提问的技巧、算法及其应用

邵维忠 H. S. Soon*

(北京大学计算机系,北京 100871) (新加坡国立大学系统科学院)

THE ART OF QUERY, ITS ALGORITHMS AND APPLICATIONS

Shao Weizhong

(*Department of Computer Science & Technology, Peking University, Beijing* 100871)

H. S. Soon

(*Institute of Systems Science National University of Singapore*)

Abstract The art of query is a kind of human intelligence. The approach described aims to make the computer to simulate this human intelligence in this paper. A set of strategies and algorithms is given to raise the efficiency of query. The art of query can be used to improve the mechanism of expert systems and some other systems by reducing their questions to users as far as possible.

摘要 巧妙的提问是人类的智能之一.本文讨论如何让机器模仿人的这种智能,给出一系列提高提问效率的策略和算法.这些技巧可以改善一些咨询系统,特别是专家系统的提问机制,使系统在为用户服务时,向用户提出的问题尽可能的简练贴切,从而降低系统的提问开销.

§0. 引 言

提问技巧(The Art of Query)在组合数学中可归属于"Ulam 问题",[1]也可从一种"问答猜谜"游戏中得到启发:出谜的人心里想好一种东西,比如说,一种动物,让人猜他想的是什么.猜谜人则通过提问题来了解这种动物的特性,比如:"它会飞吗?""它吃肉吗?"等等,根据出谜人的回答判断是什么动物.如果猜谜人具有良好的提问技巧,则只需要不多的几次提问便能猜中谜底;反之则需要提很多问题才能猜中.

在知识工程领域,许多专家系统在向用户提供咨询服务时,其工作方式也类似于上述游

* 本文1991年6月收到.作者**邵维忠**,,副教授,主要研究领域为软件工程,人工智能.**H. S. Soon**,助研,1991年硕士毕业于美国波士顿大学,主要研究领域为人工智能.

戏.比如医疗诊断系统给病人诊断时,先问病人有些什么症状,诸如是否发烧、是否呕吐等等;然后根据各种症状的有无判断这个病人患了什么病.在模式识别系统中也有类似的过程:系统先索取一些特征信息,然后通过匹配找出这些特征所对应的模式.在这类系统中,提问方法的优劣直接影响着系统的性能和效率.不久前,新加坡国立大学系统科学院承担了一个飞行导航仪故障诊断专家系统的设计任务.为了优化该系统的提问机制,笔者对提问技巧问题进行了研究,设计并验证了一系列算法与策略.这些成果已为该系统所采用,[2]并在一些国际会议上作过初步介绍.[3,4]本文将进行更完整的讨论.

我们把上面提到的动物特征、病人症状等都称作"特征"(feature),把动物名称、疾病名称都称作"目标"(target—gool).每个目标对应一组与其他目标不同的特征值.假如一个系统中总共有 m 个目标,即 $G_1,G_2,\cdots,G_m$;所有目标都是通过 n 个特征 $F_1,F_2,\cdots,F_n$ 来描述的.那么全体目标的已知信息可以用一个 m 行 n 列的矩阵来表示,矩阵的元素 a_{ij} 表示目标 G_i 在特征 F_j 上的取值,如图 1.

	F1	F2	……	Fn
G1:	a11	a12	……	a1n
G2:	a21	a22	……	a2n
…	…	…	…	…
…	…	…	…	…
Gm:	am1	am2	……	amn

图1

在一个实用的系统中,目标和特征的数量一般都不会太小.特征的数量一方面要能保证任何两个目标的特征值都不会重复,另一方面,设计者收集的特征往往带有很大的冗余性.

人脑中的知识一般都是冗余的.但人在搜寻一个目标时并不需要把用来区别全体目标的每一项特征都提问一遍,通常只需要提不太多的几个问题.因为人能根据自己的知识和经验舍弃那些无关紧要的问题,只提问那些关键性的问题,并且知道先问哪些问题更有利于及早地发现目标.本文要研究的问题是,如何让计算机模仿人类的这种智能,从而尽量减少系统对用户的提问.

我们将首先考虑一种最简单的系统模型,它满足下述条件:1.每个目标出现的概率都是相等的;2.用户回答每一个问题所付出的代价都是相等的;3.系统提出的每一个问题都能得到明确的回答;4.目标的每个特征只有 1 和 0 两种值.1 表示"是",0 表示"不是".

第 1 节—第 3 节将针对这种简单的系统模型展开讨论.讨论的问题包括:关键特征的抽取,提问次序的安排,以及提问—判断过程的设计策略.第 4 节—第 6 节将讨论一些比较复杂的情况,包括:系统中的各个目标以不同的频率出现.例如,各种疾病有不同的发病率;各个问题的"难度"可能不同,即用户在回答不同的问题时要付出不同的代价;对于某些问题,用户可能回答"不知道",系统要有相应的补充措施.最后将分析一些更复杂的情况,把特征值从布尔类型推广到其他类型,包括整数、实数、模糊逻辑表示及区间表示.

§1. 关键特征的抽取

定义:设 S 是用来区分全体目标的特征集合,S′是 S 的一个子集.假如 S′中的特征仍然能区分系统中的全部目标,而且其中每一个特征都是不可忽略的(即,忽略任何一个特征都会引起某些目标的混淆),则 S′叫做全体目标的关键特征集合,简称 KFS.

KFS 可能有多个,例如,图 2 中目标 $G_1,G_2,\cdots,G_6$ 由特征 $F_1,F_2,\cdots,F_5$ 区分.我们可以找出两个 KFS,即 $S'=\{F_1,F_3,F_4\}$ 和 $S''=\{F_2,F_3,F_4,F_5\}$.本节要讨论的是如何寻找一个尽

可能小的 KFS.

KFS 中包含的特征数量 k 的大小与目标的数量 m 有关,同时也在很大程度上与各个目标的取值分布有关.在最理想的分布下 $k=\lceil \log_2 m \rceil$,如图 3(a).最不理想的分布可使 k 达到 m−1,如图 3(b).

	F1	F2	F3	F4	F5
G1:	1	0	1	1	0
G2:	0	1	1	0	1
G3:	1	0	0	1	1
G4:	1	0	0	0	1
G5:	1	0	1	0	1
G6:	0	0	0	1	1

图2

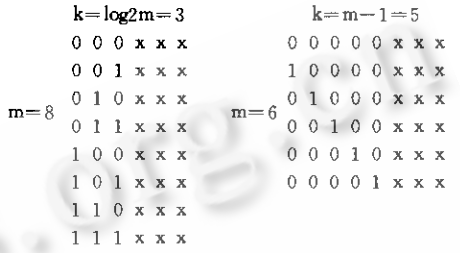

图3

在实际中,如果初步收集的特征数量 n 明显地大于 $\log_2 m$,则在大多数情况下能够找到的元素个数远小于 n 的 KFS.现在讨论如何寻找一个尽可能小的 KFS.如果想求最佳解,则需要考察 n 个特征中各种可能成为 KFS 的子集合,其数量是:

$C_n^k+C_n^{k+1}+\cdots+C_n^{m-1}$　(其中 $k=\lceil \log_2 m \rceil$)

这将使计算量达到"组合爆炸"的程度.

所以,想求最佳解的企图在现实中是行不通的.这里要介绍的是一个启发式算法,它可以得到相当不错的结果.其基本思想是,在初始的特征集合中逐个丢掉那些取值分布不好而又确实可以忽略的特征,直到任何特征都不可忽略为止.衡量一个特征好坏的标准是:这个特征被提问并获得解答之后,能否明显地缩小目标的搜索范围.

假如特征 F_i 使得 m 个目标中有 x 个目标在这个特征上取值为 1,m−x 个目标取值为 0.F_i 被提问之后,如果回答是"1",那么以后只要考虑 x 个取值为 1 的目标.反之,若回答是"0",则只要考虑 m−x 个取值为 0 的目标.根据前面中的假设,每个目标出现的概率是相等的,所以当提问 F_i 时,回答"1"和"0"的概率分别为 x/m 和(m−x)/m.因此,在 F_i 被提问之后平均搜索范围是:

$$x\cdot x/m+(m-x)\cdot(m-x)/m=(2x^2-2mx+m^2)/m$$

这是一个在 x=m/2 时达到最小值的二次函数.它表明,取值为 1 的目标数越少越好.我们用"特征的价值"这个术语来描述一个特征的优劣.第 j 个特征 F_j 的价值 W_j 定义为:

$$W_j=\frac{m}{2}-\left|\frac{m}{2}-\sum_{i=1}^{m}a_{ij}\right| \tag{1}$$

其中 a_{ij} 是第 i 个目标在第 j 个特征上的取值,即图 1 的矩阵中第 i 行 j 列元素.

在图 2 的例子中,各个特征的价值分别为:$W_1=2$,$W_2=1$,$W_3=3$,$W_4=3$,$W_5=1$.这表明 F_2 和 F_5 是最差的特征.我们看到,图 2 中第 2 列元素几乎全是 0,而第 5 列几乎全是 1.这样的列对于造成各行之间的差别没有太大的贡献,因此应该优先被忽略.

现在让我们给出求 KFS 的算法:

算法 1:求全体目标的关键特征集合

①按照公式(1)计算每个特征的价值.

②在未做任何标记的特征中挑选一个价值最小的特征(设它是 F_j).

③删除 F_j,即把所有的目标对应于 F_j 的特征值都置为 0.

④检查,如果有某些目标混淆(即剩余的特征值完全相同),则恢复 F_j,并标记 F_j 是"不

可删除的”(即:F_j 是关键特征).如果每个目标都不混淆,则标记 F_j“已删除”.

⑤如果每个特征都已做标记,则算法停止,否则转向②.

可以看出,这个算法的计算复杂性是 $n.m^2$ 级的.为了考验算法 1 的质量,我们曾利用它来抽取汉字集合的关键特征.字库中共有 6763 个汉字(6763 个目标).每个汉字用一个 16×16 的点阵表示(256 个特征).用算法 1 得到的 KFS 只包括 34 个特征(点).也就是说,在这个应用实例中约有 86%的特征是可忽略的.图 4 是部分汉字原始形状和只剩下 34 个关键特征时的形状对照.

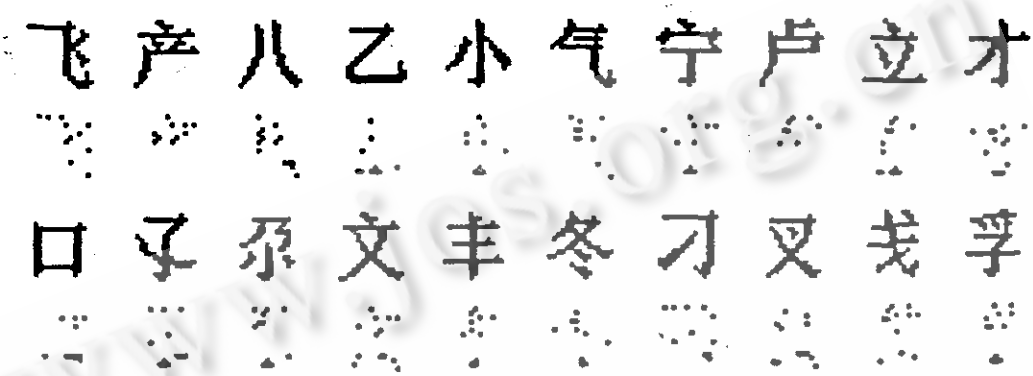

图 4

§2. 提问序列

有了 KFS 之后,系统在任何一次咨询服务中提问的次数将不超过 KFS 的特征数量,而且在许多情况下并不需要把 KFS 中的每个特征都提问一遍.如图 5 所示,G_1～G_5 的 KFS 是{F_a,F_b,F_c},在提问了 F_a 和 F_b 之后如果回答是(0,0),则已可确定目标是 G_2,不必再提问 F_c 了.

每个目标所需要的提问次数有多有少,反映系统效率的标准是平均提问次数越少越好,它取决于提问的次序.例如,在图 5 的例子中,若按 F_a,F_b,F_c 的次序提问,平均提问次数是 2.7.如果按 F_a,F_c,F_b 的次序提问,则平均提问次数是 2.4.一个好的提问序列,应该使较多的目标能在前几次提问中确定,较少的目标要到最后几次提问确定,从而使平均提问次数减少.

	Fa	Fb	Fc
G1:	1	1	1
G2:	0	0	1
G3:	1	1	0
G4:	0	1	0
G5:	0	1	1

图5

本节介绍寻找较好的提问序列的算法.其基本思想如下:对于 KFS 中的每一个关键特征做--次尝试性的删除(即删除后还要恢复),看当它被删除时会引起多少个目标混淆,此数目称作对应于这个特征的“辨别率”.选取辨别率最小的特征排在序列的最后.这种安排的效果是,使数量最少的目标需要提问到最后一个问题才能鉴别.以后的步骤是,把已经选去排队的特征真正地删除,重新计算每个剩余特征的辨别率,选出下一个特征(按从后向前的次序)排进序列,直到每一个特征都排到序列中.F_j 的辨别率 R_j 在面临每次挑选时是不同的,它可以这样定义:

R_j＝当 F_j 和已排入序列的特征都删除时混淆的目标总数　　(2)

算法 2：求提问序列

①集合 S 中包括 KFS 的全部关键特征.

②计算 S 中每个特征的辨别率.

③挑选辨别率最小的特征(设它是 F_j)；把 F_j 按从后向前的次序排入序列.

④从 S 中删除 F_j，即：使每个目标忽略关于 F_j 的取值.

⑤如果 S 为空则停止，否则转向②.

算法 2 的计算复杂性也是 $n.m^2$ 级的. 在图 5 的例子中，第一次选择时 F_a，F_b，F_c 的辨别率分别为 4，2，4，选到了 F_b. 第二次选择时 F_a 和 F_c 的辨别率都是 5，选到(比如说)F_a. 由此得到的提问序列是 F_c，F_a，F_b，其平均提问次数是 2.4.

应该指出，尽管这个算法从每一步来看似乎都是最合算的，但是整体上仍然不能保证得出最佳提问序列. 不过，它的实际效果还是不错的. 对于前边所说的 6763 个汉字的 34 个关键特征，用算法 2 安排的提问序列平均提问次数只有 14.67. 相比之下，用几种随机产生的次序来提问，平均提问次数大多在 18.5 以上.

§ 3. 提问和判断

当用户知道了一个目标的特征而不知道它是什么时，系统通过提问来了解这些特征，从而作出判断，告诉用户这个目标是什么. 系统的提问和判断可以按下述方法进行：提问之前，把它所知道的全部目标都放在一个集合 G 中. 每当提出一个问题并获得回答之后，就把 G 中与回答不符合的目标排除，接着提下一个问题. 当 G 中只剩下一个目标时，它就是用户要寻找的目标. 如果在某一步 G 成为空集合，则表明用户所要寻找的目标不在系统的知识范围之内.

提问可以按照一种预先安排的次序进行，例如，按照 § 2 中求出的提问序列逐个地发问. 这种方法可称为“静态提问机制”. 它的优点是每一步都能立即知道该提什么问题，因此提问速度很快. 缺点是，不能根据前几次提问所获得的信息为剩余的问题安排一个更好的提问次序，所以提问次数不能随机应变地进一步减少.

另一种方法是采用“动态的提问机制”. 即每次提问并且排除了与回答不符合的目标之后，就以剩余的目标和未曾提问的关键特征为背景重新计算，选出下一个最值得提的问题. 这种方法在很多情况下能够进一步减少提问次数. 但是当系统目标很多时，提出每个问题之前所进行的计算时间可能使用户难以接受.

一种折中的方法是“半动态”的提问机制：前几个问题按照预先安排的固定次序提问. 到剩余的目标已经不多时改为动态的方法.

§ 4. 概率不相等的情况

在前边几节中，我们假设每个目标都以相同的概率出现. 但是在实际中更多的情况是各个目标出现的概率不相等. 例如在医院，有些疾病发病率很高，有些病则极为罕见. 本节将在这种前提下重新考虑前几节中的问题并改进算法.

定义：一个目标的出现概率是它在一批统计资料中出现的次数与全体目标出现的总次

数之比.

目标 $G_1,G_2,\cdots,G_m$ 的出现概率用 $P_1,P_2,\cdots,P_m$ 表示. 在考虑到概率这个因素后,对于算法 1 的修改是:改变计算特征价值的公式. 即把公式(1)改为:

$$W_j=0.5-|0.5-\sum_{i=1}^{m}P_ia_{ij}| \tag{3}$$

公式(3)的推导与公式(1)类似:假如在 F_j 上取值为 1 的目标概率之和为 P. 当 F_j 被提问时,回答为 1 的概率为 P,回答为 0 的概率为 1−P. 回答为 1 时,下一次只要处理在 F_j 上取值为 1 的目标,它们出现的概率之和为 P;回答为 0 时,下一次要处理的目标出现的概率和为 1−P. 因此,下一次的平均处理量为 $P^2+(1-p)^2=2p^2-2p+1$. 当 P=0.5 时上式取最小值. 即 P 越接近 0.5 越好,越接近 0 或者 1 这两个端点就越差. 公式(3)恰好反映这一规律,其中的 $\sum_{i=1}^{m}P_ia_{ij}$ 就是 P.

在各个目标的出现概率不相等时,衡量一个提问序列质量好坏的标准不再是"平均提问次数",而是"加权平均提问次数". 即 $\tilde{t}=\sum_{i=1}^{m}P_it_i$,其中 t_i 是第 i 个目标在这个序列中所需要的提问次数. 对算法 2 的修改,主要是修改"辨别率"的定义. 现在 F_j 的辨别率 R_j 定义如下:

R_j=当 F_j 和已排入序列的特征都删除时全部混淆目标的概率总和. (4)

§5. 问题的花费

到现在为止,我们只是把减少提问次数作为唯一的追求目标. 但是在实际应用中,仅仅考虑问题的数量是不够的. 因为不同的问题在回答时所付出的代价往往大不相同. 比如,医疗诊断系统提出的问题,有的回答起来象聊天一样容易,有的则需要进行昂贵、痛苦的检查.

我们用"问题的花费"作为回答这个问题所要付出的人力、物力、时间、风险等各类代价的综合描述,用一个大于零的实数来表示. 一个特征的花费,也就是它所对应的问题的花费. 特征 $F_1,F_2,\cdots,F_n$ 的花费分别用 $C_1,C_2,\cdots,C_n$ 表示.

现在考虑对前几节讨论的内容进行补充和改进. 首先,在求关键特征集合时,重要的目的不再是让集合中包括尽可能少的关键特征,而是要让集合中的各个关键特征的花费总和尽可能少. 从这个要求出发,在决定各特征的取舍时,要以它的价值和花费的比值大小作为取舍依据. 特征 F_j 的价值/花费比按下述公式计算:

$$W_j/C_j=(0.5-|0.5-\sum_{i=1}^{m}P_ia_{ij}|)/C_j \tag{5}$$

对算法 1 的修改是:按公式(5)计算各个特征的价值/花费比,逐次挑选价值/花费比最小的特征,尝试把它删除,直到任何特征都不能删除为止. 其次,对求提问序列的算法 2 也要作修改. 各个关键特征的辨别率仍按公式(2)计算(当目标的概率不等时则按公式(4)计算),但是要挑选使得 R_j/C_j 值最小的关键特征排到提问序列的最后. 其效果是:作用较大而花费较小的问题将较早地提问;前几个问题的花费之和比较小,而且及早结束提问的可能性比较大.

对于算法 1 和算法 2 所作的修改,都是为了达到这样的目的:使各个目标的"加权平均提问花费"尽可能小. 加权平均提问花费的定义是:

$$\tilde{C}=\frac{1}{m}\cdot\sum_{i=1}^{m}(P_i\sum_{j=1}^{k_i}C_j) \tag{6}$$

其中 k_i 是第 i 个目标需要的提问次数(i=1,2,…,m).

§6. 当提问得不到回答时

系统提出的问题用户可能难于回答，宁愿系统提一些别的问题. 人与人之间的问答，遇到这种情况时，提问者可以利用自己头脑中的冗余知识提出另外一个(或几个)适当的问题，来代替不能回答的问题.

系统的知识，开始时一般也是冗余的，但在寻找 KFS 时却被缩减到了很简练的程度. 在按照 KFD 中的各个关键特征提问时，遇到无法回答的问题，就可能导致最终无法确定要寻找的目标. 因此在寻找 KFS 时，不应该把 KFS 以外的那些特征信息真正地丢弃，而应要作为一种后备知识保留下来，以便在必要时使用.

具体的处理方法，一种是预先对 KFS 中的每一个关键特征都找出它的一组后备特征(即：用这一组特征来代替这个关键特征仍能使全体目标彼此都不混淆). 在提问时，问题得不到回答它就用后备特征代替. 另一种方法是：在 KFS 中的问题都提完之后，把剩下的无法区别的目标作为一个新的目标集合，根据剩余的(即未曾提问的)特征对这些目标的分布情况，重新计算剩余特征的价值，然后逐个挑选价值较高的特征提问.

§7. 各种类型的特征值

在前几节的讨论中，我们只考虑了布尔类型的特征值. 每个目标的每一项特征只有"1"或"0"两种值. 但是在实际中许多事物的特征不能简单地用布尔逻辑来表示. 例如要描述一种动物是否凶猛，就很难只用 1 和 0 这两个数来表示，而需要采用模糊逻辑的表示方法，使特征值取 0 和 1 之间的一个实数.[5,6]更一般地，系统中目标的特征可涉及整数、实数、字符串等各种数据类型. 本节讨论一般类型特征值的处理方法. 首先讨论较简单的情况：特征值是一个单独的量，随后讨论一种比较复杂的情况：用数值区间表示的特征值.

7.1 确定的一般类型特征值

这种类型的特征用一个单独的(确定的)数值来表示. 它对应各个目标的值可能有多种，不再象布尔类型的特征那样只有 0 和 1 两种. 这意味着，我们的讨论需要从"二值特征"推广到"多值特征".

这个推广对前几节所讨论的内容的主要影响是，在抽取关键特征时，特征价值的计算方法需要更具有一般性. 现在我们在各个目标概率不等的情况下开展讨论. 假如特征 F_j 的值对全体目标而言最多可能有 J 种，它们是：$V_1,V_2,\cdots,V_J$. 我们用 P_s 表示在全体目标中特征值为 V_s 的那些目标的概率之和，$(1\leqslant s\leqslant J)$. 这样，$P_1+P_2+\cdots+P_J=1$.

在关于 F_j 的问题被提问时，回答是 V_s 的概率是 P_s，而若回答为 P_s，则下一步只要考虑特征 F_j 的值为 V_s 的那些目标，由此可以算出，F_j 被提问后处理范围平均可缩小为 $\sum_{s=1}^{J}P_s^2$. 根据这种分析，我们定义 F_j 的价值如下：

$$W_j=1-\sum_{s=1}^{J}P_s^2 \qquad (j=1,2,\cdots,n) \tag{7}$$

在 $P_1+P_2+\cdots+P_J=1$ 的约束条件下，当 $P_1,P_2,\cdots,P_J$ 都接近于它们的平均值 1/J 时 $\sum_{s=1}^{J}P_s^2$ 较小，即 W_j 较大.

公式(7)也适应于布尔类型的特征(看作 J=2 的特殊情况). 用它代替 §4 中的公式(3)，在效果上是一样的(虽然求出的数值不同). 当系统中有些特征是非布尔类型时，应该对每一项特征都用公式(7)去计算它们的价值，以便彼此比较.

对于求关键特征的算法 1 的修改，只需要用公式(7)代替公式(1)来计算特征的价值. 求提问序列的算法 2 不需要修改.

7.2 用数值区间表示的特征值

在许多情况下，特征值不能用一个确定的数来表示，而需用一个数值区间作为特征值. 例如，在医疗诊断系统中，我们不好把脑炎的体温特征定为摄氏 40 度，因为当患者的体温为 39.8 度时也未必不是脑炎. 妥善的办法应该是，用一个数值区间，比如 39 度至 41 度来描述脑炎的体温特征.

在现实世界中有下列几种情况需要用一个区间作为目标的特征值：

1. 目标是一个群体，而不是一个特殊的个体. 例如，上边提到的“脑炎”是指所有的脑炎患者，每个病人的体温难免有些差别.

2. 目标虽然是一个特殊的个体，但它可能随着时间、环境的不同而发生一些变化. 例如一个人的体重可以在一定范围内变化.

3. 系统设计者和用户对一个目标的特征值的认识和估价存在差别. 凡是需要用模糊逻辑的方法描述的特征尤其如此，例如一种病的疼痛程度，系统用 0.6 表示，用户也可能用 0.7 表示.

上述几种情况，最好都用一个数值区间作为特征值. 一般可以用一个有序对表示这个区间的下界和上界.

用区间作为特征的值时，系统寻找目标的方法有些变化，从而要求在抽取关键特征时以新的方法计算特征的价值. 图 6 表示一个动物识别系统中几种动物的体重范围，可以看到各种动物的体重区间有某些部分是重叠的，系统在识别动物时必须考虑到这种情况. 比如，当用户告诉系统，某一动物的体重是 440 公斤，系统将看到 440 这个数落在三种动物(即虎、熊和牛)的特征区间内，因此要在集合中保留这三个目标以作进一步辨别，而把其它的目标排除.

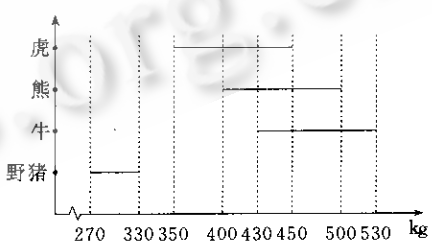

图6

现在讨论计算特征价值的方法. 在图 6 从每种动物的体重区间的下界和上界向 x 轴方向作投影，这把全部目标的特征值分布区间(270—530)划分成一些小的区间段. 这样的划分，使一个目标的特征值区间可能被分成几个小段，例如，熊的体重区间 400—500 被分为 400—430，430—450，450—500 三段. 也使每个小段可以位于零个到多个目标的特征值区间内，例如，(430,450)这个小段位于虎、熊、牛三种动物的体重区间内，而(270,330)不符合任何目标.

设特征 F_j 的值域被划分为 J 个小段，用 $A_1, A_2, \cdots, A_J$ 表示. 我们知道每个目标的出现概率，当一个目标的特征值区间被分成若干小段时，这些小段也划分了这个目标的概率，例如，熊的出现概率假定是 0.04，在 430—450 这个小段内，按长度比例划分的概率是 $0.04\times(450-430)/(500-400)=0.008$，(更精确的划分可以由系统设计者按实际分布情况指定各个小段上的概率). 对 $A_1, A_2, \cdots, A_J$ 中的任意小段 $A_s(1\leqslant s\leqslant J)$，可以把它对应的所有目标(例如，430—450 这个小段对应的虎、熊和牛)在这个小段上划分的概率累加起来，得到 P_s. P_s 就是在提出问题之后，回答的数值属于 A_s 的概率. 这样我们得到 $P_1, P_2, \cdots, P_J$，它们分别对应于 $A_1, A_2, \cdots, A_J$.

现在可以得到与公式 7 形式上相同的价值计算公式：

$$W_j=\sum_{s=1}^{j}P_s^2 \qquad (j=1,2,\cdots,n) \tag{8}$$

公式(8)的推导过程也和公式(7)相同，不再重叙. 不同之处只是 P_s 的定义和计算有所区别. 值得高兴的是，公式(8)和(7)不但形式上相同，而且求出的特征价值也是彼此可比较的，因此当系统中包含不同类型的特征时，可以用统一的原则和算法选取关键特征及安排提问序列.

结束语：本文介绍了寻找关键特征集合及优化提问序论的策略和算法，并从简单的系统模型推广到目标概率及提问花费不等的情况以及非布尔类型的特征值. 这些技巧可以明显的减少系统的提问开销.

本文主要是针对提问技巧这个问题本身展开讨论的. 在实际系统中提问常常是和推理交叉进行的. 提问技巧在系统中可以作为单纯改善提问的辅助措施，也可以在设计时考虑把一部分推理规则中包含的知识转化为目标的特征，相对集中地采用提问—匹配式的搜索. 如何把提问与推理有机地结合，尚有待进一步探讨.

参考文献

1 J. Czyzowicz, A. Pelc and D. Mundici, Solution of Ulam's Problem on Binary Search with Two Lies. Journal of Combinatorial Theory, Series A 49, 1988, 384—388.

2 H. H. Teh and A. H. Tan., Connectionist Expert Systems — A Neural—Logic Models' Approach, In Proceedings, Inter—faculty Neuronet Seminar, National University of Singapore, 1989.

3 W. Z. Shao and H. S. Soon, Intelligent Query Mechanism for Expert Systems, IEEE 14th Annual International Computer Software & Applications Conference, Oct. 1990, 599—604, Chicago.

4 W. Z. Shao and H. S. Soon, The Art of Qeury and Its Applications in Knowledge Engineering, IT Works '90, Singapore, June 1990.

5 L. A. Zadek, Fuzzy Set, Information and Control, 8(1965), 338—353.

6 汪培庄，模糊集合及其应用，上海科技出版社，1983.