Proceedings of the 6th World Congress on Intelligent Control
and Automation, June 21 - 23, 2006, Dalian, China

The Relationship between Decision Trees and the Scale of Train Data Set

Ning Chen[1], Meilin Zhu[2], Yong Jiang[3], An Chen[4]

[1)] Institute of Mechanics, Chinese Academy of Sciences, Beijing 100080
(E-mail: ningchen74@yahoo.com)
[2)] School of Management and Engineering, Nanjing University, Nanjing 210093
(E-mail: zhuml@nju.edu.cn)
[3)] China Aerospace Science & Industry Corporation, Beijing 100074
(E-mail: change68@sina.com)
[4)] Institute of Policy and Management, Chinese Academy of Sciences, Beijing 100080
(E-mail: anchen@gscas.ac.cn)

***Abstract*—Decision Trees need train samples in the train data set to get classification rules. If the number of train data was too small, the important information might be missed and thus the model could not explain the classification rules of data. While it is not affirmative that large scale of train data set can get well model. This Paper analysis the relationship between decision trees and the train data scale. We use nine decision tree algorithms to experiment the accuracy, complexity and robustness of decision tree algorithms. Some results are demonstrated.**

***Keywords*—decision tree, train data, accuracy, complexity**

决策树算法与训练样本规模的关系研究

陈宁[1]　朱美琳[2]　姜勇[3]　陈安[4]

[1)]中国科学院力学研究所 北京 100080
[2)] 南京大学工程管理学院 南京 210093
[3)] 航天科工集团 31 所 北京 100074
[4)] 中国科学院科技政策与管理科学研究所 北京 100080

摘　要 决策树是由训练数据集的样本得到的，如果样本太少，就可能遗漏了重要的信息，得到的模型也就不能正确反映数据的分类规律。但是，训练样本过多，也不一定能得到好的模型。本文通过实验对各种决策树的样本规模进行了分析，分别采用九种决策树算法对决策树的准确度、复杂度及鲁棒性进行了测试，并得到了一些结果。

关键词 决策树，训练样本，准确度，复杂度

1．引言

在诸多的分类方法中，决策树是一种常用的，快速直观的分类方法，在很多领域中都有广泛的应用。

顾名思义，“决策树”有一个树状的结构，包括结点和用于连接结点的线，而结点又分为根结点、内部结点和叶结点三类。每个结点对应一个样本集，而根结点对应整个样本集，内部结点对应一个样本子集，叶结点对应一个类标志。根结点和内部结点都包含一个对样本属性的测试，根据测试的结果将样本集划分为两个或多个子集，每个子集生成一个分支，分支用测试的属性值来标识。叶结点包含一个类标志，表示对应样本集的类别。从叶结点的角度来看，决策树把整个数据空间划分为若干子空间，属于一

国家自然科学基金项目（ 资助号：70201003 ）

个子空间的所有样本都被标识为相应叶结点的类别。

决策树的构造通常包括两个步骤：首先利用训练集生成决策树，然后再对决策树进行剪枝。决策树的生成是一个从根结点开始、从上到下的递归过程，一般通过不断地将训练样本分割成子集的方式来构造决策树。可能还会需要对决策树进行修剪，删除多余分支。使用决策树对新样本进行分类时，从根结点开始对该样本的属性进行测试，根据测试结果确定下一个结点，直到到达叶结点为止，叶结点标识的类别就是新样本的预测类别。

决策树是由训练数据集的样本得到的，如果样本太少，就可能遗漏了重要的信息，得到的模型也就不能正确反映数据的分类规律。但是，训练样本过多，也不一定能得到好模型，本文通过实验对各种决策树的样本规模进行了分析。

2．常用的决策树算法概述

决策树是应用最广泛的分类算法之一，它具有以下几个方面的优点：（1）模型构造不需要其它的领域知识，避免了专家系统中领域知识的瓶颈问题，学习过程需要的参数也较少，对训练集没有任何要求，能处理离散型和连续型的数据，分类的准确度较高。（2）决策树的构造过程是可监测的，相对于需要多次迭代才能达到稳定的神经网络分类算法，决策树算法的学习时间较短。（3）决策树的每个叶结点对应一条分类规则，构造过程也就是分类规则的发现过程。这种层次结构的树模型表达形式简单，容易理解，对新样本的分类过程只需要一系列的测试就可完成，执行效率高。

决策树算法有很多种，但是到目前为止，还没有一种算法对任何数据集的分类质量优于其他所有算法。在具体的应用中，要根据数据的特征和问题的性质来选择合适的算法。以下是一些常用的决策树算法或工具。

（1）ID3 [1, 2] ：采用信息理论和贪心搜索算法选择分裂特征。

（2）C4.5 [3, 4]：对 ID3 的改进，采用基于信息增益比的特征选择策略和基于最小描述长度的剪枝方法，可以处理连续型属性和缺失数据，具备分类规则的推导功能。

（3）CHAID [5]：采用基于$\chi 2$ 统计测试的特征选择方法构造决策树，连续型属性必须先作离散化处理，利用连续表记录所有可能的分裂方式。

（4）CART [6]：采用交叉测试的剪枝方法，能对数据进行预处理，支持 Bagging 和 Boosting 技术。

（5）Elisee [7]：采用$\chi 2$ 统计测试的二分法，使得分裂后各子集之间的差距最大，通过计算样本的实际分布和模型之间的偏差来确定最优的分裂方式。

（6）SIPINA [8, 9]：首先对连续型属性离散化，根据样本的数学特征通过启发式方法搜索归纳图，发现最优的特征和分裂方式，直至没有满意的分裂为止。

（7）QR-MDL [10]：利用 MDL 准则构造决策树，并对树进行剪枝，兼顾到模型的准确性和复杂性。

（8）WDTaiqm [11, 12]：以贝叶斯测试和评价为理论建立的决策树模型，同时考虑到模型的信息完备性和复杂性。

（9）PolyAnalyst：基于信息理论和统计测试方法进行决策树生成。

近年来，研究者将现代技术和决策树算法相结合，研究出很多决策树工具，例如可提供可视化的图形界面，支持数据抽取、管理和预处理的功能，又如可以把决策树转换成 XML 格式存储，还有交互式的决策树生成工具，以及可嵌入在 Excel 中的决策树等等。

3．训练样本及实验方法

在对决策树的研究中，我们发现训练样本的规模与决策树模型的优劣有一定的关系。本文以德国信用卡数据库（German Credit Database）作为实验数据集进行分析，该数据集记录了顾客的个人信息和信用情况，共有 20 个属性，其中 7 个是连续型，13 个是离散型，数据库中共有 1000 个样本，根据信用情况分为“GOOD”或“BAD”两类，分别有 700 个样本和 300 个样本。

我们分别采用 9 种决策树分类算法：ID3、C4.5、CART、CHAID、Elisee、WDTaiqm、PolyAnalyst、Sipina 和 QR-MDL，表 1 列出了这些算法所采用的属性选择及剪枝策略。

表 1　九类决策树算法的特征

算法	属性选择策略	剪枝策略
C4.5	Gain ratio	Statistical test
CART	Gini index	Cross validation
ChAID	Chi-square	Direct stopping rule
Elisee	Chi-square	Direct stopping rule
ID3	Gain ratio	－
Sipina	Fusinter	Cross validation
WDTaiqm	Bayesian approach	Information quality measure
QR-MDL	MDLP	MDLP
Polyanalyst	Shannan Information	Statistical test

具体地说，实验过程包括三个步骤：

（1）数据准备：对数据集进行随机抽样，生成一个测试集和一组大小不等的训练集，使得训练集和测试集没有重复样本，而且不同类别的样本数大致相等。在我们的实

验中，测试集采用了 300 个样本，而训练集的大小从 50 逐渐增加到 300，每次增加 50 个样本。

（2）决策树构造：对每个训练集，分别采用不同的决策树算法构造决策树，然后利用测试集评价准确度。

（3）性能分析：评价分类算法的指标通常有模型构造的速度、模型的准确度及模型的复杂度等。当训练数据比较小时，决策树算法的速度不是主要问题，而准确度就相对地显得更加重要。分类模型的复杂度，也就是表达方式的简单程度，决定了模型是否容易理解并迅速作出决策，本文采用树的层次和结点数来衡量决策树的复杂度。

4. 实验结果分析

以下是 9 种决策树分类算法对实验数据集的评价结果，我们分别从训练集的规模与准确度的关系，训练集的规模与结点数的关系、决策树的结点数与准确度的关系以及鲁棒性等方面对各种算法进行了实验与比较。

4.1 训练样本的规模与准确度的关系

针对不同规模的训练集，我们分别采用九种决策树算法构造决策树模型，并测试得到准确度，最终计算出平均准确度。从表 2 中的实验数据，可以发现训练样本数对决策树准确度的影响。随着训练样本的增多，准确度明显地有所提高。但当训练样本的个数超过某个阈值之后，模型的准确度不再有明显的改善。例如，当训练集为 50 个样本，有 7 种算法的准确度在 50%～55%之间，1 种算法的准确度在 56%～60%之间，还有 1 种算法的准确度大于 66%，平均准确度为 54%。当样本个数增加到 150 时，平均准确度提高到 63%。但当训练样本个数多于 150 之后，准确度并没有明显的提高。

表 2　决策树的平均准确度

训练样本个数	训练准确度（%）				
	50~55	56~60	61~65	66~	平均
50	7	1	0	1	54
100	3	6	0	0	55
150	0	1	4	4	63
200	0	3	2	4	63
250	1	3	1	4	64
300	3	0	2	4	63

以 150 个训练样本为例，表 3 是九种决策树算法在该训练数据上得到的准确度、结点数、叶结点数以及决策树深度的指标，表 4 则是用所得的决策树模型在测试样本集上的准确度。

表 3　九种算法在训练数据上的结果

算法	训练准确度(%)			结点数	叶结点数	深度
	类 1	类 2	全体			
C4.5	92	87	89	10	53	32
CART	41	64	53	2	3	2
CHAID	55	80	67	2	5	4
Elisee	67	77	72	3	5	3
ID3	91	91	91	9	69	40
Sipina	56	64	60	2	3	2
WDTaiqm	93	93	93	7	22	10
QR-MDL	93	93	93	8	64	34
Polyanalyst	55	80	67	2	5	4

表 4　九种算法在测试数据上的准确度

算法	测试准确度(%)		
	类 1	类 2	全体
C4.5	65	56	61
CART	60	79	69
CHAID	60	79	69
Elisee	67	73	70
ID3	61	62	61
Sipina	52	63	57
WDTaiqm	59	65	62
QR-MDL	61	58	60
Polyanalyst	59	80	70

图 1 显示了对不同的数据集和算法，训练样本数与准确度之间的关系。在实验中，各种算法都表现出了不错的分类能力，在不同的数据集中，各算法的准确度也会随之变化，整体情况来看，CART、CHAID、Elisee 和 Polyanalyst 的准确度相对较高，QR-MDL 算法的表现较弱。

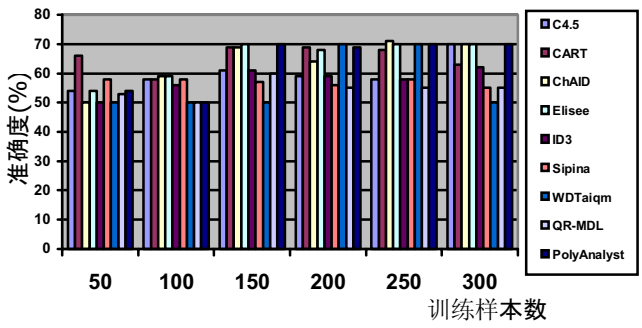

图 1　训练样本数与准确度的关系

4.2 训练样本的规模与结点个数的关系

图 2 给出了三个数据集的训练样本数与结点个数之间的关系。随着训练样本的增多，构造的决策树有越来越复杂的趋势，特别是没有采取剪枝策略的算法（如 ID3 算法），结点数的增长更加明显。

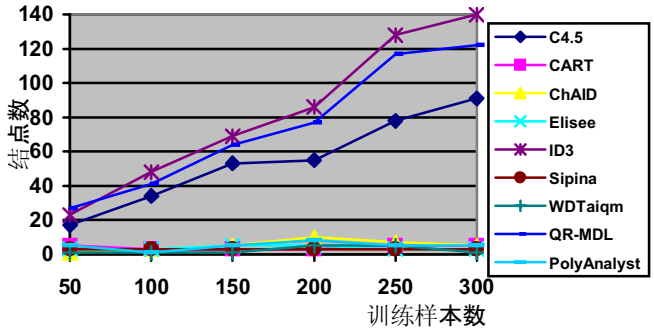

图 2　训练样本数与结点数的关系

4.3 决策树的结点个数与准确度的关系

如果训练集里不存在异常样本，即两个样本的条件属性完全相同，却属于不同的类，只要决策树足够复杂，总能满足所有的训练样本，从而达到完全拟合的效果。但这样的模型往往太依赖特殊的样本，这样就很可能得出错误的规则，导致决策时准确度不高。以 CART 算法和 CHAID 算法为例，图 3 显示了 CART 算法决策树的结构，有 5 个结点，其中 4 个为叶结点，可得到 4 条分类规则：

R1: STATUS<0 → BAD (74.51%);

R2: STATUS<200 → BAD (62.9%);

R3: STATUS= “No account” → GOOD (75%);

R4: STATUS>=200 → GOOD (73.24%);

图 4 显示了 CHAID 决策树的结构，包含 10 个结点，其中 7 个叶结点，可得到 7 条分类规则：

R1: STATUS<0 → BAD (75%);

R2: STATUS<200 → BAD (63%);

R3: STATUS>=200 → GOOD (75%);

R4: STATUS=“No account” and PLAN= “bank” →BAD (100%);

R5: STATUS=“No account” and PLAN= “stores” → BAD (50%);

R6: STATUS=“No account” and PLAN=“none” and AGE<=27.5 → BAD (54.55%);

R7: STATUS=“No account” and PLAN=“none” and AGE>27.5 → GOOD (92%);

我们分别利用这两个决策树模型对测试数据进行分类，前者的准确度达到 69%，而后者的准确度只有 64%。显然，后者虽然在结构上比前者复杂得多，但分类准确度却较低，可见对 “STATUS”的细分并没有提高决策树的能力，只使得模型复杂度有所增加。

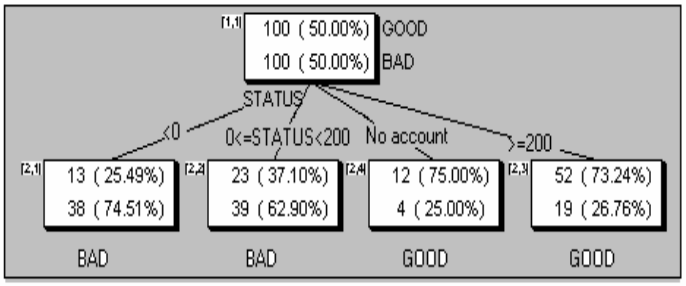

图 3　CART 算法决策树

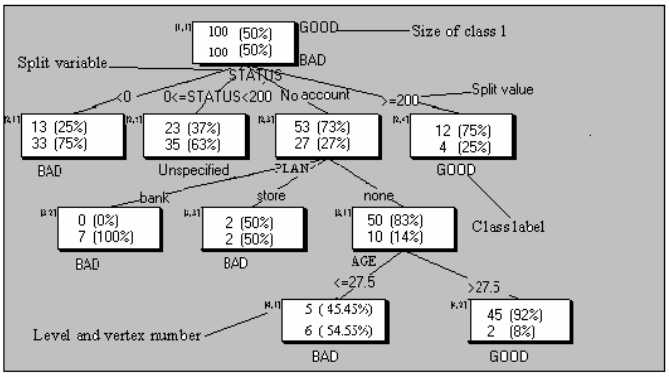

图 4　CHAID 算法决策树

在图 5 中，以横坐标代表决策树结点的个数，纵坐标代表准确度，可以看到结点个数与准确度之间的关系，当结点的个数太多或者太少时，决策树的分类准确度都比较低，这是因为太简单的决策树包含的信息太少，不足以得到正确的类别，而太复杂的决策树又容易受噪声的干扰，也会降低分类的准确度。

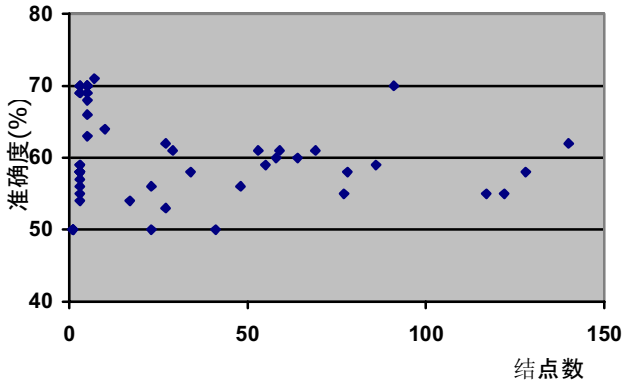

图 5　结点数与准确度的关系

4.4 鲁棒性分析

我们分析了决策树算法的鲁棒性，对于给定的训练数据

集（150 个样本），将测试集的规模从 150 个样本逐渐递增到 850 个，每次增加 50 个样本，从表 5 及表 6 的结果可以看到，各算法的准确度并没有随着样本规模的变化而产生太大的变化，这说明决策树算法具有较强的鲁棒性。

表 5　九种算法的鲁棒性分析

测试样本数	C4.5	CART	CHAID	Elisee	ID3
150	66	64	64	68	68
200	66	66	66	69	66
250	70	66	66	71	71
300	70	55	55	64	70
350	69	65	55	64	70
400	67	65	65	67	70
450	65	61	61	66	69
500	67	61	61	64	68
550	67	62	62	65	67
600	67	65	65	70	71
650	65	62	62	67	69
700	65	62	62	67	69
750	66	63	63	67	69
800	66	64	64	68	70
850	66	63	63	68	70

表 6　九种算法的鲁棒性分析（续）

测试样本数	Sipina	WDTaiqm	QR-MDL	PolyAnalyst
150	68	63	65	62
200	63	60	61	60
250	70	66	68	64
300	67	63	65	64
350	67	64	66	65
400	69	66	65	62
450	69	66	65	62
500	66	63	65	62
550	68	63	66	65
600	70	67	66	65
650	66	64	67	64
700	67	65	65	63
750	67	66	66	63
800	68	66	66	65
850	68	66	67	64

5．结论

决策树算法以其快速直接的性能被广泛应用在分类领域，但数据规模的大小会对算法的准确率有一定的影响。本文在一个给定数据集上研究了九种决策树算法的准确度及复杂度。实验结果显示，当数据规模增大时，决策树的结点数随之增加，呈现出越来越复杂的形态，并对决策树的准确度带来干扰。

参考文献

[1] J. Ross Quinlan, “Learning Efficient Classification Procedures and Their Applications to Chess and Games”, in R. S. Michealski, J. G. Carbonell, eds. Machine Learning: An Artificial Intelligence approach, vol. 1 Pha Alto, CA: Tiogo, 1983.

[2] J. Ross Quinlan, “Simplifying Decision Tree”, Int. J. Man-Mach Studies, vol. 27, no. 3, pp. 221-234, 1987.

[3] J. R. Quinlan, “C4.5: Programs for Machine Learning”, Morgan Kaufman, San Mateo, 1993.

[4] J. Ross Quinlan, “Improved Use of Continuous Attribute in C4.5”, Journal of Artificial Intelligence Research, vol. 2996, no. 4, pp. 77-90.

[5] G. V. Kass, “An Exploratory Technique for Investigating Large Quantities of Categorical Data”, Applied Statistics, 1980, vol. 29, pp. 119-127.

[6] L. Breiman, J. H. Friedman etc., “Classification and Regression Trees”, CA: Wadsworth, Belmont, 1984.

[7] J. P. Bouroche, N. Tenenhaus, “Quelques Methods de Segmentation”, Rairo, vol. 42, pp. 29-42, 1970.

[8] D. A. Zighed, J. P. Auray, G. Duru, “SIPINA Method et Logiciel”, Alexandra Lacassagne-Loyn, 1992.

[9] D. A. Zighed etc., “A Discritization Method of Continuous Attributes in Induction Graphs”, in Proc. 13th European Meeting on Cybernetics and Systems Research, Vienna, pp. 997-1002.

[10] J. Ross Quinlan, “Inferring Decision Trees using the Minimum Description Length Principle”, Information and Computations, vol. 80, pp. 227-248, 1989.

[11] L. Wehenkel, “A Probabilistic Framework for the Induction of Decision Trees”, University of Liege Report, 1992.

[12] L. Wehenkel, “Decision Tree Pruning using an Additive Information Quality Measure”, In B. Bouchon-Meunier, L. Valverde and R.R. Yager eds., Uncertainty in Intelligent Systems, Elsevier - North Holland Press, pp. 397-411, 1993.