计算机科学1999Vol. 26№. 1

一个通用的系统故障诊断模型

An Universal Model of System Fault Diagnosis

何佳洲　周志华　陈兆乾

（南京大学计算机科学系　计算机软件新技术国家重点实验室　南京 210093）

Abstract　In this paper, fault diagnosis is studied as a classificatory problem, we constructed a universal fault model in order to grant the generality of the approach. The algorithm IHMCAP which combines symbolic and neural learning balances the uses of the dynamic information and the experimental knowledge in the system fault diagnosis. In the meantime, the incremental learning ability of the algorithm also can acquire highly diagnostic accuracy and capability of learning and diagnosis of the novel faults.

Keywords　Fault diagnosis, Machine learning, Neural network

狭义上，故障诊断包括两层含义即故障检测和故障定位，前者是指测试一个系统，判断其是否存在故障，而后者则是指在前者的基础上进一步定位故障点。广义上，故障诊断的内容，除了包含要判断系统有无故障以及故障产生后找到故障部位，还包含对故障特性的分析以及所采取的措施，例如让系统降低功能运行，或中断系统运行防止故障的传播和灾难性事故的发生。

故障诊断的主要方法有：依赖于动态模型的方法和不依赖于动态模型的方法。前者要求控制系统的动态过程能用数学模型来描述，因而应用受到很大限制。实际中广泛采用的是不依赖于动态模型的方法，其种类繁多，包括专家系统方法、故障树诊断方法、人工神经网络方法、模式识别方法、模糊数学方法等，然而以上各种诊断方法均不同程度地存在以下缺点：(1)缺乏通用性。依赖于动态模型的方法、专家系统的方法、故障树诊断方法与具体的系统密切相关，而不同的系统采用神经网络诊断时，网络类型与结构也不一样，因此，针对一种系统开发的诊断系统，在移植到其它系统时会遇到很大困难；(2)系统先验知识和动态数据的利用不均衡。表现在要么过分地依赖于系统的先验知识，如动态模型的方法、故障树诊断方法等，要么对系统的先验知识利用较少，如神经网络方法；(3)学习能力不足。动态模型的方法、故障树诊断方法等缺乏学习能力，而专家系统中的知识库系统搜索效率低的固有缺点，以及一般的神经网络在学习时易陷于局部极小，也导致这一类系统通常只有较低级的学习能力。

增量式 IHMCAP 算法，运用分类的观点处理故障诊断问题。其优点是：首先建立一个通用故障模型，通过寻找属性值与故障类型的对应关系对故障进行诊断，从而保证通用性；其次，该算法采用符号学习与神经网络相结合的策略，因此，不仅较合理地利用系统的先验知识，而且使之在与动态数据利用上取得了均衡；同时又由于其具有增量学习能力，可以对新增故障类型进行学习，因此确保了系统具有较强的学习能力。

1　IHMCAP 算法

IHMCAP 算法[1~4]方法采用离散属性优先的策略，结合神经网络算法，依据属于多个类别用离散属性和连续属性描述的实例集，进行多概念知识获取学习，生成结合神经网络的混合型二叉判定树。算法由符号学习和神经网络学习两部分组成。

符号学习部分对示例集的离散属性分量加以分析，得到判定树。处理过程如下：对每一个示例集 E，

何佳洲　硕士生，主要从事雷达数据实时处理、神经网络等方面的研究工作。周志华　硕士生，主要从事神经网络、机器学习等方面的研究工作。陈兆乾　教授，主要从事专家系统、神经网络、机器学习等方面的研究工作。

我们可以得到一个条件，根据 E 中的示例满足条件与否将此集合中的元素划分为两部分，分别称为 E^+ 和 E^-。条件形式通常是 $a_i=v$，(a_i 是示例的属性 A_i 的取值，$v\in Dom(A_i)$)。满足此条件的示例划分到 E^+，其余的示例划分到 E^-。用相同的策略处理 E^+ 和 E^- 这两个新增加的示例集合，并且继续这个过程直到无法从示例集中构造规则为止。最后就可以得到一棵二叉树，树叶结点都表示某一个概念，树中的其它结点都代表某个条件，我们称之为判定树。以下为算法的形式描述。

定义1 集合 $E=\{x|c=\langle c,a_1,\cdots,a_n\rangle, c\in C, a_i\in A_i\}$

我们称 E 为示例集，E 中的元素 x 为示例，集合 C 为概念，A_i 为属性；n 为示例集合 E 的属性数，$|C|$ 为示例集合 E 中概念数，$|E|$ 为示例数；示例 x 的第 i 个分量为 x 的概念值，x 其它分量 a_i 为 x 属性 A_i 的取值。

定义2 $RL=\{\langle X,v\rangle, X\in\{C,A_1,\cdots,A_n\}, v\in X\}$，$RL$ 称为条件集合，$\langle X,v\rangle$ 称为条件。集合 $T=\{(i,r)|i\in N, r\in RL, RL$ 是条件集合$\}$，称为二叉判定树，如果 T 满足下列条件：

$\forall i,r(i,r)\in T$

$\exists fr,ybr,ebr(\langle\lceil\frac{i-1}{2}\rceil,fr\rangle\in T)\wedge(\langle 2\lceil\frac{i-1}{2}\rceil+1,ybr\rangle\in T)\wedge(\langle 2\lceil\frac{i-1}{2}\rceil+2,ebr\rangle\in T)$

其中，$\lceil\ \rceil$ 是取整函数。

输入：示例集合 E

输出：二叉判定树 T

0. $N\leftarrow\{(E,0)\}$，$T\leftarrow\varnothing$

1. $(S,No)\leftarrow$SelectE(N)，$r\leftarrow$Formula(S)

2. WHILE NOT(Void(r)) DO

(I) $S^+\leftarrow\{x|x\in S$ 且 x 满足条件 $r\}$，$S^-\leftarrow\{x|x\in S$ 且 x 不满足条件 $r\}$

(II) $T\leftarrow T\cup\{(No,r)\}$；$N\leftarrow N\cup\{(S^+,2*No+1),(S^-,2*No+2)\}-\{(S,No)\}$

(III) $(S,No)\leftarrow$SelectE(N)，$r\leftarrow$Formula(S)

3. STOP

其中，$N=\{(Ex,n)|Ex\subseteq E, n\in N+\}$；

SelectE$\in f:\{N\}\rightarrow N$，SelectE(N)从当前判定树中选择一个需要继续划分的叶结点；

Formula$\in f:N\rightarrow RL$，Formula(S)根据示例集生成划分条件；

Void$\in f:RL\rightarrow$BOOLEAN，Void(r)判断条件是否为空

符号学习之后是人工神经网络(ANN)学习，我们用 ANN 处理符号学习算法难以处理或处理效率不高的分类问题。

判定树划分到一定程度之后，示例集可能会因所有的属性都曾经用来作为划分条件而找不到新的划分条件，而无法继续划分。也可能因其他原因(如剩余属性皆为连续属性)，规则生成算法无法构造新的划分条件。同时此示例集中的示例并不属于同一个概念，这时我们就可以在此示例集合所属的判定树的树叶结点上引入 ANN，对此树叶上的示例进行分类。

这里使用的 ANN 为 FTART[2]，是一种自适应谐振神经网络算法，该算法只需一遍学习，在样本数较少时也可实现样本空间的有效划分。同时，它在处理新增的输入模式时，与传统前馈型的 BP 算法不同，不再重新生成网络连结权，只需在网络的第二层适当增加神经元，将其与部分已有的神经元相连，再适当调整新增连接权，即可覆盖新增模式。

2 基于 IHMCAP 算法的故障诊断

2.1 故障模型[5,6]

故障在系统运行时，以差错和失效的形式表现出来。系统失效则系统必然出现差错，反之则不然。就像系统出了差错后不一定失效一样，故障系统有时也可能不出差错。最好的故障诊断方法是在系统尚未出差错时，就将故障诊断出来，例如，修正软件的编译错误、事先发现电路上的虚焊等。事实上，相当数量的故障，直到系统出现差错、失效以至于崩溃时，人们才会意识到它们的存在。因此，人们通常更关心动态情形下对系统故障的诊断。

令 FS 是系统 S 的故障集，$FS=\{f_0,f_1,f_2,\cdots,f_n\}$，$f_0$ 表示无故障，这里将无故障作为一种特殊的故障类型。

$AS_i=\{A^i|A^i$ 属性作为 S 的动态属性，能表示由故障 f_i 导致的差错$\}$

$IN\_AS=\{A^i|A^i$ 属性作为 S 的输入属性，能决定 S 动态特性$\}$

$$AS'=\bigcup_{i=0}^{n}AS_i,\quad AS=AS'\cup IN\_AS$$

由此得到系统 S 的故障集 FS 对应的属性集 AS，不妨设 $AS=\{A_1,A_2,\cdots,A_m\}$。$t$ 时刻诊断出系统 S 的故障 f_i 即可表示为：$\exists\Phi$ 使得下式成立：

$$\Phi(A_1(t),A_2(t),\cdots,A_m(t))=f_i,\ i=0,1,\cdots,n \tag{1}$$

式中 $A_j(t)$ 表示属性 A_j 在时刻 t 的取值，$j=1,\cdots,m$。

Φ 也就代表具体的故障诊断方法。若把 Φ 看作是系统专家进行的诊断，则其自变量的取值应是属性集 AS 在 $0\sim t$ 时间段上的所有取值，而诊断结果则可能不仅仅是某个 f_i，而是 FS 中若干个故障的逻辑运算。因此，从本质上讲，故障诊断就是寻找故障的表现形式（差错）到故障本身的映射。

2.2 诊断过程

具体诊断过程共分五步，叙述如下：

（1）选择学习样本。随机地抽取一定数量的属性-故障值对，以尽可能覆盖故障及差错空间为准，以此作为学习和训练例子。

（2）初步学习训练。对例子数据用 FTART 网络进行学习，然后，从属性集中删除那些权值很小的作用不大的属性，从而使属性个数得以压缩，以减少决策树和神经网络的规模。

（3）部分属性离散化。对取值范围较小的整型属型进行离散化处理，减少连续属性的个数，同时也为符号学习创造条件。

（4）学习训练和增量学习。对例子数据，用 IHMCAP 算法进行学习，形成决策树和神经网络。对新增的属性-故障值数据，在已有的决策树和神经网络的基础上，进行增量学习。

（5）验证诊断效果。用其余属性-故障值对，对已

（下转第84页）

（上接第90页）

③标识主题：主题提供了一个控制读者（分析员、管理员或客户）在一段时间内所考虑和理解模型的多少部分的机制，同时也给出了 OOA 模型中各图的概况。

④定义属性：属性是描述对象或分类结构实例的数据单元。

⑤定义服务：为每个对象和分类结构定义所需要的行为。

（3）特点：过程抽象；数据抽象；信息隐蔽；继承。

6.3 OOD

（1）基本思想：根据 OOA 的结果，对系统进一步细化。

（2）基本步骤：

①设计问题域部分：进一步设计需计算机处理的系统范围。

②设计人机交互部分：针对属性和服务，设计人机交互方式。

③设计任务管理部分：标识和设计任务及任务中的有关服务。

④设计数据管理部分：提供数据管理中对象的检索及存贮的基础。

6.4 OOPL

OOPL 是一门代码组装技术，符合人们最终获得软件 IC 的愿望，它的主要特点是引入了类机制、封装、继承、多态性、动态联编等概念来进行程序设计。

实现 OOA 和 OOD 并不一定非要求有 OOPL，但 OOPL 可提供对 OOA 和 OOD 很有效的支持。

6.5 OO 方法的主要优点和局限性

（1）优点：①归纳和演绎思想的综合体现。②问题空间和解空间的同构。③继承机制的引入，很好地支持了重用性。④对象机制有力地支持了信息隐藏的概念。⑤多态性、持久性和动态联编对程序设计起到了很好的作用。

（2）局限性：OO 方法从计算机角度看有它巨大的优势，但 OO 方法在建立客观系统模型方面有不足之处。OOA 一开始就有很多计算机方面的术语和概念不容易被一般用户或参与应用软件开发的业务人员所了解，即使了解了，也很难正确使用，为真正掌握这些概念需要有一定的计算机背景知识，所以 OOA 在应用软件开发中，建立客观系统的描述方面不能被普遍接受和推广使用。

结束语 结构化分析和设计方法使用最早也最广，在我国目前的软件开发中仍具有生命力，原型法在我国得到了一定程度的应用。随着 Windows 95及其环境的推广应用，为面向对象的方法提供了支持，使它逐渐为人们接受和使用。

参 考 文 献

1 潘锦平. 软件开发技术. 上海科学技术出版社，1985

2 罗晓沛、侯炳辉. 系统分析员教程. 清华大学出版社，1992

3 Cameron J R. Basic Concepts and Notation of JSD. 1986

4 詹姆斯 马丁. 战略数据规划方法学. 北京，1987

5 绍维忠等译. 面向对象的分析. 北京大学出版社，1992

6 汪成为等. 面向对象的分析、设计及应用. 国防工业出版社，1992

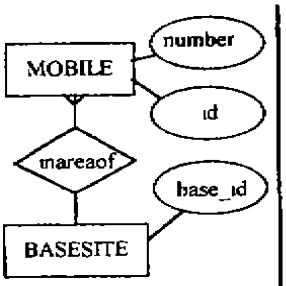

图4 接电话情形实例的实体关系描述

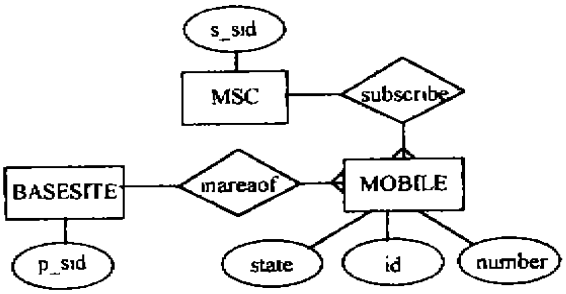

图5 初始化连接情形实例的实体关系描述

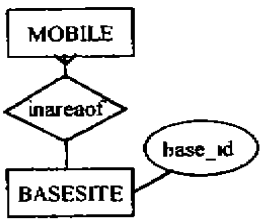

图6 转移呼叫情形实例的实体关系描述

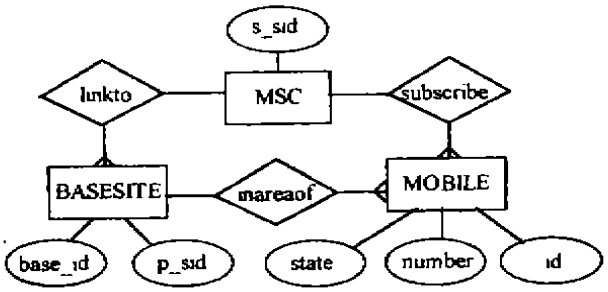

图7 需求定义的实体关系模型

参 考 文 献

1 Jacobson I, et al. Object Oriented Software Engineering. A Use Case Driven Approach. Addison-Wesley, 1992

2 Regnell B, et al. Improving the use driven approach to requirements engineering. Proc. RE'95, 1996

3 Anderson J S, Durley B. Using scenarios in deficiency-driven requirements engineering. Proc. RE'93

4 金凌紫，等. 情形实例驱动的软件需求模型自动生成. 计算机学报，已录用

5 张朝良，等. 情形实例描述一致性与完备性检查和需求模型的自动综合算法. 计算机研究与发展，已录用

6 Jin Lingzi, Zhu Hong. Description and Analysis of Use Scenarios in Requirements Engineering. Proc. SEKE'98, In press

（上接第87页）

形成的决策树和神经网络进行验证，当正确率不满足要求时，则将那些不能进行正确诊断的值对按一定的比例增加到例子中，返回(4)重新进行学习，直到诊断精度满足要求。得到一个最终的故障学习诊断模型。

结论 本文在故障集和与之对应的差错属性集的基础上，构造系统的故障模型，用 IHMCAP 算法寻找属性值与故障类型之间的对应关系，由此对故障进行诊断。理论分析及应用表明，该方法不仅精度高、通用性好、学习能力强，而且在利用系统的先验知识与动态数据上也取得了均衡。今后，我们的研究课题是如何将该算法应用于系统故障的在线诊断。

参 考 文 献

1 陈兆乾，刘宏，等. 一种混合型多概念获取算法 HMCAP 及其应用. 计算机学报，1996，19(10)，753～761

2 陈兆乾，周戎，等. 一种新的自适应谐振算法 FTART. 软件学报，1996，7(8)：458～465

3 陈兆乾，周志华，等. 混合型学习模型 HLM 中的增量学习算法. 软件学报，1997，8(11)

4 周志华，陈兆乾. HMCAP 的新实现方法及其应用. 微型计算机，1997，17(1)：30～32

5 胡 谋. 计算机容错技术. 中国铁道出版社，1995

6 杨士元. 数字系统的故障诊断与可靠性设计. 清华大学出版社，1989