

# 规则向量投影算法 ——一种归纳机器学习方法\*

李绍成

(苏州职工业余大学电子工程系 苏州 215004)

**摘要** 为了支持在事实不完全或不充分环境中的有效推理,作者提出了一种归纳机器学习方法,并设计了一个规则向量投影算法,使用本文介绍的算法可对原始知识实行归纳,生成含一系列全新分类概念和推理路径的网络知识库,基于该知识库的机器推理系统,在作出诊断决策时所需事实量可大为减少,因此在信息量不足的情况下仍能具有很高的推理性能。

**关键词** 专家系统,机器学习,归纳学习,知识库,分类。

## 1 AI 系统归纳机器学习

归纳机器学习是人工智能研究的热点之一,加州大学 R. Clymer 开发了一个面向对象的业务评估模型 OpEM,其中使用了模糊事实,定义了模糊对象,对系统的规则树进行归纳,修改了规则的适配强度,优化了系统的有效性<sup>[1]</sup>;Rutgers 大学 Haym Hirsh 使用背景知识,以更适合归纳学习的形式重新表示了训练数据,改善了决策树和神经网络归纳学习的效果。归纳学习的目标是要得到一个有标签的训练数据集,并形成能预测未来数据标签的分类,归纳学习的成功,训练数据必须以一种形式编码,以重新组织内在的规律,这是归纳学习的关键,但大多归纳学习都是假定了提供的数据已具有一种适宜学习的形式。<sup>[2]</sup>

## 2 规则向量投影算法 RVPA

规则向量投影算法 RVPA(rule vector projection algorithm)是作者为 QDD 系统<sup>[3]</sup>设计并实现的一种归纳机器学习方法,为了支持在事实不充分的环境中进行有效推理,需要对缺乏现成类别标签的原始规则样本空间进行分类;另一方面,随着机器学习的进展,原始规则样本空间会不断变化,也迫使机器去自动实现这种分类,RVPA 被设计成一种非监督的机器学习,它能工作于无模式类别的原始规则样本空间,重新组织彼此独立的原始知识,归纳出内在规律,最大限度地压缩了决策所需的信息量,生成结构良好的分类网络知识,同时也支持这个空间自动地跟踪原始规则样本空间中的任何变化。

RVPA 共分 2 个主要部分:投影和联网,投影又分向一维坐标投影和向二维坐标投影。

\* 作者李绍成,1947年生,讲师,主要研究领域为人工智能应用。

本文通讯联系人:李绍成,苏州 215004,苏州职工业余大学电子工程系

本文 1996-01-16 收到修改稿

联网又分一级网络假设联网、二级网络假设联网和原始规则库联网。

## 2.1 隐层网络生成算法

隐层网络生成算法包括一维坐标投影算法和二维坐标投影算法。

### 2.1.1 一维坐标投影算法

一维坐标投影算法把原始规则向量向特征空间的一维坐标投影,以生成一级网络假设的算法。

设:  $H=[h_1, h_2, \dots, h_m]^T$  为原始假设样本集,  $C=[c_1, c_2, \dots, c_n]^T$  为特征样本集, 则原始规则空间可以表示为

$$H = \Sigma \cdot C \quad (1)$$

其中  $\Sigma$  为  $m \times n$  原始特征矩阵。

若令  $\Sigma^+$  为  $\Sigma$  的伪逆阵, 则

$$C = \Sigma^+ \cdot H \quad (2)$$

$\Sigma^+$  为  $n \times m$  网络规则的假设矩阵。

对伪逆矩阵  $\Sigma^+$  的求解即一维坐标投影算法, 算法的 prolog 定义如下:

```
classify; - or_rule(., ., DNO, COND), diss(DNO, [DCHR]), classify1(DCHR, COND), fail.
```

算法对原始规则库中每一条规则进行逐条投影, 即把 DNO 号代表的假设写入 COND 特征集中每一个特征相应坐标暂存器 diss\_subset 中, classify1 便是做具体写入工作的谓词, 定义如下:

```
classify1(., []).
classify1(DCHR, [BNO|BR]); - diss_into(DCHR, [BNO]), classify1(DCHR, BR).
diss_into(DCHR, COND); - diss_subset(DISEASES, COND), not(find_str(DCHR, DISEASES)),
  retract(diss_subset(DISEASES, COND)), assert(diss_subset([DCHR|DISEASES], COND)), !.
diss_into(., COND); - diss_subset(., COND), !.
diss_into(DCHR, COND); - assert(diss_subset([DCHR], COND)).
```

对投影写入的定义包括3种情况: ①坐标暂存器已建立, 且其中还无待写入的假设名, 则写入该新假设名。②坐标暂存器已建立, 但其中已有待投入的假设名, 不重复写入。③坐标暂存器还未建立, 生成一个暂存器, 并写入第1个假设名。

一维坐标投影结果是进一步生成一级网络规则及网络假设的基础。

### 2.1.2 一级隐层网络生成算法

所谓一级隐层网络生成算法即建立知识领域名与一级网络假设的弧的算法, 定义如下:

```
classify; - diss_subset(DISEASES, [BNO]), create_network1(DISEASES, [BNO]), fail.
```

算法把规则向量的一维投影结果逐条转换成网络假设和网络规则, 直到所有的投影结果都转换结束为止, create\_network1 是直接建网的谓词, 定义如下:

```
create_network1([DISEASE], COND); - concat("probable", DISEASE, P_DISEASE),
  diss(DNO, [P_DISEASE]), getdnr(1, RNO), assert(rule(RNO, 1, DNO, COND)), !.
create_network1([DISEASE], COND); - concat("probable ", DISEASE, P_DISEASE),
  getdnr(1, DNO), assert(diss(DNO, [P_DISEASE])),
  getdnr(1, RNO), assert(rule(RNO, 1, DNO, COND)), !.
create_network1(DISEASES, COND); -
  diss(DNO, DISEASES), getdnr(1, RNO), assert(rule(RNO, 1, DNO, COND)), !.
create_network1(DISEASES, COND); -
  getdnr(1, DNO), assert(diss(DNO, DISEASES)),
  getdnr(1, RNO), assert(rule(RNO, 1, DNO, COND)), !.
```

建网就是具体生成网络假设和网络规则,生成的网络假设分2种情况:①单集假设时,加上“probable”字头,以区别于原始知识;②非单集假设时,就直接生成假设和规则。

### 2.1.3 二维坐标投影算法

若一维坐标投影形成的一维网络假设带“probable”字头,说明在原始规则中有部分规则与样本集其他规则正交.这是一个理想的情况,即确定假设的条件中含有独特的特征,所以一旦确认这个特征,就可以确定这个假设.但大多数网络假设形成的是一个子集,有的甚至还较大.二维坐标投影算法就是要归纳出特征空间中每2个坐标能确定的假设子集.若把一维坐标投影算法看成是从特征空间到假设空间的一种单变换,那么二维(乃至  $n$  维)坐标的投影算法可以看成是从特征空间到假设空间的一种复变换.设原始规则向量为:

$$R_0 = H_{1i}(C_{ij}) \quad (3)$$

其中,  $H_{1i}$  为原始假设样本集中第  $i$  个单集;  $C_{ij}$  为  $H_{1i}$  假设成立时所需确认的特征子集,属于特征样本集中的第  $j$  个子集.那么,一级网络规则就可表达为:

$$R_1 = H'_{si}(C_{1j}) \quad (4)$$

其中,  $C_{1j}$  为特征样本集中第  $j$  个单集;  $H'_{si}$  为  $C_{1j}$  特征确认时所包含的假设子集,属于一级网络假设集中的第  $i$  个子集.那么,二级网络规则就可表达为:

$$R_2 = H''_{si}(C_{2j}) \quad (5)$$

其中,  $C_{2j}$  为二元特征集,是特征样本集中第  $j$  个二元集;  $H''_{si}$  为  $C_{2j}$  特征确认时所包含的假设子集,属于二级网络假设集中的第  $i$  个子集.

一维坐标投影算法是求  $H'_{si}$  和  $C_{1j}$  的过程,由于  $C_{1j}$  为单集,故算法主要集中在求  $H'_{si}$ . 二维坐标投影算法求  $H''_{si}$  和  $C_{2j}$  的算法都较复杂.不难发现把一维坐标算法中的单项假设也定义成集合是形式上的简化,定义如下:

```
classify: - or_rule(-, -, DNO, COND), diss(DNO, [DCHR]), classify2(DCHR, COND), fail.
```

和定义一维坐标投影算法类似,算法对原始规则库中的每一条规则进行逐条投影,即把 DNO 号代表的假设写入 COND 特征集中每2个特征组成的坐标暂存器 `diss_subset` 中, `classify2` 是做写入工作的谓词,定义如下:

```
classify2(-, [- | []]).
```

```
classify2(DCHR, [DH1 | [DH2 | DR]]): - classify21(DCHR, [DH1 | [DH2 | DR]]), classify2(DCHR, [DH2 | DR]), !.
```

```
classify21(-, [- | []]).
```

```
classify21(DCHR, [DH1 | [DH2 | DR]]): - diss_into(DCHR, [DH1, DH2]), classify21(DCHR, [DH1 | DR]), !.
```

从 COND 特征集中每次选2个特征, `classify2` 和 `classify21` 的递归定义简化了  $C_2^2$  的算法. 规则向量的二维坐标投影结果是进一步生成二级网络规则及网络假设的基础.

### 2.1.4 二级隐层网络生成算法

和一级网络生成算法类似,二级网络生成算法是建立一级网络节点和二级网络节点之间弧的算法,定义如下:

```
classify: - diss_subset(DISEASES, [H1, H2]), create_network2(DISEASES, [H1, H2]), fail.
```

算法把规则向量的二维坐标投影结果逐条转换成网络假设和网络规则,直到所有的投影结果都转换结束为止.由于算法中规定不再重复生成相同的假设,所以实际上二级网络节点将覆盖部分一级网络节点;另外,二级网络生成算法只在一级网络节点中寻找可以用2个坐标表达的假设,剩余的假设将由表层网络生成算法完成与原始规则空间联网.

create-network2是直接建网的谓词,定义如下:

```

create-network2([DISEASE],COND):- concat("probable ",DISEASE,P-DISEASE),
diss(DNO,[P-DISEASE]), create-network21(DNO,COND,COND),!.
create-network2([DISEASE],COND):- concat("probable ",DISEASE,P-DISEASE),
getdnr(1,DNO),assert(diss(DNO,[P-DISEASE])), create-network21(DNO,COND,COND),!.
create-network2(DISEASES,COND):- diss(DNO,DISEASES),create-network21(DNO,COND,COND),!.
create-network2(DISEASES,COND):- getdnr(1,DNO),
assert(diss(DNO,DISEASES)), create-network21(DNO,COND,COND),!.

```

create-network2谓词的结构同 create-network1谓词,但 create-network1只生成1条规则,而 create-network2要生成2条规则,因为二级网络假设含有2个特征坐标.

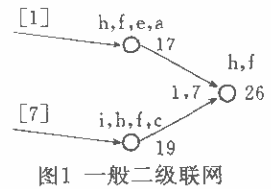
create-network21是生成二级网络弧的谓词,定义如下:

```

create-network21(-,[],-).
create-network21(DNO,[BNO|BR],COND):- rule(-,-,DNO1,[BNO]),
DNO1=DNO, create-network21(DNO,BR,COND).
create-network21(DNO,[BNO|BR],COND):- rule(-,-,DNO1,[BNO]), getnrn(1,RNO1),
assert(rule(RNO1,DNO1,DNO,COND)), create-network21(DNO,BR,COND).

```

create-network21一次生成2条弧的情况可用图1来表示,节点26是二级网络假设,节点17和19是2个一级网络假设,它们通过 BNO 变量自动寻找. create-network21还规定了一个例外的情况是节点本身不能自联,这种情况发生在被二级网络覆盖的原一级网络的节点上,因此这种例外情况下只生成一条网络弧.



### 2.2 表层网络生成算法

表层网络生成算法完成一、二级网络假设中的终节点与原始规则空间的联网工作以及完成一级网络假设中非终节点中需要增补的联网,算法的定义如下:

```

classify:-
rule(-,DNO,-,-), rule(-,-,DNO,-),
make_leaves_point(DNO),fail.
classify:-
terminal_goal(-,DNO), not(or_rule(-,-,DNO,-)),
diss(DNO,P-DISEASES),
set-terminal_rule(DNO,P-DISEASES),fail.

```

算法的第1部分是网络终节点与原始规则空间的联网,terminal\_goal 是检查网络终节点的谓词,即取出网络每一个终节点的假设集,用 set-terminal\_rule 谓词把它们与原始规则空间联网,定义如下:

```

set-terminal_rule(-,[]):- !.
set-terminal_rule(DNO1,[P-DISEASE|R]):- remove_probable(P-DISEASE,DISEASE),
diss(DNO2,[DISEASE]),or_rule(-,-,DNO2,COND),
getnrn(1,RNO),assert(rule(RNO,DNO1,DNO2,COND)),!, set-terminal_rule(DNO1,R).
remove_probable(PDISEASE,DISEASE):- fronttoken(PDISEASE,"probable",D),frontchar(D,-,DISEASE),!.
remove_probable(PDISEASE,DISEASE):- DISEASE=PDISEASE.

```

由于网络假设中存在着单集标识“probable”,联网时首先应该除去这种标识,才能与原

始假设样本集中的假设取得统一,remove\_probable 就是检查并除去“probable”标识的谓词.算法的第2部分是一级网络非终节点增补联网,用 make\_leaves\_point 谓词实现检查和增补,定义如下:

```
make_leaves_point(DNO):-assert(diss_list([]),sum_diss(DNO,SUM_DISSL),diss(DNO,DISEASES),
    sublist_str(DISEASES,SUM_DISSL,DISS),set_rule(DNO,DISS).
```

make\_leaves\_point 实现增补联网是先统计出 DNO 号节点下属各节点的假设 SUM\_DISSL,利用 sublist\_str 谓词找出网络假设 DISEASES 与下属假设 SUM\_DISSL 之间的漏缺项 DISS,利用 set\_rule 谓词做具体增补这种漏缺的联网工作.

统计下属节点假设的谓词 sum\_diss 定义如下:

```
sum_diss(DNO,-):-rule(_,DNO,DNO1,-),diss(DNO1,DISEASES),sum_dlist(DISEASES),fail.
sum_diss(-,SUM_DISSL):-diss_list(SUM_DISSL),retract(diss_list(SUM_DISSL)).
sum_dlist([]).
sum_dlist([P_DH|DR]):-remove_probable(P_DH,DH),diss_list(SUM_DISSL),
    retract(diss_list(SUM_DISSL)),assert(diss_list([DH|SUM_DISSL])),sum_dlist(DR).
```

sum\_diss 是对 DNO 节点的每一个下属节点 DMO1的假设集 DISEASES 进行统计,最后取出结果返回 SUM\_DISSL 变量.统计工作具体由 sum\_dlist 谓词做,为了与原始假设样本集取得一致,要消除可能存在的“probable”标识.

在下面将给出的 set\_rule 谓词定义中,对空间假设集的检测有两层含意:①满足递归定义的需要;②在上述 sublist\_str 谓词找出的假设漏缺项 DISS 为空时,实际的增补不发生.set\_rule 的定义如下:

```
set_rule(-,[]):-!.
set_rule(DNO,[DH|DR]):-getrnr(1,RNO),diss(DNO1,[DH]),or_rule(-,-,DNO1,COND),
    assert(rule(RNO,DNO,DNO1,COND)),set_rule(DNO,DR).
```

### 2.3 RVPA 基本思想

RVPA 是在特征空间里把用特征表示的假设向量  $R_0$  作向特征坐标投影的运算.在向一维特征坐标投影时实际上是形成了一个假设空间,得到了在假设空间中的每一个特征的表达  $R_1$ ,在向二维特征坐标投影时,又得到了在假设空间的每2个特征的表达  $R_2$ .由于特征可以看成是一种约束,当这个约束不断增加的时候,得到的假设子集中的元素将不断减少.当得到的假设子集为单集的时候,实际上假设空间又变为特征空间,不过这时候表达假设单集的特征集  $C_m$  可能会小于原始规则中的相应表达  $C_mj$ .然而这个小于却代表着一种优化,代表着一种从实例上升到概念后又回到实例后的提高.这个过程可以用图2表示.

RVPA 的意义不只是产生了这种优化的表达,更重要的是建立了一系列的网络隐层概念.这些隐层概念不仅可用于机器在事实不充分环境中的推理,而且还可用于系统的解释:作为用户在给机器输入知识之后,回过头来向机器学习怎样使用这些知识的理论.

## 3 结 论

根据式(1) $H = \Sigma \cdot C$ ,一个实验库原始规则样本集就可具体表示为如式(6),式中  $\Sigma$  为  $10 \times 10$  原始特征矩阵.从式中可知假设  $h_2$  为 [aaa] 成立的条件是特征集  $[c_1, c_3, c_4, c_{10}]$  中的所有元素都要获得确认.

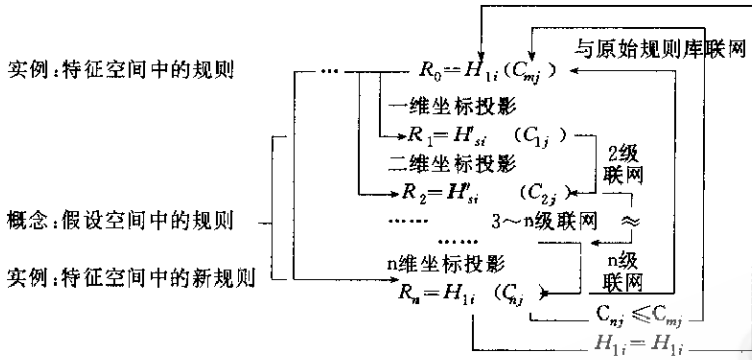


图2 RVPA图解

$$\begin{matrix} h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \\ h_{10} \\ h_{11} \end{matrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \\ c_{10} \end{matrix} \quad (6)$$

基于式(6)的原始知识, RVPA 自动生成了分类规则网络, 其中含对原假设  $h_2$  的机器推论——分类新概念:  $h_2$  为 [probable aaa] 成立的条件是特征集  $[c_3, c_4]$  中所有元素都要获得确认. 图3为使用该机器推论的一次模拟诊断过程.

QDD 专家系统机器学习的宗旨是要自动生成一个新的高性能分类规则网络, 由该网络支持系统推理机能在事实不充分的环境中作出有效推理. 分类规则网络之所以能具有这样的性能, 是基于 RVPA 对原始规则库在全局范围内所作的全部思考, 形成了一系列原始规则库所没有的新的分类概念知识, 这是 RVPA 的主要贡献.

RVPA 自动生成一个含大量新概念的分类规则网络, 其实质是解决了从事物的原始特征中找出能在群体中识别个体的全部最小特征集, 由于后者比前者信息量大为减少, 故在解决诊断分类问题时, 就不必要求确认全部特征. 就象善素描者能用很少的线条勾画出人物肖像一样, 这种能力的具备对于人类领域专家来说需要以年计的培养时间, 但 RVPA 把这个过程缩短到了瞬间.

显然, RVPA 的应用可支持广阔的领域, 比如可在汉字多面的特征中找人们容易接受的特征组合作为候选汉字输入法, 就可以为发明者大幅度提高工作效率等.

```

Enter Goal: domain
Is it true that 7-cond: unknow(事实不充分)
Is it true that 1-cond: unknow(事实不充分)
Is it true that 3-cond: yes
Is it true that 4-cond: yes
Is it true that 10-cond: unknow(事实不充分)
I think they are:[probable aaa,] with 100% certainty.
  
```

ok  
sure  
why  
more

图3 推理机在事实不充分的环境中使用 RVPA推论, 仍能找到用户目标aaa

参考文献

- 1 Wildberger A Martin. AI & Simulation. SIMULATION,1995,64(6).
- 2 Hirsh Haym. Using background knowledge to improve inductive learning. IEEE Expert, 1994,9(5):3.
- 3 李绍成. 一个基于分类的专家系统——饲料质量及禽病专家系统的研制[硕士论文]. 西安交通大学,1991.

**RULE VECTOR PROJECTION ALGORITHM——  
AN INDUCTIVE MACHINE LEARNING METHOD**

Li Shaocheng

(Department of Electronic Engineering Worker & Karder's Spare Time College of Suzhou Suzhou 215004)

**Abstract** To support the effective reasoning in the circumstances of incomplete and insufficient facts, the author presented an inductive machine learning method, and designed a rule vector projection algorithm. With the method implemented by the algorithm the primitive knowledge are processed by inductive approach, created a network knowledge base full of new classification concepts and paths for reasoning. Based on the knowledge base the quantity of facts needed in a diagnostic decision making by a machine inference system can sharply be reduced, thus a high performance can still be achieved by the system, despite of short information.

**Key words** Expert system, machine learning, inductive learning, knowledge base, classification.