维普资讯 http://www.cqvip.com

对一种新的决策树建立方法的研究

The Research for a New Method of Building Decision Trees

郭景峰　米浦波　刘国华

（燕山大学计算机科学与工程系　河北秦皇岛066004）

Abstract The methods of building decision trees that have been developed all begin from training data sets. And then they select the best attribute to split the data by their different criterion. At last a decision tree is got which can classify and predict. To be different to other methods, a new method of building decision trees is present in this paper. It begins from training data set in which all data belong to one class. At last it gets a composite decision tree.

Keywords Decision trees, Support confidence, Over-fitting, Pruning

1　决策树

数据挖掘技术中的决策树技术是首先从机器学习领域得来的，它与关联规则技术作为数据挖掘技术的两个大方向，在许多领域都有广泛的应用，如：医学、地质学、天文学、物理学、金融领域等等，并发挥着巨大的作用。决策树技术能建立分类系统及产生预测系统[1]。从训练样本集出发，它的建立是一个递归过程。它重复执行以下过程：根据评价标准选择数据中的某一属性作为分割标准，将当前节点（数据集）分割成子节点（数据子集），如果各个子节点中的数据属于同一类，以类名标注，过程结束，否则过程在包含不同类数据的子节点中进行。最后产生一个可以自动对数据进行分类的树形结构，也就是树形的知识表示形式，并且能直接转化为决策规则。

最初从训练数据集中产生的是一棵单一的决策树，而在1987年 Quinlan 提出了复合树的思想。而后，Breiman 提出的 Bagging 以及 Freund 和 Schapire 提出的 Boosting 两种新的分类学习系统，正是复合树思想的实现，它们都大大提高了系统的预测能力[2]。Bagging 和 Boosting 两者都是首先通过评价形成一个分类器（决策树）集合，然后对各个分类器评价、选择、综合，最终获得的结果是一棵复合树。

无论最终获得的是一棵单一的决策树还是复合树，算法都是从训练数据集开始的。训练数据集中的数据的类别属性值各有不同。文章提出了一种新的决策树建立方法，这种方法并不是直接从训练数据集开始，而是将训练数据集按照数据的类别属性值将它划分为若干个训练数据子集，使各个子集中的数据都属于一个类别，然后在各个数据子集上建立决策树，再进行综合，形成一棵复合树。这样的复合树与上述的复合树是不相同的。象绝大部分的决策树建立算法一样，该算法的最后一步是对复合树实行剪枝，以消除过度一致，提高预测的精确性，让知识在一个更高、更清晰的层次上得以体现。

2　几个定义

为了清楚地对算法进行描述，首先要下几个定义。

定义1　对训练数据集按照类别属性进行划分，获得 N（N=总的类别数）个训练数据子集，称之为同类训练数据集，记做 $STSet_i$，其中 $i=1,2,\cdots,N$。

定义2　设某一同类训练数据集中的数据量为 M，其中数据的任意一个条件属性 e 有 j 个值，各个不同值覆盖的数据量分别是 $|e_1|,\cdots,|e_j|$，那么 e 对这个同类训练数据集的支持度（简称为 e 的支持度）定义为：

$$Support\ Confidence(e)=j*\sum_{i=1}^{j}(\frac{e_i}{M})^2 \quad (1)$$

定义3　从同类训练数据集中生成的决策树，我们称之为同类决策树。

定义4　设有一棵同类决策树 T，它的深度记做 $depth$，叶子总数记为 $leaves$，那么这棵树的评价值定义为：

$$value(T)=c_d*w_d*depth+c_l*w_l*leaves+C \quad (2)$$

其中，c_d 是 $depth$ 的平衡系数，w_d 是 $depth$ 的权值；c_l 是 $leaves$ 的平衡系数，w_l 是 $leaves$ 的权值；C 是总的平衡因子。以上几个参数可以根据数据的类型、数据的分布情况以及实际要求而设定。

定义5　把一个训练数据集划分为 N 个同类训练数据集后，从中将会获得 N 棵同类决策树，对这 N 棵树进行综合得到决策树称为完全合成树。

定义6　从训练数据集中建立模式〈A,V,C,F,J,R〉。A 代表属性；$V\in DOM(A)$；C 代表类别。那么，在训练数据集中，满足属性是 A、属性值是 V 的数据量记为 F，即 $F=|V(A)|$；满足属性是 A、属性值是 V，且类别为 C 的数据量记为 J，即 $J=|V_C(A)|$；R 是 F 和 J 的比率，即

$$R=\frac{|V(A)|}{|V_C(A)|} \quad (3)$$

性质1　根据定义6我们可知当 $R=1$ 或趋向于1时，属性 A 取值为 V 的所有数据的类别属性值是 C，即这些数据同属于一类。

根据性质1，我们可以建立一棵根为 A，叶为 C，树枝为 V 的决策树 OT。当用 OT 对数据集进行分类时，它能把属性 A 取值为 V 的数据从数据集中分离出来。

3　一种新的决策树建立方法

已有的决策树建立方法通常是从训练数据集出发，以各自不同的评价标准选择“最优”属性对当前数据集进行分割，直到当数据集中的数据同属一类时，决策树建立完成；为了消除在建立过程中形成的过度一致问题[1]，还要对树进行剪枝，以提高精度。从经典的决策树建立方法 ID3开始引入了“熵”的理论，从此“熵”成为了研究决策树技术的重要手段。它的引

*）此项目得到河北省自然科学基金资助，项目号（600225）

维普资讯 http://www.cqvip.com

进确实提高了决策树分类的准确度与速度，但是具有更多个属性值的属性通常更容易被当作“最优”属性被挑选出来对数据进行分割。

本文提出的新算法对决策树的建立并不是直接从训练数据集开始，而是从同类训练数据集出发；利用属性对同类训练数据集的支持度建立决策树，考察的是属性对类别的支持程度，支持度越高，说明对分类起的作用越大，并且与属性的值的多少无关；另外，不需要象以往的算法那样，在各个子节点都需要重新执行“选择－分割”的操作，对各个属性进行一次评价就可以。从这个方面说，这种方法在建立决策树的速度上是有一定优势的。

决策树的建立通常包括两个阶段，一是树的建立，另一个是树的剪枝。从训练数据集中建立的决策树通常是过度一致的，也就是说，对训练数据而言，它过于有针对性，太特殊化。这样的决策树在训练数据集中通常有很高的精确性，但在实际应用中精确性却会大大降低。为了消除这种不良现象，大部分的决策树方法都要经历剪枝阶段。由于我们提出的新算法不是直接从训练数据集进行，因此对同类决策树合成后所得的完全合成树存在更为严重的过度一致问题。在解决这个问题上，我们采用合理的剪枝策略对完全合成树实行剪枝，以提高决策树在实际应用中的精确度，同时使它能够清晰地表现知识，提高知识的可读性。

以下就决策树建立的两大阶段进行说明。

3.1 决策树的建立

主算法开始执行之前要做的准备工作有三个：

(1)按定义6对训练数据集进行处理，如果所得的 OT 不止一棵，即 $|OT|>1$，对它们进行综合，结果仍然以 OT 标注；如果 OT 不存在，记 $|OT|=0$。

(2)将训练数据集中满足条件“属性 A 取值为 V，R 值为1或趋向于1”的数据删除，因为覆盖该知识面的决策树已经建立。

(3)对经过预处理所得的训练数据集按数据的类别属性进行划分，形成 N（N 为总的类别数）个同类训练数据集 $STSet_i$，$i=1,2,\cdots,N$，作为算法执行的对象。

主算法描述：

```
BEGIN
(1)From i=1 to N
    按照公式(1)的定义，计算 STSet_i 中的各个属性对于
    STSet_i 的支持度；
    按支持度由高到低的次序对属性进行排列；
    按属性排列的次序对 STSet_i 分割，建立同类决策树
    T_i；
    Endfor
(2)For i=1 to N
    按照公式(2)的定义，计算 T_i 的评价值；
    Endfor
(3)以评价值由低到高的顺序对单一决策树进行排序，评
    价值最低的单一决策树记为 TT_1，评价值最高的是
    TT_N；
(4)if OT=0 then
      For i=2 to N
          T=TT_i 与 TT_{i-1}合并；//T 为中间变量
          TT_i=T；
      Endfor
  Else
      T=OT 与 TT_1合并；
      TT_1=T；
      For i=2 to N
          T=TT_i 与 TT_{i-1}合并；
          TT_i=T；
      Endfor
  Endif
(5)Tree=T，为一棵完全合成树。
END
```

在这一部分中需要说明的问题主要有两个：(1)属性对同类训练数据集的支持度越高，表明它越支持该类与其它类的区分，因此支持度越高越有利于分类；(2)同类决策树的评价值是对一棵同类决策树性能的衡量，其中的权值、平衡系数、平衡因子是起调节作用的参数。由于树的深度越小、叶节点数越少，树的性能越优良[1,3]，因此评价值越小越有优势。

3.2 决策树的剪枝

对决策树做剪枝工作的目标一般有两个，一是提高知识的聚集程度，使之更易于人们的理解；另一个是使决策树具有更高的精确度[4]。通常的做法是计算决策树的子树在测试数据集中的分类准确度，使它被某叶节点或其它子树代替后具有更高的精度，考察的是分类的准确度；I. Shioya 等人对决策树进行剪枝考虑的是知识的语义层次[5]，考察的是知识的表现度和可读性。剪枝的两大目标通常是相辅相成的。决策树的适当规模可以带来最佳的精度(CART)。

由于新算法产生的过度一致问题是无法避免且无法忽略的，我们必须进行剪枝过程。在这个过程中采用的方法是悲观修剪法(Pessimistic Pruning)[4]。这种方法有两大优势：(1)由于决策树的每一个子树最多只做一次评价，因而拥有更高的速度；(2)与其它方法不同的是，它不需要一个与从产生决策树的训练样本集相脱离的测试集，避免了一个形成高要求测试集的过程，即它可以在由训练数据集中某些数据构成的测试集中进行。

首先要说明几个概念。

定义7 设 S 是决策树 T 的子树，它包含了 $L(S)$ 个叶节点，ΣK 是测试集的数据量，ΣJ 是 S 在测试数据集中运行产生的被错误分类的数据总量，那么对 S 的一个更悲观的预计是：在 ΣK 个实际数据中的错误分类总数是：

$$SE=\Sigma J+L(S)/2 \tag{4}$$

定义8 设 N' 是某测试样本集的样本量，该测试集对决策子树 S 进行检测后，可以求得 S 在未来实际数据中可预计的错误分类数 SE，那么决策子树 S 具有的标准分类错误数是：

$$se(SE)=\sqrt{\frac{SE\times(N'-SE)}{N'}} \tag{5}$$

定义9 当 S 被某叶节点 L 代替后，在同一个测试样本集中进行测试，观察到分类错误的样本量是 E，那么对 E 的更实际的评估是 EE：

$$EE=E+1/2 \tag{6}$$

现在我们可以开始讲解 Pessimistic Pruning 的具体操作：对于决策树 T 的某个有 $L(S)$ 个叶节点的子树 S，以及可能用来替代 S 的某个叶节点 L。通过测试和计算可得 S 的 SE 和 $se(SE)$；也可得到 EE。如果 $EE<SE+se(SE)$，那么Pes-

（下转第129页）

维普资讯 http://www.cqvip.com

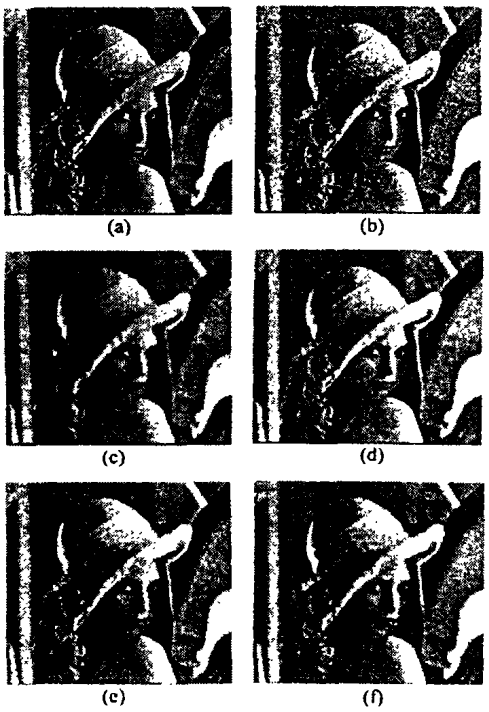

图2 实验结果

结论 对于含有高斯和脉冲混合噪声的图像，PGF 滤波是一种很有效的自适应滤波方法。但是该方法在进行同组成员归类时，引入了 Fisher 线性判别，从而在算法具体实现时需要耗费计算机大量的运算时间，很不适合适时图像处理。本文提出的改进算法是先计算图像的局部能量来判断像素是否为脉冲噪声，若是便用中值滤波滤除，反之将滤波窗口中像素特征变化最小方向上的点归为同组成员，然后可进行 PGF 滤波，可大大提高 PGF 滤波的运算速度。实验结果证明了该算法是一种快速的、能滤除混合噪声又能保护图像边缘信息的滤波方法。

参 考 文 献

1 赵荣椿. 数字图像处理导论. 西北工业大学出版社，1995
2 Astola J，Haavisto P，Neuvo Y. Vector median filters. In：Proc. of IEEE，vol. 78，April，1990. 678～689
3 Tang K，Astola J，Neuvo Y. Adaptive nonlinear multivariate image filtering for mixed noise removal. In：Proc. of ISCAS. vol. 1. 1993. 427～430
4 Deng Y，Kenny C，Moore M S，Manjunath B S. Peer group filtering and perceptual color image quantization. In：Proc. of ISCAS，1999
5 Hewer G，et al. Peer group processing for segmentation：[Technical Report #97. 01]. ECE Dept. UCSB，1997
6 Kenny C，et al. Peer Group Image Enhancement. IEEE Tran，ON Image Processing，2001，10(2)：326～334
7 Cherikh F A，et al. Impulse noise removal in highly corrupted color images. In：Proc. of ICIP ，vol. 1. 1996. 997～1000
8 Carmona R A，Zhong Sifen. Adaptive Smoothing Respecting Feature Direction. IEEE Tran，ON Image Processing，1998，7(3)：353～358
9 Di Zenzo S. A note on the gradient of a multi-image. Comput. Vision Graphics Image Proc，1986，33：116～125

（上接第122页）

simistic Pruning 将会用叶节点 L 代替子树 S；否则将寻找另一个叶节点对它进行替代，结果也可能是不存在这样的叶节点。整个剪枝过程如此反复进行，直至 T 中的所有子树都找不到一个比它更“优秀”的叶节点代替为止。

现在举一个简单的例子说明该方法。如，某一决策子树 S 覆盖了训练数据集中的2000个样本，该子树有4个叶节点，为了更容易地说明问题，就用这同样的2000个样本作为测试样本集对它进行剪枝。如此，$\Sigma J=0$，$L(S)=4$，$N'=2000$。且设该子树被某个叶节点 L 代替后产生2个错误（由于训练数据集也是测试集，可以容易地观察到），即 $E=2$。那么有 $SE=\Sigma J+L(S)/2=2$，$EE=E+1/2=2.5$，$se(SE)=1.41$。$EE<SE+se(SE)$，所以可以用叶节点 L 代替子树 S。

对决策树进行剪枝的过程无疑是复杂而耗时的，但它却能够提高最终的准确度，以及降低树的复杂性，缩小它的规模，使知识得以清晰、准确地表现。同时，Pessimistic Pruning 与其它方法比较有较高的速度。因此我们的剪枝策略采用 Pessimistic Pruning，以提高完全合成树的精度和表现度。

4 后继工作

数据的类型和数据的分布无疑会影响决策树建立方法的效率，如何建立它们之间的联系，我们的新算法在什么样的数据类型及数据分布情况下能获得更优的性能；算法中的调节因子应该如何根据实际情况设定，以及如何在分类系统中实时地实现自动设置；如何提高剪枝的效率。以上是我们下一步在理论研究工作中的主要内容。在实验方面重点是算法的实现和对比。由于 C4. 5已经成为商业系统，有较多的获取途径，且具有良好的性能，各种新方法在形成之初大都做试验与之比较，评价新算法的优劣，因此我们的比较对象为 C4. 5。试验与理论是相辅相成，相互起着促进作用的。

结论 以往的决策树建立方法都是从训练数据集开始执行算法，在当前数据集中挑选最优属性对数据进行分割，最终获得决策树。文章提出了一种新的决策树建立方法，该方法以同类训练数据集作为算法进行操作的对象。以支持度作为衡量属性的标准，能寻求到确实最支持某一类别的属性，同时在各个同类训练数据集中，每一个属性的支持度仅仅需要计算一次，能够提高算法的效率。所有的同类决策树生成后，对它们的综合并不是无序的，以性能最好的同类决策树为中心，其它同类决策树以由优到劣的次序向它靠拢。最后，对已经形成的完全合成树做剪枝操作，提高决策树的准确度。

从理论上说，这种方法高效、精确度高，有广泛的应用价值。

参 考 文 献

1 Aptè C，Weiss S. Data Mining with Decision Trees and Decision Rule. Future Generation Computer Systems，1997，13：197～210
2 Quinlan J R. Bagging，Boosting，and C4. 5. In：Proc. Thirteenth National Conf. on Artificial Intelligence，AAAI Press，1996. 725～730
3 Agrawal R ，et al. Database Mining：A Performance Perspective. IEEE Trans. On Knowledge and Data Engineering，1993，5(6)：914～925
4 Quinlan J R. Simplifying Decision Trees. Int. J. Man-Machine Studies，1987，27：221～234
5 Shioya I. Miura T. Knowledge pruning in decision trees. In：Proc. of 12th IEEE Intl. Conf. on Tools with Artificial Intelligence，2000(ICTAI). 40～43