故障诊断专家系统的知识网络设计*

黄 波 倪重匡 高丽萍

(复旦大学计算机系 上海 200433)

摘要 该文针对识别与分类两种故障诊断模型,提出了一种知识网络的结构设计方法,并给出了详细的推理算法与故障判决算法.此种知识网络通过消息传递来实现不精确推理,具有分布式特性.该文还提出了网络结点判决函数的机器学习算法,采用卷包裹算法来实现判决函数边界点集的选取,减少了曲线拟合的计算量.

关键词 故障诊断专家系统,知识网络,可信度对,判决函数,机器学习.

中图法分类号 TP391

在故障诊断领域中,人们不得不佩服领域专家在问题诊断过程中所表现出来的快速反应与惊人的洞察力.作为对领域专家的模拟,故障诊断专家系统无疑扮演着一个重要的角色.如何使故障诊断专家系统从最少量的数据快速、准确地得出正确的诊断结果,达到与人类专家相近,甚至更高的诊断水平,正是故障诊断专家系统的设计者们所热衷于探索的问题.[1,2]

人类知识呈网状结构,因此,故障诊断专家系统的知识也可用网状模型来表示,但由于故障诊断领域所固有的某些特征,这种知识网络又不同于一般的神经网络.一般的神经网络通过神经元的变换函数来实现从输入到输出的变换,并且每个神经元的变换函数都相同.[3]本文所介绍的知识网络在网络结点与弧的表示方法及推理过程中的消息传播方法上,都有其独有的特点,各网络结点的判决函数也不相同,从而形成一种新型的知识网络模型.

1 知识网络结构

在介绍故障诊断专家系统的知识网络模型之前,先阐述两个基本概念:现象与故障.现象是指诊断对象的正常或不正常状态.通过测试点值,可直接对其正常与否进行估计.故障是指无法通过测试点值进行直接判定的诊断对象的不正常状态.

故障诊断专家系统的任务便是利用测试点值和知识网络的推理,最后判决出诊断对象所存在的故障.但是现象与故障间并不一定存在明显的推理关系,因此便引进了许多中间假设.通过众多的中间假设把现象与故障联系起来,从而更精确地表达领域知识.

1.1 网络结点

与现象、中间假设、故障相对应,知识网络中的结点可分为3层:现象结点层、中间结点层及故障结点层,如图1所示(各种弧的具体含义见下文).

对于知识网络中的每个结点 N,都有一个正的可信度 $e_N^{\rightarrow}$ 和一个负的可信度 $e_N^{\leftarrow}$,且满足 $e_N^{\rightarrow}, e_N^{\leftarrow} \in [0,1]$.若 $e_N^{\rightarrow}=1$, $e_N^{\leftarrow}=0$,表示结点 N 所代表的假设完全正确;若 $e_N^{\rightarrow}=0, e_N^{\leftarrow}=1$,则表示结点 N 所代表的假设完全错误.这两个可信度的笛卡尔积构成了一个正方形区域,此正方形区域又可被不相交的3条曲线 c_1, c_2, c_3 划分为4个小区域(如图2所示).$(e_N^{\leftarrow}, e_N^{\rightarrow})$ 落入 TRUE, FALSE, CONFLICT 或 UNKNOWN 区域,分别表示此结点所对应的假设可认为是真、假、出现矛盾、或真假性不可判定.

设 c_1, c_2, c_3 的曲线方程分别为 $c_1: e_N^{\leftarrow}=f_N^{\rightarrow}(e_N^{\rightarrow})$, $c_2: e_N^{\leftarrow}=f_N^{\leftarrow}(e_N^{\leftarrow})$, $c_3: e_N^{\leftarrow}=f_N(e_N^{\rightarrow})$,称 $f_N^{\rightarrow}, f_N^{\leftarrow}, f_N$ 为网络结点 N 的判决函数.p^+(坐标为 $(0, e_N^{\leftarrow+})$)是 $f_N^{\rightarrow}$ 与 $e_N^{\leftarrow}$ 轴的交点,$e_N^{\leftarrow+}$ 表明了当结点 N 所代表的假设的负的可信度为0时,假设可认为是真所要求的正的可信度的下界;p^-(坐标为 $(1, e_N^{\rightarrow-})$)是 $f_N^{\leftarrow}$ 与直线 $e_N^{\rightarrow}=1$ 的交点,表明了当结点 N 所代表的假设

---

* 作者黄波,1973年生,博士生,主要研究领域为并行编译,人工智能,模式识别.倪重匡,1944年生,教授,主要研究领域为人工智能与信息处理,模式识别.高丽萍,女,1973年生,硕士生,主要研究领域为人工智能与信息处理.

本文通讯联系人:黄波,上海 200433,复旦大学计算机系

本文1997-03-15收到原稿,1997-06-26收到修改稿

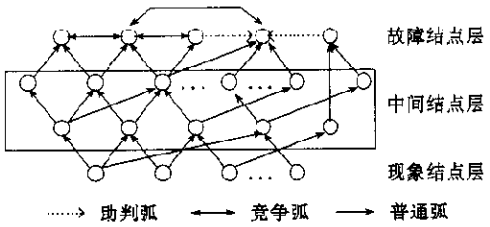

图1　知识网络的层次结构

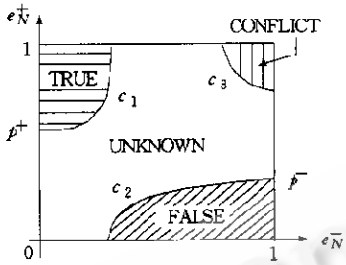

图2　网络结点的判决函数对判决区域的划分

的负的可信度为1时,假设可认为是假所要求的正的可信度的上界.对于每个网络结点 N,$f_{\bar{N}}$,$f_{\bar{N}}$,f_N 都可能不同.在推理过程中,结点的可信度对$(e_{\bar{N}},e_{\bar{N}})$具体落入哪个区域是很重要的,这将决定网络结点的信息传播方向.

1.2　网络中的弧

根据所连的结点及在故障诊断推理中所起的作用,此类知识网络中的弧可分为普通弧、竞争弧、助判弧3类.

(1) 普通弧:存在于现象结点与中间结点、中间结点与中间结点、中间结点与故障结点之间,用单向实箭头弧表示(见图1).设弧上的权重为 $R_{ij}(R_{ij}\in[-1,1])$,代表置信度,表示从前置结点所代表的假设推理到后继结点所代表的假设的可信程度.从每个 N_i(现象结点或中间结点)出去的普通弧又可分为 t 弧和 f 弧两类.它们分别代表$(e_{\bar{N}},e_{\bar{N}})$落入TRUE区域和落入FALSE区域时推理信息的传播方向.

(2) 竞争弧:存在于两个故障结点之间,用双向实箭头弧表示(见图1).设弧上的权重为 $C_{ij}(C_{ij}\in[0,1])$.若两个故障结点 N_i,N_j 间存在一条竞争弧,则表示 N_i 所代表的假设与 N_j 所代表的假设相互影响.只要一方的可信度对落入TRUE区域,便要促使对方的 $e_{\bar{N}}$ 扩大,从而使对方的可信度对落入FALSE区域的可能性增大.

(3) 助判弧:存在于两个故障结点之间,用单向虚箭头弧表示(见图1).设弧上的权重为 H_{ij},其中 $H_{ij}\in[0,1]$.若两个故障结点 N_i,N_j 间存在一条从 N_i 到 N_j 的助判弧,则表示若 N_i 的$(e_{\bar{N}_i},e_{\bar{N}_i})$落入TRUE区域内,则将以一定的方式促使 N_j 中的 $e_{\bar{N}_j}$增大,从而使$(e_{\bar{N}_j},e_{\bar{N}_j})$落入TRUE区域的可能性增大.

2　决策推理与故障判决

以上述的知识网络为基础构造的故障诊断专家系统的推理判决结构可分为初始化、推理控制和故障判决3个模块.初始化模块负责从数据采集器中读入测试点数据,经过模糊化运算后送入知识网络的现象结点,初始化消息队列,并对其他网络结点进行初始化;推理控制模块通过对网络结点间所传送消息的处理与控制,逐层推理出可能发生的故障,并把可能发生的故障放入故障判决队列;故障判决模块经过竞争与助推判决,选择出最有可能发生的故障集合.具体控制结构如图3所示,下面对这3个模块逐一详细介绍.

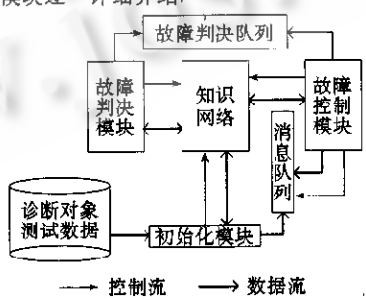

图3　新型知识网络的推理控制结构

2.1　网络初始化模块

网络初始化模块负责把测试点的数据值转化到[0,1]上,这可以通过模糊化算子来实现.当然,对于不同的应用领域,或同一应用领域的不同现象结点,可定义不同的模糊化算子,通过这些模糊化算子的运算,便完成了现象结点可信度对的初始化,对于中间结点和故障结点,令 $e_{\bar{N}}=e_{\bar{N}}=0$.除了对知识网络结点的可信度对进行初始化外,初始化模块

还负责完成消息队列的初始化.每一个消息是如下的一个结构.

(接收消息的结点名称 N,要传递给消息接收结点的可信度 X)

对消息队列的操作,可用以下两个函数来实现.

SendMessage(N,X):把(N,X)放入消息队列尾部

GetMessage(N,X):从消息队列首部读取消息(N,X)

对于任一个现象结点 N_i(如图 4 所示),有如下规定:

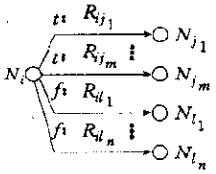

图4 消息传送示例结点图

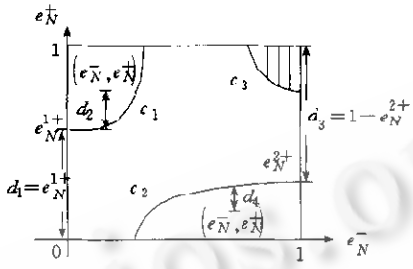

图5 消息传送中可信度的计算

如果($e_{N_i}^-$,$e_{N_i}^+$)落在 TRUE 区域内,则它必须向所有以 t 弧连接的后继结点 N_{j_p}($1\leqslant p\leqslant m$)发送消息(N_{j_p},X_1),其中 $X_1=R_{ij_p}*(e_{N_i}^{1+}+e_{N_i}^+-f_{N_i}^+(e_{N_i}^-))$.(括号内和式的含义即为图 5 左方的 d_1+d_2,$d_1=e_{N_i}^{1+}$,$d_2=e_{N_i}^+-f_{N_i}^+(e_{N_i}^-)$).

如果($e_{N_i}^-$,$e_{N_i}^+$)落在 FALSE 区域内,则它必须向所有以 f 弧连接的后继结点 N_{l_q}($1\leqslant q\leqslant n$)发送消息(N_{l_q},X_2),其中 $X_2=R_{il_q}*(1-e_{N_i}^{2+}+f_{N_i}^+(e_{N_i}^-)-e_{N_i}^-)$(括号内和式的含义即为图 5 右方的 d_3+d_4,$d_3=1-e_{N_i}^{2+}$,$d_4=f_{N_i}^+(e_{N_i}^-)-e_{N_i}^+$).

按照以上两条规定,当现象结点的可信度对初始化完成后,初始化模块便会通过 SendMessage 函数向消息队列中填入消息,等待推理控制模块的处理.

2.2 推理控制模块

推理控制模块通过处理消息队列中的消息以及控制各中间结点的消息发送来完成故障的初步判决.对于每个中间结点,规定只有当它接收完它的所有前置结点发送过来的消息后,才能向它的后继结点发送消息,所发送消息中可信度的计算同现象结点.当一个结点 N 接收一个它的前置结点发送过来的消息(N,X)后,(e_N^-,e_N^+)以如下的方式发生变化:若 $X\geqslant 0$,则 $e_N^+=e_N^+ + X*(1-e_N^+)$;若 $X<0$,则 $e_N^-=e_N^- - X*(1-e_N^-)$.具体的推理算法见算法 1,其中 CONFLICTSIGN为一全局变量,当它等于 TRUE 时,说明推理中出现矛盾,这有可能是初始测试数据采集有误,也有可能是由于知识网络不太完整.

算法 1. 推理算法

```
while (消息队列不为空)do {
    GetMessage(N,X);
    if (X≥0) then
      e_N^+ = e_N^+ + X*(1-e_N^+);
    else e_N^- = e_N^- - X*(1-e_N^-);
    if (结点 N 已接收完它的所有前置结点发送过来的消息) then
      if (结点 N 为故障结点) then
        把结点 N 送入故障判决队列;
      else if (e_N^- 在 f_N^+ 的定义域内且 e_N^+ > f_N^+(e_N^-)) then {
            e_N = e_N^{1+} + e_N^+ - f_N^+(e_N^-);
            for (每一个跟 N 以 t 弧相连的后置结点 N_i) do {
                X = e_N * R_i;
                SendMessage(N_i,X);
            }
        } else if (e_N^+ 在 f_N^- 的定义域内且 e_N^+ < f_N^-(e_N^-)) then {
              e_N = 1 - e_N^{2+} + f_N^-(e_N^-) - e_N^+;
              for (每一个跟 N 以 f 弧相连的后置结点 N_j) do {
                  X = e_N * R_j;
                  SendMessage(N_j,X);
              }
          } else if (e_N^- 在 f_N 的定义域内且 e_N^+ > f_N(e_N^-))
                {CONFLICTSIGN = TRUE;break;}
}
```

2.3 故障判决模块

经过推理控制模块的推理,故障判决队列中已放入了可能发生的故障结点,故障判决模块通过一定的判决以及对竞争弧与助判弧的处理,最后判决出最有可能发生的故障集合.

判决规则 1. 若结点 N_i 与 N_j 为故障结点,且两结点间有一竞争弧连接,弧上的权重为 C_{ij},则当 $(e\bar{N}_i, e\bar{N}_i)$ 落入 TRUE 区域时,定义 $e\bar{N}_j$ 以如下的方式递增.

$$e\bar{N}_j = e\bar{N}_j + (1-e\bar{N}_j) * C_{ij} * (e\bar{N}_i + e\bar{N}_i - f\bar{N}_i(e\bar{N}_i)) \tag{1}$$

结点间的竞争应是相互的,当 $e\bar{N}_j$ 以公式(1)的方式递增后,若 $(e\bar{N}_j, e\bar{N}_j)$ 不再落入 TRUE 区域内,则竞争结束.否则,由于 $(e\bar{N}_j, e\bar{N}_j)$ 落入 TRUE 区域,$e\bar{N}_i$ 也以与公式(1)相似的方式递增,如此反复下去,必然使 $(e\bar{N}_j, e\bar{N}_j)$ 和 $(e\bar{N}_i, e\bar{N}_i)$ 向各自的 TRUE 区域的边缘移动.若经过有限步,两结点中的任一个不再落入 TRUE 区域内,或者有一个节点 $e\bar{N}$ 的增加量小于某个很小的正数 ε(ε 可以在进行竞争判决前通过人机交互设定),则结束竞争.

判决规则 2. 若存在从结点 N_i 到结点 N_j 的助判弧,弧上的权重为 H_{ij},且 $(e\bar{N}_i, e\bar{N}_i)$ 落在 TRUE 区域内,$(e\bar{N}_j, e\bar{N}_j)$ 落入 FALSE 区域,则定义 $e\bar{N}_j$ 以如下方式增加.

$$e\bar{N}_j = e\bar{N}_j + (1-e\bar{N}_j) * H_{ij} * (e\bar{N}_i + e\bar{N}_i - f\bar{N}_i(e\bar{N}_i)) \tag{2}$$

有了以上两个判决规则,故障判决算法便可描述如下.

算法 2. 故障判决算法

STEP1. 把故障判决队列中那些 $(e\bar{N}, e\bar{N})$ 落在 FALSE 或 UNKNOWN 区域,且不与任何其他故障结点有弧连接的结点从故障判决队列中删去,然后转 STEP2.

STEP2. 若故障判决队列中无竞争结点对,则转 STEP3,否则,取一竞争结点对 N_i, N_j,置 HASACTION 为 FALSE.

(A) 若 $(e\bar{N}_i, e\bar{N}_i)$ 落入 CONFLICT 区域中,置 CONFLICTSIGN 为 TRUE,停止判决;若 $(e\bar{N}_i, e\bar{N}_i)$ 落入 FALSE 或 UNKNOWN 区域中,且不存在从其他故障结点到 N_i 的助判弧,则把 N_i 从故障判决队列中删去,置 HASACTION 为 TRUE,转(B).

(B) 若 $(e\bar{N}_j, e\bar{N}_j)$ 落入 CONFLICT 区域中,置 CONFLICTSIGN 为 TRUE,停止判决;若 $(e\bar{N}_j, e\bar{N}_j)$ 落入 FALSE 或 UNKNOWN 区域中,且不存在从其他故障结点到 N_j 的助判弧,则把 N_j 从故障判决队列中删去,置 HASACTION 为 TRUE,转(D);否则,转(C).

(C) 若 $(e\bar{N}_i, e\bar{N}_i)$ 与 $(e\bar{N}_j, e\bar{N}_j)$ 皆落在 TRUE 区域内,根据判决规则 1,N_i, N_j 经竞争判决后,或者有一结点(记为 N_*)不再落入 TRUE 区域内,或者两结点仍落在 TRUE 区域内.若 $(e\bar{N}_*, e\bar{N}_*)$ 落入 CONFLICT 区域中,置 CONFLICTSIGN 为 TRUE,停止判决;若 $(e\bar{N}_*, e\bar{N}_*)$ 落入 FALSE 或 UNKNOWN 区域中,且不存在从其他故障结点到 N_* 的助判弧,则把 N_* 从故障判决队列中删去,置 HASACTION 为 TRUE,转(D).

(D) 重复 STEP2,直到故障判决队列中无竞争结点对,或者对故障判决队列中的所有竞争结点对,HASACTION 为 FALSE,然后转 STEP3.

STEP3. 在故障判决队列中删去如下这种结点 N_*:$(e\bar{N}_*, e\bar{N}_*)$,落入 FALSE 或 UNKNOWN 区域中,且不存在从其他故障结点到 N_* 的助判弧,然后转 STEP4.

STEP4.

REPEAT

从故障判决队列中选取这样的结点 N_*:$(e\bar{N}_*, e\bar{N}_*)$,落入 FALSE 或 UNKNOWN 区域中,并且对于它的所有现存助判弧的前置结点 $N_i (i=1,2,\ldots,m)$,$(e\bar{N}_i, e\bar{N}_i)$ 落入 TRUE 区域内,则

for i=1 to m

$$e\bar{N}_* = e\bar{N}_* + (1-e\bar{N}_*) * R_{i*} * (e\bar{N}_i + e\bar{N}_i - f\bar{N}_i(e\bar{N}_i))$$

(其中 R_{i*} 为从 N_i 到 N_* 的助判弧上的权重)若最终 $(e\bar{N}_*, e\bar{N}_*)$ 仍然落在 FALSE 或 UNKNOWN 区域内,则把 N_* 从故障判决队列中删去.

UNTIL 故障判决队列中的所有结点皆落在 TRUE 区域内

最后仍留在故障判决队列中的那些故障结点便是最有可能发生故障的结点集合.

3 网络结点判决函数的机器学习

从上文所介绍的知识网络的推理判决算法中可知,每个网络结点的判决函数在故障的推理判决中起着很重要的

作用,因此,如何从领域专家所给定的知识中总结出每个网络结点的判决函数 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$,$f_{\bar{N}}^{0}$ 便成为构造此类专家系统的一个重要问题.领域专家虽然不可能给出 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$,$f_{\bar{N}}^{0}$ 的具体形式,却能够根据经验判断出给定的可信度对$(e_{\bar{N}}^{-},e_{\bar{N}}^{+})$大致落在图2所示的哪个区域.于是在知识获取时,对于每个网络结点 N,我们可以得到相应的4个点集 $P_{\bar{N}}^{+}$,$P_{\bar{N}}^{-}$,$P_{\bar{N}}^{0}$,$P_{\bar{N}}^{*}$,它们分别表示领域专家认为落在TRUE区域、FALSE区域、CONFLICT区域及UNKNOWN区域内的点集.但从图2我们可以看出,在构造判决函数 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$,$f_{\bar{N}}^{0}$ 时,实际上只需要 $P_{\bar{N}}^{+}$,$P_{\bar{N}}^{-}$,$P_{\bar{N}}^{0}$3个点集.于是网络结点判决函数的机器学习便等价于如下的问题:对于每个网络结点 N,如何根据 $P_{\bar{N}}^{+}$,$P_{\bar{N}}^{-}$,$P_{\bar{N}}^{0}$3个点集拟合出能表征 c_1,c_2,c_3 曲线方程的判决函数 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$,$f_{\bar{N}}^{0}$.

3.1 判决函数的数学特征

从图2可以看出,判决函数 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$,$f_{\bar{N}}^{0}$ 的定义域与值域都是[0,1]的真子集.为了保证在故障判决过程中每个网络结点内的可信度对$(e_{\bar{N}}^{-},e_{\bar{N}}^{+})$唯一地落在TRUE,FALSE,CONFLICT及UNKNOWN4个区域之一,$f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$,$f_{\bar{N}}^{0}$ 还必须是定义域内的单值函数,且互不相交.除此之外,领域专家的经验表明,判决函数在各自的定义域内可近似地看成凸函数($f_{\bar{N}}^{+}$,$f_{\bar{N}}^{0}$ 下凸,$f_{\bar{N}}^{-}$ 上凸).由于定义域都是[0,1]的子集,实践证明,用四阶多项式就可以达到很好的逼近.[4]

3.2 拟合点集的选取

根据 $P_{\bar{N}}^{+}$,$P_{\bar{N}}^{-}$,$P_{\bar{N}}^{0}$3个点集拟合判决函数 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$,$f_{\bar{N}}^{0}$ 的关键在于拟合点集的选取.虽然拟合出来的曲线 $e_{\bar{N}}^{+}=f_{\bar{N}}^{+}(e_{\bar{N}}^{-})$与直线 $e_{\bar{N}}^{+}=1$及直线 $e_{\bar{N}}^{-}=0$围成的区域未必能包含 $P_{\bar{N}}^{+}$ 内的所有点,但由于领域专家的知识不可能完全正确,因此,除了有少数的几个点落在拟合出的TRUE区域外,对 $f_{\bar{N}}^{+}$ 的正确性并无多大的影响.$f_{\bar{N}}^{-}$ 与 $f_{\bar{N}}^{0}$ 的拟合也是这样.尽管如此,以下所描述的判决函数的机器学习算法总是尽量使拟合出来的 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$,$f_{\bar{N}}^{0}$ 与单位正方形的边界所围成的区域以最大的限度分别包含 $P_{\bar{N}}^{+}$,$P_{\bar{N}}^{-}$,$P_{\bar{N}}^{0}$.

现在先以 $f_{\bar{N}}^{+}$ 的学习为例,阐述一下拟合点集的选取方法.

由于 $f_{\bar{N}}^{+}$ 在定义域内是下凸函数,因此,可以用求凸包的卷包裹法求得点集 $P_{\bar{N}}^{+}$ 在 $f_{\bar{N}}^{+}$ 一侧的边界点集.为了使求凸包的执行时间减小,可以先对 $P_{\bar{N}}^{+}$ 依次进行以下的预处理.

排序:对 $P_{\bar{N}}^{+}$ 内的点集按 $e_{\bar{N}}^{-}$ 的增序排序.

扫描:用平行于 $e_{\bar{N}}^{+}$ 轴的等间隔扫描线对排序后的 $P_{\bar{N}}^{+}$ 进行初次边界筛选.第1条扫描线从 $e_{\bar{N}}^{-}=0$开始,然后扫描线依次往 $e_{\bar{N}}^{-}$ 轴的方向平移一个扫描间隔,在相邻的两条扫描线间(包括左侧的扫描线)选取一个 $e_{\bar{N}}^{+}$ 最小的点,所有这样的点组成边界点候选点集 $Q_{\bar{N}}^{+}$.经过扫描选取后的边界点候选点集 $Q_{\bar{N}}^{+}$ 内点的个数比 $P_{\bar{N}}^{+}$ 内点的个数有明显减少,此时便可开始运用求凸包的卷包裹法求得 $Q_{\bar{N}}^{+}$ 的凸包边界点集 $B_{\bar{N}}^{+}$.但有一点必须说明:在领域专家给定的 $P_{\bar{N}}^{+}$ 内,必须包含如前所述的点 $p^{+}:(0,e_{\bar{N}}^{+})$,$e_{\bar{N}}^{+}$ 满足如下的条件:$\forall\,(e_{\bar{N}}^{-},e_{\bar{N}}^{+})\in P_{\bar{N}}^{+}$,$e_{\bar{N}}^{+}\leqslant e_{\bar{N}}^{+}$.由 p^{+} 的定义及边界点候选点集 $Q_{\bar{N}}^{+}$ 的扫描选取方法可知,$Q_{\bar{N}}^{+}$ 必包含 p^{+}.在用求凸包的卷包裹法求 $Q_{\bar{N}}^{+}$ 的凸包边界点集 $B_{\bar{N}}^{+}$ 时,p^{+} 点便是算法的起始点.按逆时针方向求得凸包边界点集 $B_{\bar{N}}^{+}$,对 $B_{\bar{N}}^{+}$ 内每一点进行边界扩展(即把每一点的 $e_{\bar{N}}^{+}$ 值减小一个很小的正数 ε,以期使拟合出来的 TRUE 区域尽量包含 $P_{\bar{N}}^{+}$ 内的点),从而得到 $f_{\bar{N}}^{+}$ 的拟合点集 $B_{\bar{N}}^{+^{+}}$.

$f_{\bar{N}}^{-}$,$f_{\bar{N}}^{0}$ 的拟合点集的选取与 $f_{\bar{N}}^{+}$ 相似,限于篇幅,这里不再赘述.

3.3 不相交测试

通过对拟合点集 $B_{\bar{N}}^{+^{+}}$,$B_{\bar{N}}^{+^{-}}$ 及 $B_{\bar{N}}^{+^{0}}$ 运用最小二乘法便可分别拟合出 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$ 及 $f_{\bar{N}}^{0}$.从前面的讨论可知,$c_1:e_{\bar{N}}^{+}=f_{\bar{N}}^{+}(e_{\bar{N}}^{-})$,$c_2:e_{\bar{N}}^{+}=f_{\bar{N}}^{-}(e_{\bar{N}}^{-})$,$c_3:e_{\bar{N}}^{+}=f_{\bar{N}}^{0}(e_{\bar{N}}^{-})$应互不相交.因此,对拟合出来的 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$ 及 $f_{\bar{N}}^{0}$,还应进行不相交测试.

不相交测试主要是检查以下3个方程:$f_{\bar{N}}^{+}(e_{\bar{N}}^{-})=f_{\bar{N}}^{-}(e_{\bar{N}}^{-})$,$f_{\bar{N}}^{-}(e_{\bar{N}}^{-})=f_{\bar{N}}^{0}(e_{\bar{N}}^{-})$和 $f_{\bar{N}}^{+}(e_{\bar{N}}^{-})=f_{\bar{N}}^{0}(e_{\bar{N}}^{-})$在各自公共的定义域内是否有解.只要任一方程有解,便要求领域专家对 $B_{\bar{N}}^{+^{+}}$、$B_{\bar{N}}^{+^{-}}$ 及 $B_{\bar{N}}^{+^{0}}$ 上位于解两边的点的 $e_{\bar{N}}^{+}$ 进行适当的纠正,然后再对修改后的 $B_{\bar{N}}^{+^{+}}$,$B_{\bar{N}}^{+^{-}}$ 及 $B_{\bar{N}}^{+^{0}}$ 重新进行最小二乘拟合,直到3个方程在各自公共的定义域内都无解为止.

3.4 学习算法的验证

下面通过实验来说明判决函数学习算法的正确性.

图6是取 $f_1(e_{\bar{N}}^{-})=1.5-\cos(\pi*e_{\bar{N}}^{-})$,$f_2(e_{\bar{N}}^{-})=\lg(e_{\bar{N}}^{-}+0.4)$,$f_3(e_{\bar{N}}^{-})=8^{(0.7-e_{\bar{N}}^{-})}$时学习出来的判决函数与基函数的图形比较.这3个函数分别具有 $f_{\bar{N}}^{+}$,$f_{\bar{N}}^{-}$ 及 $f_{\bar{N}}^{0}$ 的性质,其定义域分别为$0\leqslant e_{\bar{N}}^{-}\leqslant 1/3$,$0.6\leqslant e_{\bar{N}}^{-}\leqslant 1$和$0.7\leqslant e_{\bar{N}}^{-}\leqslant 1$.在本次实验中,扫描线间隔为0.005,拟合点集 $P_{\bar{N}}^{+}$,$P_{\bar{N}}^{-}$ 与 $P_{\bar{N}}^{0}$ 是随机生成的.表1是相应的学习误差,由在各函数的定义域内取等间隔的点计算而得,计算公式如下.

$$E_1=\sum_{j=1}^{m_1}|f_{\bar{N}}^{+}(e_{j\bar{N}}^{-})-f_1(e_{j\bar{N}}^{-})|,\quad E_2=\sum_{j=1}^{m_2}|f_{\bar{N}}^{-}(e_{j\bar{N}}^{-})-f_2(e_{j\bar{N}}^{-})|,\quad E_3=\sum_{j=1}^{m_3}|f_{\bar{N}}^{0}(e_{j\bar{N}}^{-})-f_3(e_{j\bar{N}}^{-})|,$$

$m_i(i=1,2,3)$是误差计算的点数,其中 $m_1=66$,$m_2=80$,$m_3=61$.

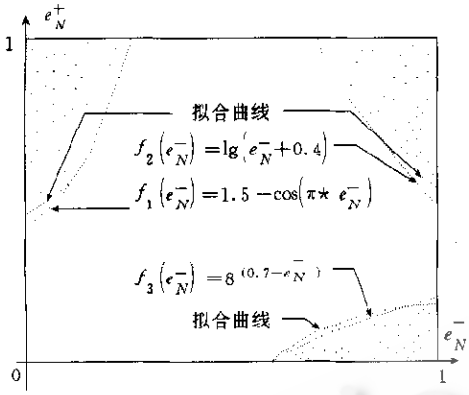

图6　判决函数的拟合结果

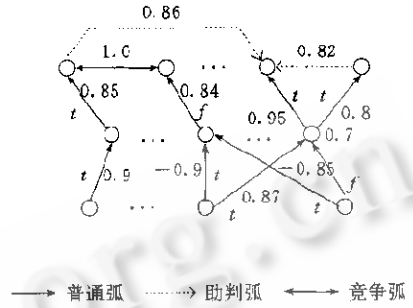

图7　一个实验知识网络模型

表1

基函数	机器学习所得的判决函数	学习误差
$f_1(e_N^-)=1.5-\cos(\pi * e_N^-)$	$-33.608\,905x^4-25.777\,193x^3-1.665\,5x^2+0.485\,644x-0.500\,57$	0.006 293
$f_2(e_N^-)=\lg(e_N^-+0.4)$	$1.145\,782x^4-2.931\,571x^3+2.125\,624x^2+0.191\,459x-0.395\,43$	0.005 785
$f_3(e_N^-)=8^{(0.7-e_N^-)}$	$2.041\,684x^4-1.550\,49x^3-1.919\,53x^2+0.085\,26x+1.921\,924$	0.009 446

4　模拟实验

下面通过实验来验证上文所述的故障诊断知识网络的推理与故障判决算法的正确性.

图7的知识网络取自某一仿真实验室故障诊断专家系统,弧上的数字代表权值,t 与 f 表示此弧的类型.每个结点的判决函数通过上述判决函数的机器学习算法而得到.

此实验的初始化数据来源于一个仿真实验室故障诊断对象.在实际的诊断对象中采集一大批数据,通过正态模糊化算子完成数据的初始化工作.实验的目的是检验通过此知识网络的推理诊断所得的结果与实际的结果是否相符.

我们取了50组数据,其中25组是实际发生故障的数据(一组数据可能导致多个故障发生),其他25组是没有任何故障发生的数据.实验结果见表2、3.

表2　25组实际发生故障的数据的诊断结果

诊断出的故障与实际发生的故障完全吻合	只诊断出部分故障	把实际未发生的故障误诊断为已发生故障	没有诊断出任何故障
22组	2组	0组	1组

表3　25组实际未发生故障的数据的诊断结果

诊断为无任何故障发生	诊断为有故障发生
24组	1组

以上两个表格的诊断结果表明:虽然存在误判,但此种知识网络的推理与故障判决算法从总体上来说是有效的.

5　总结与展望

从网络结点的构成来看,本文所阐述的故障诊断专家系统的知识网络表示法包含了面向对象的特征.与现象结点层、中间结点层与故障结点层相对应,网络结点可分别用3个基类来表示,3类结点的属性在基类中分别进行定义,具体的网络结点作为基类的实例而拥有各类的数据成员和操作;从推理的实现过程(即结点间的消息传递)来看,此知识网络表示法还具有分布式特性.在知识网络的机器学习方面,本文在介绍网络结点判决函数的相关概念及其数学特征的基础上,提出了结点判决函数的机器学习算法,实验证明是有效的.由于此种知识网络的结构与一般的知识网络不同,所以,知识网络中结点间信息传递的权系数的机器学习仍是一个可进一步研究的课题.

参考文献

1 Luca Console, Claudio Borlo, Alberto casale *et al*. Dealing with uncertainty in a distributed expert system architecture. In: Bouchon-Meunier B, Yager P R, Zadeh LA eds. Proceedings of the 3rd International Conference'90 on Information Processing and Management of Uncertainty in Knowledge-based Systems. Berlin Heidelberg: Springer-Verlag, 1990. 568～577

2 Hayes J E, Michie D, Mikulich L I. Theory of approximate reasoning. Machine Intelligence, 1989,9(2):149～194

3 张立明. 人工神经网络的模型及其应用. 上海:复旦大学出版社,1992.
(Zhang Li-ming. The Model of Artificial Neural Network and It's Application. Shanghai: Fudan University Press, 1992)

4 徐利治等. 函数逼近的理论与方法. 上海:上海科学技术出版社,1983.
(Xu Li-zhi *et al*. The Theories and Methods of Function Approach. Shanghai: Shanghai Science and Technology Press, 1983)

The Design of the Knowledge Network for Fault Diagnosis Expert System

HUANG Bo NI Zhong-kuang GAO Li-ping

(*Department of Computer Science Fudan University Shanghai* 200433)

Abstract In this paper, a structural design of the knowledge network for the identification and classification fault diagnosis model is introduced together with the detailed reasoning algorithm and fault diagnosis algorithm. Reasoning with uncertainty is realized through the message propagation among the nodes, which makes it have the characteristic of a distributed system. In addition, the machine learning algorithm for the judgement function is also presented. This algorithm adopts the package wrapping algorithm to select the border points sets, which makes the speed of curve approach more quickly.

Key words Fault diagnosis expert system, knowledge network, uncertainty pair, judgement function, machine learning.