doi: 10.7690/bgzdh.2015.10.023

决策树算法在装备故障检测中的应用

陈秀芳，许利亚，刘晓春，王喜权
（中国卫星海上测控部技术部，江苏 江阴 214431）

摘要：为快速定位装备的故障点，引入决策树方法。将故障特征作为测试属性，故障点作为类标记。采用 ID3 算法计算相关信息熵对故障记录进行划分，构建所需要的决策树，用预剪枝的方法控制树的生长。以某设备故障记录样本为例，建立决策树发现故障特征和故障点之间的关联。结果表明：利用该方法对故障点进行预测，可有效缩短装备故障检测的平均时间。

关键词：故障检测；决策树；测试属性；ID3 算法
中图分类号：TJ03　　**文献标志码：**A

Application of Decision-tree Algorithm in Equipment Fault Detection

Chen Xiufang, Xu Liya, Liu Xiaochun, Wang Xiquan
(*Technology Department*, *Satellite Maritime Tracking & Controlling Department of China*, *Jiangyin* 214431, *China*)

Abstract: In order to quick location problem of the equipment, method of decision tree is introduced. The fault characteristics of failure records are treated as the tested attribute and fault point is treated as the class. It uses the ID3 decision tree algorithm of data mining theory to mining the data. Then the decision tree model, which can predict some fault point by the information of failure records, is created. The decision tree is simplified by the pre-pruning at the process of tree growth. Take some failure records of the equipment as example, the association between fault characteristics and fault point is found with decision tree algorithm. The results show that the method can predicate fault point and effectively shorten the average time for equipment fault detection.

Keywords: fault diagnosis; decision tree; test attributes; ID3 algorithm

0　引言

设备故障的诊断一般要掌握丰富的设备原理知识才能进行，当装备发生故障时，往往其外部可观察特征也会发生变化。设备的故障诊断主要是根据其在运行过程中出现的各种状态信息对故障进行分析和识别的。故障状态信息与故障点之间存在着非常复杂的非线性映射关系。挖掘故障可观察特征和故障点间的关系，帮助进行故障分析和定位具有实际意义。数据挖掘技术是利用大量的原始数据，通过分析挖掘出有价值的信息，已经在许多领域得到成功运用。决策树学习作为数据挖掘领域中的一种重要知识，是应用最广的推理算法之一。它分类精度高，操作简单。该算法的最大优点是结构比较简单，不需要了解很多背景知识，具有分类和预测的特性且简单快速[1]；并且能方便、清晰地用图形化的方式表现挖掘结果[2]，相较于关联规则挖掘，其计算复杂度相对较小[3]。

笔者利用决策树方法，将历史故障诊断记录作为训练样本，采用 ID3 算法以故障可观察特征作为测试分裂属性，故障点为类标号，对记录进行相应划分，采用预剪枝的方式控制树的生长来形成决策树。当故障发生时，根据决策树模型预判出相应最可能的故障发生部位，以提高故障检测效率。

1　故障特征决策树

1.1　故障特征决策树基本思想

决策树是以样本属性作为叶结点，用属性的取值作为分支的树型结构。决策树建树的基本原理是递归的将训练数据集拆分成子集，以便每一个包含目标变量类似的状态，这些目标是可预测属性[4]。在拆分中利用信息理论的原则，进行拆分属性选择。设某装备的故障特征集是 $d=\{d_1, d_2, \cdots, d_n\}$，故障点集 $e=\{e_1, e_2, \cdots, e_m\}$，$d$ 是测试属性集，e 是类标记集。形成的故障特征决策树根节点是训练集，一个内部节点代表一个故障特征的测试，一条边代表一个测试结果，叶子代表某个故障点或某种故障处理方式。对于任一内部节点的属性值是离散的，对于每个故障该故障特征要么存在 1(表示故障特征存在)，要么不存在 0(表示故障特征不存在)。

1.2　决策树算法选择

目前常用的算法有 ID3、C4.5 和 CART 是分类算法中较为成熟的内容。ID3 算法使用信息增益作

收稿日期：2015-05-22；修回日期：2015-06-28
作者简介：陈秀芳(1977—)，女，江苏人，工程硕士，工程师，从事信息研究与应用研究。

为属性选择度量。该度量基于信息论的熵理论，选择具有最高信息增益的属性作为节点的分裂属性。用该属性划分训练集后使结果中元组分类所需的信息量最小，该算法的缺陷是偏向于具有大量值的属性[5-7]。C4.5 算法是 ID3 的后继，该算法使用信息增益扩充来克服信息增益度度量偏向选择具有大量值的属性，可有效防止对分类无意义的划分。它使用分裂信息值 SplitInfo(D)将信息增益规范化。CART 算法使用 Gini 指标对每个属性进行二元划分，其强制结果树是二叉的。而在故障特征决策树中属性只有 2 个取值{0, 1}，该特性很好地规避了 ID3 增益度度量偏向算法缺陷，所以笔者采用 ID3 算法属性信息增益进行决策树归纳。

1.3 决策树构造过程

1) 分裂属性选择。

ID3 使用信息增益作为属性选择度量，设故障记录集 D 为类标记元组的训练集，假定类标记属性(故障点)具有 n 个不同值，定义 n 个不同的类 c_i(i=1, …, n)。设 $C_{i,D}$ 是 D 中 C_i 类的元组的集合，IDI 和 IC$_{i,D}$I 分别是 D 和 $C_{i,D}$ 中元组的个数。对 D 中元组分类所需的期望信息由下式计算：

$$\mathrm{Info}(D)=-\sum_{i=1}^{n}P_i\log_2 p_i。\tag{1}$$

其中 P_i 是 C_i 类在 D 中的概率，由下式计算：

$$P_i=|C_{i,D}|/|D|。\tag{2}$$

假设 D 中的元组按照故障特征 A 划分，从以上分析可知 A 属性具有 2 个不同值{0, 1}。那么可以用属性 A 将 D 划分为 2 个子集{D_1, D_2}，用该属性对元组进行划分后，要进行进一步准确的分类，需要的信息量 Info$_A$(D)见下式：

$$\mathrm{Info}_A(D)=-\sum_{j=1}^{G}\frac{|D_j|}{D}\mathrm{Info}(D_j)；\tag{3}$$

$$\mathrm{Info}(D)=-\sum_{i=1}^{G}p_{ij}\log_2 p_{ij}。\tag{4}$$

其中 p_{ij} 为 C_i 类在 D_j 中的概率，由下式计算：

$$p_{ij}=\left|C_{i,j}\right|\Big/\left|D_j\right|。\tag{5}$$

其中：$|D_j|$ 为当属性 A=a_i(i=1,2)的元组数；$|C_{i,j}|$是 C_i 类在 D_j 集合中的元组的个数。

信息增益 Gain(A)是属性 A 的信息增益：

$$\mathrm{Gain}(A)=\mathrm{Info}(D)-\mathrm{Info}_A(D)。\tag{6}$$

由上式可知：属性 A 的信息熵 Info$_A$(D)的值越小，信息增益 Gain(A)的值越大，属性 A 也就对分类提供的信息量越大。根据 ID3 算法思想，为了确保能找到一颗简单的树，在决策树分类属性选择时选择属性增益大的，在选定分裂属性后对训练集进行划分。

2) 故障树的剪枝。

决策树的剪枝原因是避免决策树过拟合样本。如果仅仅按照前面的算法递归生成的决策树非常详细并且庞大，每个属性都被详细地加以考虑，决策树的树叶节点所覆盖的训练样本都是纯的，如果用这个决策树来对训练样本进行分类，将会出现对于训练样本而言，这个树表现良好，误差率极低且能够正确地对训练样本集中的样本进行分类[8]。但训练样本中的错误数据也会被决策树学习，成为决策树的部分，而对于测试数据的表现就没有想象中的好，也可能极差。Quinlan 教授的实验表明，在数据集中过拟合的决策树错误率比经过简化的决策树的错误率要高[9]。在 J.Mingers 关于 ID3 算法的研究中，通过对 5 种包含噪音的学习样例的实现发现，多数情况下过度拟合导致决策树的精度降低了 10%～25%[10]。剪枝策略可有效防止创建一颗庞大的决策树。剪枝分为先剪枝和后剪枝，先剪枝有一个视野效果的缺点，在相同的标准下，也许当前的扩展不能满足要求，但是更进一步的扩展能够满足要求，这将使得算法过早地停止决策树的构造。但是由于剪枝不必生成整棵决策树，且算法相对简单，效率很高，适合解决大规模问题。为了简化计算，笔者采用预剪枝，其具体策略如下：

1) 到达此节点的实例个数小于某个阈值时可停止树的生长。

2) 引入替代错误率。在计算过程中，对子树的某个分支上继续划子集时，虽然所有样本并不属于同一类，但是不同类别的记录数如果相差很大时，就引入错误替代率公式：

$$\mathrm{Pure}=\frac{n-n'}{m}。\tag{7}$$

式中：n 表示分支的记录数；n'表示该分支中多数类别的记录数；m 表示训练集的记录总数。利用该公式计算的值，如果小于某个阈值时，则将子树转化为叶结点，否则就继续调用第 1 步进行进一步分解。

算法递归以上操作，直到无故障特征属性可用于划分当前样本子集或满足树停止生长的条件。

2 应用实例

本实例所使用的数据来源于某设备故障记录，目的是利用这些数据建立决策树发现故障特征和故障点之间的关联。

第 1 步：抽取 320 条数据，取其数据中的 70%为训练元组，30%数据为测试数据。统计测试属性故障特征集及故障点集如表 1。

表 1　某设备故障特征集标志和故障点集标志

故障特征标志代码	故障特征
A1	驱动马达控制器加“使能”异常
A2	驱动柜面板电压、电流故障不稳异常
A3	制动器打开指示是否灯异常
A4	电机制动器打开异常
A5	信号调理抽屉速度指令输出异常
A6	速度环路板速度指令输出异常
A7	驱动马达控制器加“高压”异常
A8	驱动柜 PLC“运行”输入控保异常
故障点标志代码	**故障点**
B1	驱动柜 PLC 故障
B2	速度环路板故障
B3	ACU 信号调理板故障
B4	减速箱故障
B5	旋转关节故障
B6	行星减速器故障

第 2 步：按以上分析对数据预处理，如表 2。

表 2　规范处理后的故障特征和故障点记录

类标号	故障特征集						
	A1	A2	A3	A4	A5	A6	A7
B1	0	0	0	1	0	0	1
B2	1	0	0	0	1	1	0
B3	0	0	0	1	0	0	1
B4	1	1	1	1	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
B6			1			1	0

在这 224 条记录的中故障点的计数统计如表 3。

表 3　故障点计数统计

类标号	故障点计数	类标号	故障点计数
B1	50	B4	34
B2	92	B5	24
B3	16	B6	8

第 3 步：通过式 (6) 逐个计算故障特征信息增益度，选取最能划分训练集中实例的属性。

首先计算信息熵

$$\mathrm{Info}(D)=-\frac{50}{224}\log_2\frac{50}{224}-\frac{92}{224}\log_2\frac{92}{224}-\frac{16}{224}\log_2\frac{16}{224}-\frac{34}{224}\log_2\frac{34}{224}-\frac{24}{224}\log_2\frac{24}{224}-\frac{8}{224}\log_2\frac{8}{224}=2.001。$$

从属性 A_1 开始，对 A_1 的每个值观察 B_1 到 B_6，分布如表 4。

表 4　A_1 值与 B_1 到 B_6 分布情况

故障特征和故障点搭配	记录数
A_1=1 且 B_1=1	48
A_1=1 且 B_2=1	10
A_1=0 且 B_1=1	2
A_1=0 且 B_2=1	82
A_1=0 且 B_3=1	34
A_1=0 且 B_4=1	16
A_1=0 且 B_5=1	24
A_1=0 且 B_6=1	8

计算按 A_1 划分所需要的信息：

$$\mathrm{Info}(D_{A_1})=\frac{58}{224}(-\frac{48}{58}\log_2\frac{48}{58}-\frac{10}{58}\log_2\frac{10}{58})+\frac{166}{224}(-\frac{82}{166}\log_2\frac{82}{166}-\frac{34}{166}\log_2\frac{34}{166}-\frac{16}{166}\log_2\frac{16}{166}-\frac{8}{166}\log_2\frac{8}{166}-\frac{2}{166}\log_2\frac{2}{166}-\frac{24}{224}\log_2\frac{24}{224})=1.263。$$

A_1 的信息增益度为

$$\mathrm{Gain}(A_1)=\mathrm{Info}(D)-\mathrm{Info}_{A_1}(D)=2.001-1.263=0.738。$$

同理可得其他故障特征的信息增益度，结果如表 5 所示。

表 5　其他故障特征的信息增益度

故障特征	信息增益度 Gain	故障特征	信息增益度 Gain
A_2	0.169 3	A_5	0.569
A_3	0.260 1	A_6	0.093
A_4	0.685 0	A_7	0.638

由此可以看出：所有故障特征中 A_1 最能划分训练集中数据，所以首先选择 A_1 属性对记录进行划分。在本实例中，为了确保分类精确度错误率 p=0.017 8，实例个数小于 8 时树停止生长，观察 A_1 属性进行划分后在 A_1=1 分支中 58 条记录 48 条是 B_1，利用错误替代率公式计算：

$$\mathrm{Pure}=\frac{58-48}{224}=0.044\,6。$$

所以 A_1=1 分支要接着划分。算法构建出的最终决策树如图 1 所示。

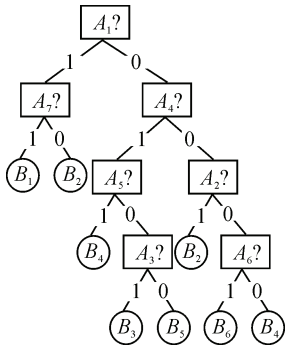

图 1　构建的决策树

决策树生成以后，对从根到树叶的每条路径形成一个故障点判断规则，用测试数据对判断规则准确性进行检测发现，该决策树在测试数据集中表现较好，准确率达到 95.8%。

3　结束语

在决策树算法中把故障特征作为测试属性，利用信息熵方法进行数据分类，发现故障特征和故障

点之间的关系，为故障诊断提供支持。该方法直观、可用性强，但对数据预处理中故障特征的提取要求较高，故障数据需逐条处理，同时需要专家和设备经验丰富的设备管理员的支持，系统建设初期的成本较高，有一定的应用局限性。

参考文献：

[1] 刘小明, 李辉, 蒋吉兵, 等. 基于故障树和神经网络的火箭故障诊断方法[J]. 计算机仿真, 2010, 27(7): 38-42.

[2] Jiawei Han, Micheline Kambei. 数据挖掘概念与技术[M]. 北京: 机械工业出版社, 2007: 184-203.

[3] 邹媛. 基于决策树的数据挖掘算法的应用与研究[J]. 科学技术与工程, 2010, 10(18): 4510-4515.

[4] 裴小英. 机电设备智能故障诊断系统中的数据挖掘[J]. 装备制造技术, 2007(9): 80-81.

[5] 谭俊璐, 武建华. 基于决策树规则的分类算法研究[J]. 计算机工程与设计, 2010, 31(5): 1017-1019.

[6] 苏亚丁. 基于决策树的数据挖掘技术在口腔诊疗中的应用[D]. 石家庄: 河北科技大学, 2010: 3-10.

[7] 邢晓宇, 余建坤, 陈磊, 等. 决策树算法在学生考试成绩中的应用[J]. 云南民族大学学报(自然科学版), 2009, 18(1): 77-80.

[8] 杨睿通, 贺兴时, 李建辉, 等. 基于决策树的空间数据处理策略[J]. 西安工程大学学报, 2013, 27(1): 110-114.

[9] Qninlna J R. Induction of decision trees[J]. Machine Learning, 1986, 3(1): 81-106.

[10] Breslow L, Aha D W. Sim-pliying decision trees: A survey[J]. Knowledge Engineering Review, 1997, 12(1): 23-46.

****

（上接第 77 页）

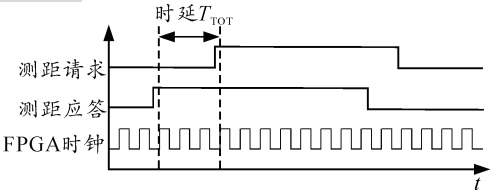

图 3　无线测距时间测量示意图

将 T_{TOT} 代入公式 (1) 中，其中 T_{TAT} 无法直接测量给出，影响 T_{TAT} 的因素主要是 FPGA 内部处理时延、基带模数/数模转换时延以及射频收发回路时延构成，但上述时延均较固定，波动范围小，可通过试验测量得到一个平均值代入公式计算，从而实现两点之间的距离测量。

4　测距误差分析

受电磁波波长对于距离解析度的影响，无线通信频率越高则测距精度越高，其最大误差可等效为 1 倍波长 λ，本无线通信模块采用 400 MHz 的频点。

FPGA 的内部时钟频率直接影响计数器的精确性，内部时钟频率越高，则测距精度越高。采用沿触发计数方式，最大误差可表示为时钟周期的一半 $1/2f$，本模块 FPGA 的内部时钟倍频可达 300 MHz。

基带数模/模数转换的误差主要受转换器件的影响，存在不确定性，采用高速转换器件时，转换过程的不确定时间可控制在 2 ns 以内，加上中间过程的部分误差累计总共可控制在 5 ns 内。

因此，整个无线测距模块的最大测距误差 σ 可表示为

$$\sigma=\lambda+1/2f+C/5\ \text{ns}=2.75\ \text{m}。$$

5　结束语

笔者采用异步应答时间测量方式的无线测距通信模块，在实现数据无线通信的基础上增加了无线测距功能，采用异步应答方式降低同步时序控制造成的时延，同时采用全双工通信架构，省去了收发转换导致的不可控时延，可有效提高测距精度。该方式在无线传感器网络测距及通信方面具有良好的应用前景。

参考文献：

[1] Girod L, Bychovskiy V, Elson J, et al. Locating Tiny Sensors in Time and Space: A Case Study[C]. Freiburg: IEEE, 2002: 214-219.

[2] 黄智伟. 射频电路设计[M]. 北京: 电子工业出版社, 2005: 170-174.

[3] 黄智伟. 无线通信集成电路[M]. 北京: 北京航空航天大学出版社, 2005: 200-205.

[4] 黄智伟. 无线发射与接收电路设计[M]. 北京: 北京航空航天大学出版社, 2007: 37-42.

[5] Maxim Company. MAX2510 Datasheet[R]. San Jose, California, USA: MAXIM Company. www. maxim-ic.com.