华、吉根林, 决策树分类技术研究[J], 计算机工程,2004 , (9) .