第14卷 第1期 北京工业大学学报 Vol 14, No.1
1988年3月 JOURNAL OF BEIJING POLYTECHNIC UNIVERSITY Mar., 1988

数据库的自修正、自扩充及其实现

刘 谊 万家增 （北京经济学院）
林文孝 杜 鸿 （大庆乙烯信息中心）
葛明浩 （北京工业大学）

【摘要】结合《部OA管理系统》的核心DB——代码库的自维护，在统计启发式搜索的基础上，讨论以产生式系统为工具的模糊数据基中知识元和元知识的自修正和自扩充，并叙述了系统自学习的实现技巧和过程.

关键词：自学习，自修正自扩充，产生式系统

1. 引言

人们在学习中，常常从若干具体事例中归纳出某个结论来，把这种归纳结论的方法用于计算机中，就成为AI中的自学习[1].

1.1 自学习的发展阶段[2]

1.1.1 直接模拟人脑的办法来设计计算机自学习的功能. 如1943年Mcculloch的二值逻辑模型，1958年Rosenblatt和Minsky的学习模型，Samuel的跳棋程序等.

1.1.2 以符号概念为对象的学习. 如Winston的拱门结构学习，Michalsk的AQLL解法，Quinlan的国际象棋残局程序等.

1.1.3 以元知识为基础的自学习. 如Glauber的推导化学反应定律，Steahl的确定化学组织成份的程序等. 在这一阶段中，另有一种通过类比进行的自学习，这一分支迄今仍有不少困难，如Mitchell的LEX程序.

1.2 自学习系统的主要功能

1.2.1 吸入信息（信息输入）：吸入信息有多种方法[3]，

(1) 用形式化知识表示的方法，它提供知识编辑的功能；

(2) 在原有的知识基础上获取新的知识；

(3) 按照一定的知识表示模型，通过例子学习专家知识；

(4) 直接录入一些元知识.

1.2.2 对系统进行搜索：搜索的目的是为了产生新的规则或新的信息. 目前，AI的搜索方法有启发式搜索，向前搜索，向后搜索，深度（或广度）优先搜索等等.

1.2.3 把可靠的规则和信息添加到规则和数据库中，使之达到系统规则和信息的修正和扩充.

1.2.4 对于已经形成的规则和信息加以利用.

本文于1987年12月28日收到.

2. 数据库的自修正和自扩充

2.1 解题系统的组成和解题的过程

AI中的任何一种解题系统都可以看作是由一个数据库，一组运算符和一个解释程序三个部分所组成的系统．数据库代表所要解决的问题；每个运算符代表了解题的一个步骤，它作用于数据库并改变数据库的状态；而解释程序则包含了解题的策略，即在什么情况下运用哪个运算符去改变数据库的状态以及怎样进行改变．

在解题过程中的每个时刻，数据库都处于一定的状态，数据库的所有可能状态的全体称为状态空间．若把每个状态看成是节点，则整个状态空间就形成了一个有向图．整个有向图可以是不连通的，而在每个连通的子图中，每个弧就代表一个运算符，它把数据库从一个状态引向另一个状态．如果从代表初始状态的那个节点出发，有一条路径通向目标状态，则称此目标状态所代表的问题在当前的初始状态下是可解的．具体给出从初始状态到目标状态通路上的每一条弧（运算符），也就给出了解题过程．

在解题过程中到达过的所有状态的集合称之为搜索空间．若令状态空间为$\mathscr{S}$，搜索空间为$\mathscr{A}$，显然有

$$\mathscr{A}\subset\mathscr{S}$$

在状态空间中，解的路径不一定是唯一的，即使是最短路径也不一定是唯一的．

2.2 产生式系统的组成及其特点

在部机关OA系统中，采用产生式系统作为“OA代码库”自学习的工具．一般来说，产生式系统应具备自返性和传递性，即如果A，B，C属于产生式系统，

则

$$ARA;\ ARB\wedge BRC\rightarrow ARC$$

一个产生式系统也是由三个部分所组成，它包括一组规则，一组数据基和一个解释程序．

2.2.1 规则

规则是指产生式本身，它分成两部分：左部（LHS）和右部（RHS）．通常左部用以表示情况，即在什么条体下产生式被调用；右部表示动作，即产生式被调用后所需执行的动作．在检查产生式左部情况时，常用匹配的方法，即查看当前数据基中是否存在规则左部所指示的情况．如存在则认为匹配成功；否则认为失败．匹配成功时执行右部规定的动作，一般是对数据基中的数据作某种处理，例如“添加”、“置换”、“删除”等等．

2.2.2 数据基

数据基相应于解题系统的数据库，它存放的数据既是构成产生式的基本元素，又是产生式作用的对象．这里所说的数据基是广义的，一般来说，一个数据基就是一个知识元．

2.2.3 解释程序

解释程序负责整个产生式系统的运行，包括规则左部和数据基的匹配，并决定何时停止运行．

相对于其它方式来看，产生式系统具有以下特点：

(1) 具有规范型的格式．每个规则都有如下形式：LHS(信息)$\longrightarrow$RHS（控制动作）．

(2) 匹配过程不产生副作用．匹配失败不影响原数据基的有效性．

(3) 按照不同的用处，可把知识划分成若干类别并存放在数据基中．每类知识又可划分

成若干个知识元，由规则来指明它们之间的相互关系.

(4) 知识元（数据基中的数据），元知识（数据基的使用规则）高阶元知识（使用规则的规则）都可以模块化，并使数据基和规则基具有可扩性和可修改性.

(5) 运行过程采用“数据驱动”方式，控制流是不可见的，因此一个产生式只有通过调用和修改数据库，才能实现对其它产生式的影响.

(6) 机器具有可读性，由机器识别产生式，并对语法、语义进行检查并回答用户提出的问题.

由此可见，产生式系统适用于那些知识元之间或元知识之间相互独立、松散型的系统. 部机关OA代码库”正属于这种情况，因此利用产生式系统建立代码库的自学习是完全适宜的.

2.3 自学习中的统计启发搜索

产生式系统的搜索过程有两种推理方法. 其一是向前推理，如图1所示.

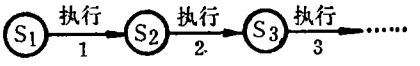

图 1

向前推理的基本思想是把某产生式的左部与数据基中的数据匹配，若匹配成功，则执行产生式右部的动作. 这种动作可能是向数据基增加数据或规则，也可能是修改数据基中的数据、规则，或其它的处理. 前一个产生式执行的结果将使后一个产生式受到激发，依此类推，可完成数据基的修正和扩充.

另一种是向后推理，如图2所示.

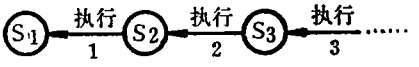

图 2

向后推理的基本思想是查看目标是否为 S_1，如果是 S_1 则执行成功，否则需查看数据基的规则，看有没有规则使得执行 S_1 产生 S_2，若有这样的规则存在，则认为搜索成功，以下依此类推.

比较上述两种推理方法，不难发现，向前推理既适用于对数据（知识）进行推理，又适用于控制，因此比较容易实现数据基中的规则和数据的修正和扩充. 在《部机关OA代码库》的自学习中，选择的就是向前推理. 一般来说，向后推理只适用于数据的搜索，而不适用于控制，因此这种方法常用于专家系统，如“中医诊断”系统等等.

在搜索过程中，往往因为数据基中有多组规则从而发生冲突. 为此，我们在《部机关公文管理系统代码库》的自学习中，利用统计的方法进行启发式搜索. 即在搜索过程中，当一个节点展开时，可以得到这个节点的一组后继节点，这些节点依照它们的地位和内在联系，各有不同的估计值，由估计值就可以确定下一次将展开哪个节点. 这种借助统计学原理去确定搜索的方向，在带有随机性的搜索过程中，使用尤为方便.

下面采用SPRT法[4]，分析在状态空间 $\mathcal{F}$ 上进行启发式搜索的情况.

设 $x_1 x_2 x_3 \cdots\cdots x_n$ 相互独立，并假设 H_0：$\mu=\mu_0$；H_1：$\mu=\mu_1$，则和式

$$S_n=\sum_{i=1}^{n}\ln\frac{f(x_j,\ \mu_1)}{f(x_i,\ \mu_0)}\qquad(n\geqslant 1)\qquad(1)$$

当 $-b\leqslant S_n\leqslant a$ 时，继续进行搜索.

当 $S_n\leqslant -b$ 时，接受假设H_1（搜索失败）

当 $S_n\geqslant a$ 时，接受假设H_0（搜索成功）

其中，a，b 是预先给定的常数 $0<a<b<\infty$

在(1)式中，我们假设 $f(x_i,\ \mu_1)$ 和 $f(x_i,\ \mu_0)$ 都服从正态分布：

$$f(x_1,\ \mu_0)\sim N(\mu_0,\ \delta^2)\qquad i=1\cdots n$$

$$f(x_i,\ \mu_1)\sim N(\mu_1,\ \delta^2)\qquad {}_i=1\cdots n$$

那么

$$f(x_i,\ \mu_0)=\frac{1}{\sqrt{2\pi}\delta}\ e^{-\frac{1}{2}\left(\frac{x-\mu_0}{\delta}\right)^2}\qquad(2)$$

$$f(x_i,\ \mu_1)=\frac{1}{\sqrt{2\pi}\delta}\ e^{-\frac{1}{2}\left(\frac{x-\mu_1}{\sigma}\right)^2}\qquad(3)$$

把 (2)，(3) 式代入 (1) 式中，则有：

$$S_n=\sum_{i=1}^{n}\ln\frac{f(x_i,\ \mu_1)}{f(x_i,\ \mu_0)}$$

$$=\sum_{i=1}^{n}\ln\ e^{\frac{(x_i-\mu_0)^2-(x_i-\mu_1)^2}{2\sigma^2}}$$

$$=\sum_{i=1}^{n}\frac{1}{2\delta^2}[(x_i-\mu_0)^2-(x_i-\mu_1)^2]$$

$$=(2(\mu_1-\mu_2)\sum_{i=1}^{n}x_i+n(\mu_0^2-\mu_1^2))/2\delta^2\qquad(4)$$

由前面假设条件，当 $S_n\leqslant -b$ 时，接受H_0，由 (4) 式得

$$\frac{\left(2(\mu_1-\mu_2)\sum_{i=1}^{n}x_i+n(\mu_0^2-\mu_1^2)\right)}{2\sigma^2}\leqslant -b$$

经整理得

$$\sum_{i=1}^{n}x_i\leqslant\frac{n(\mu_0+\mu_1)}{2}-\frac{b\delta^2}{\mu_1-\mu_0}$$

当 $Sn\geqslant a$ 时，接受假设H_1（即拒绝假设H_0），由 (4) 式得：

$$\left[2(\mu_1-\mu_0)\sum_{i=1}^{n}x_i+n(\mu_0^2-\mu_1^2)\right]\Big/2\delta^2\ \geqslant a$$

整理得：

$$\sum_{i=1}^{n}x_i\geqslant\frac{a\delta^2}{\mu_1-\mu_0}+\frac{n(\mu_0+\mu_1)}{2}$$

统计推断就是这样通过从母体中选用样本，并在这些样本的基础上推断结果，以确定搜索方向.

由于统计中存在着“小概率”错误. 即有可能犯第Ⅰ类或第Ⅱ类错误，这些错误会导致搜索“误入歧途”，甚至造成整个系统的失败. 因此必须作一个限制：当搜索若干步时，还不能得到结论（搜索成功或失败），就应该回过头来做其它方向的搜索，以免走到“死胡同”中去.

假设搜索空间构成的有向图G是连续的（可连通的），则这个图的基础图（Fundamental GRAPh）必然含有一棵生成树T，T包含了G的全部节点，而且T也连通. 对树T的搜索可以实现对G的搜索. 因此，在讨论G的搜索问题时可以先把它看作是一个树T（G的生成树T）.

设 $T=\{V, E\}$，且存在 V'和V''，$V'\subset V$；$V''\subset V$. 并有$V'\cap V''=\phi$；$V'\cup V''=V$. 若V'是已经展开的节点集，V''是尚未展开的节点集，则可以从根节点开始，逐步扩大已经展开的节点集，直至得到我们所期望的结果为止。

如图3所示，设根结点为v，$v_1, v_2\cdots\cdots v_n$是v的子树中的根节点，设$f(v_i)$ $i=1, 2\cdots\cdots n$是v_i的估价函数，我们对v的每个子节点v_i，$i=1\cdots\cdots n$的估计函数进行总体抽样，得出v的每个子节点是接受假设H_0，还是拒绝假设H_0. 并在接受假设H_0的子节点中，找出可能犯错误最小的那个节点，作为进一步展开的对象. 如果搜索已经超过了预先规定的步数（说明搜索可能走向了歧途），这时就应退回到以前曾展开的节点上去，向其它的方向上搜索. 依此类推，总能够在较大的概率上查找到所期望的目标.

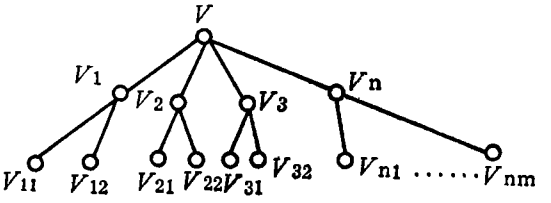

图 3

关于统计启发搜索流图如图4所示.

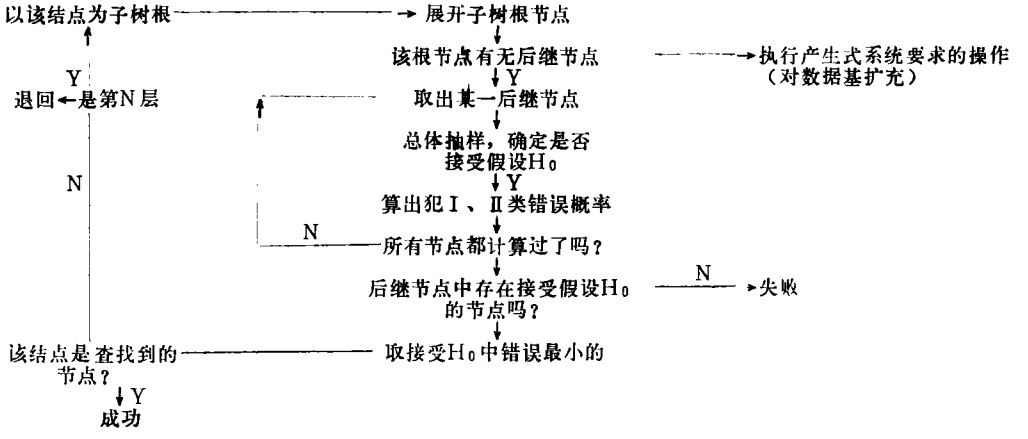

图 4

2.4 模糊信息的匹配

产生式的左部 LHS 与数据基的匹配往往是不精确的，这就涉及对数据基的数据进行模糊查找的问题．为此，需要采用模糊数据库作为数据基．相对于精确数据库 PRDB 而言，模糊数据库 FRDB 至少应包含一个等价关系：

设$\widetilde{S}_j$：$b_j \times b_j \to [0, 1]$

若 $a, b, c \in b_j$，则$\widetilde{S}_j$ 满足

(1) $\widetilde{S}_j(a, a)=1$ （自返性）；

(2) $\widetilde{S}_j(a, b)=\widetilde{S}_j(b, a)$（对称性）；

(3) $S_j(a, c) \geqslant \max(\min S_j(a,b), S_j(b,c))$（传递性）．

通常 PRDB 的值域是原子的，而 FRDB 的值域却不一定是原子的，在某些情况下，FRDB 的值域可以继续划分，这是区分 PRDB 和 FRDB 的重要标志之一．

模糊关系数据库 FRDB 必然存在笛卡尔集上的一组关系：

$$\mathscr{F}=A_1 \times A_2 \times \cdots \times A_m$$

这种关系正是描述和定义《部机关OA代码库》的基础，我们把它分成三个层次．

2.4.1 设 a 是关系 $A_1 \times A_2 \times \cdots \times A_m$ 中的一个成员，则代码的任一组数据关系都包含在$\mathscr{F}$中间，记作 $a=(a_{11}, a_{12}, \cdots, a_{1m})$；$a_{ij} \in D_j$．此时，FRDB 就变成了 PRDB．

2.4.2 设 $B_1, B_2, \cdots, B_m$ 是关系的 m 个属性，则可以建立属性关系表$\widetilde{R}(B_1, B_2, \cdots, B_m)$．

2.4.3 建立数据关系和属性关系的模块化结构，形成数据基的模式．

在 FRDB 中进行数据与 LHR 的匹配，可以利用相似关系进行．只要在代码库中建立一张相似表，就可以对 FRDB 中的数据进行添加、修改和删除等工作，以实现数据基的修正和扩充，从而达到自学习功能的实现．

3. 结束语

在《部机关 OA 系统》中，代码库是其它几个独立数据库的先导库，因此，代码库自学习功能的实现非常重要，代码库能否保持最丰富、最新鲜的数据是其它几个数据库，乃至整个系统正常开发和运行的关键．由于政治体制改革的不断深入，组织机构的不断变化，致使数据唯一性（一致性）的问题变得尤为突出．因此，对核心数据库建立自动数据和规则的扩充和修改，以保证系统能够在较长的时期内活跃在用户之间，具有十分重要的意义．如同其它数据库一样，自学习数据库亦存在数据规范化问题，这是一个值得重视的课题．

参考资料

[1] 冯天瑾．智能机器与人．北京：科学出版社，1983：129

[2] 陆汝钤．人工智能发展概况．计算机工程与应用，1987；(4)：6～7

[3] 何志均，俞瑞钊，童学军．专家系统工具以及 ZDEST—1．计算机工程与应用，1987：(4)：24

[4] 张 钹，张 铃．统计启发式搜索方法．计算机学报，10 (6)：330

[5] 张晏青．Fuzzy 关系数据库的初探．计算机研究与发展，24(8)：57

The Self-defend and Self-correct of DB and Its Pratice

Liu Yi et al.

[Abstract]In combination with studying the central DB for"Management System of OA Document of Minister" and on the basis of heuristic search of statistics, this paper discusses how to self-defend and self-correct the knowledge unit and the meta-knowledge in Fuzzy data base which takes production system as their tool, and describes the self-study skill and procedure of the system.

Key words: self-study, self-defend self-correct, production system