概率逻辑公式集分解的合并聚类算法*

张晨东 陈火旺 刘凤岐

(国防科技大学计算机系 长沙 410073)

摘要 为使概率逻辑的不确定性推理方法能应用于较大规模的知识库,本文基于一个实际专家系统知识库的开发经验,在概率逻辑公式一致性区间的一般算法基础上,为概率逻辑公式集的分解设计了一种合并聚类算法.对于不同背景的概率逻辑知识库,只要公式集具有一定的分层结构性质,该算法就能保证Dantzig-Wolfe分解的联合计算模型适用于概率逻辑推理.测试结果表明,该算法对于数10个变量和子句的实例可收到很好的效果.

关键词 概率逻辑,一致性,分解算法,聚类,专家系统,概率推理.

中图分类号 TP18

基于知识的系统常面临知识不完全或不确定的情况,因此,出现了多种不确定性知识的表示和推理方法.Nilsson提出概率逻辑(Probabilistic Logic[1])的不确定性知识表示和处理方法具有多方面的优点.[2~4]它使得从领域专家获取知识的过程具有更大的灵活性,而把困难的知识库一致性维护及不完全信息下的推理工作交给系统完成,这样就对系统功能提出了更高的要求.到目前为止,概率逻辑公式集的一致性问题仍然没有令人满意的通用算法.文献[1]虽提出了概率逻辑知识库的一致性维护的线性规划模型,但模型的规模随着知识库中原子公式的数量呈指数式增长,使得该模型只适应非常小的问题而不能达到实用.为此许多研究者进行了深入探索并取得了一定成果.[3~7]在这些研究进展中,有的只进行理论分析而未给出实际可行的算法,有的虽给出了可行算法,又对问题结构进行了严格的限制.这说明寻求通用的可行算法也许是不可能的.但已有成果却给出了一个十分明确的研究方向,即寻求不同的分解计算方法以降低计算复杂性.本文基于一个专家系统知识库的开发经验,认为概率逻辑计算模型在许多实际背景下有着自然的分层—梯阶性质,即可用分解和简化的结构来表示具有复杂结构的问题,使得求解原问题的计算复杂性降低.[8~11]对于这种结构的问题可进行分解—协调计算(称联合解算,即采用由粗到细,从全局到局部的逐层深入解决问题的方法).这一方法是可行的,即不超出现有的计算设备的实际计算能力,且算法的复杂性也被控制在可接受的程度.这一方法保持了Nilsson模型的线性规划特征,与已有的模

* 本文研究得到国家自然科学基金和国家863高科技项目基金资助.作者张晨东,1960年生,博士生,主要研究领域为人工智能.陈火旺,1936年生,教授,博士导师,主要研究领域为软件自动化,计算机科学理论,人工智能.刘凤岐,1938年生,教授,主要研究领域为计算机软件,人工智能.

本文通讯联系人:张晨东,长沙 410073,国防科技大学计算机系

本文1996-06-10收到修改稿

型相比[5]，实际处理问题的规模可成倍增加，原模型中计算规模随原子公式数量呈指数式增长的趋势得到有效控制. 为了使这一方法得到推广，使之应用于其它领域具有相似结构特点的概率逻辑知识库的开发和维护过程中，需要解决一般的概率逻辑公式集分解问题. 本文设计了概率逻辑公式集的合并聚类分解算法，使得按算法进行分解后的结构能适应联合解算方法的要求. 经过具有一定规模的模拟数据计算测试，算法具有很好的适用性.

1 概率逻辑公式集一致性赋值区间计算及联合解算方法

本文论及的概率逻辑公式由一般的命题逻辑公式加上概率赋值函数组成，概率赋值函数对每一个命题公式赋予唯一的一个概率函数值. 即若 a 为命题逻辑公式，P 为概率赋值函数，则 $P(a)=p$ 为概率逻辑公式(其中 $p\in[0,1]$). Nilsson 进而指出，若对命题公式的概率赋值函数不加限制，则可能出现不一致的情况.[1,2]为此，Nilsson 引入了可能世界语义模型，指出如果能由可能世界构造一个概率空间，且概率测度与每个命题公式的概率赋值函数值一致，那么这些命题公式的概率赋值函数为一致的. 可能世界集取命题公式集所包含的原子公式集的真值赋值的全体. 设命题公式集为 A，A 中的原子公式集为 $atom(A)$，对原子公式集的真值赋值函数集为 $S=\{s|s:atom(A)\to\{0,1\}\}$，集合 S 的基数为 $|S|=2^{|atom(A)|}$. 对每一个原子公式集的真值赋值函数，都存在唯一的一个对命题公式集的真值赋值函数与之对应. 若记对命题公式集的真值赋值函数集为 $T=\{t|t:A\to\{0,1\}\}$，则存在集合 S 与 T 之间的一个对应关系 $U=\{u|u:S\to T\}$. 这样，可以通过 T 上的概率测度来间接地考察 S 上的概率测度. 现需要考察一组对命题公式的概率赋值 $P:A\to[0,1]$是否是一致的，也就是要考察满足如下线性约束的概率测度 $x:T\to[0,1]$是否存在：

$$\Sigma_{t\in T}\,t(a)x(t)=P(a),\ (a\in A);$$
$$\Sigma_{t\in T}\,x(t)=1;$$
$$x(t)\geqslant 0.$$

若要求出某一公式 $a_i\in A$ 的一致性概率赋值区间，可构成如下线性规划模型：

$$\begin{aligned}&\min/\max && \Sigma_{t\in T}\,t(a_i)x(t);\\ &s.t. && \Sigma_{t\in T}\,t(a)x(t)=P(a),\ (a\in A\backslash\{a_i\});\\ & && \Sigma_{t\in T}\,x(t)=1;\\ & && x(t)\geqslant 0.\end{aligned}$$

以上模型实际上是构造一个满足约束的概率空间 $(S,K(S),x)$，或是在概率可测空间 $(S,K(S))$上求一个满足约束的概率测度 x. 这里 $K(S)$为包含集合 S 中全部单点集并由 S 子集形成的 σ 代数，实际问题中 S 总是有穷的，$K(S)$包含了 S 的所有子集. 以上模型中约束式的列数为 $|T|=|S|=2^{|atom(A)|}$. 稍大规模的问题就会遇到计算设备和算法的限制. 而针对分层一梯阶结构的联合解算方法克服了上述模型的限制. 联合解算方法可简述如下：

该方法要求的前提条件是概率逻辑公式集 A 可以划分为若干个允许原子公式相交的子集 $A_1,A_2,\ldots,A_N$，它们满足：

(1) 局部问题计算可行性：$2^{|atom(A_i)|}$的问题规模是实际可以计算的；这保证每个局部问题的线性规划模型中系数矩阵的列数不超过可解的规模.

(2) 全局问题计算可行性：记 $atom(A_{i,j})=atom(A_i)\bigcap atom(A_j)$，若 $atom(A_{i,j})\neq\varnothing$，其中 $i,j=1,2,\ldots,N$，且 $i\neq j$，则 $\Sigma 2^{|atom(A_{i,j})|}$ 的问题规模是实际可计算的. 这保证全局问题的线性规划模型系数矩阵的行数不超过可解的规模.

(3) $A_i\bigcap A_j=\varnothing$，$(i,j=1,2,\ldots,N,i\neq j)$，且 $\bigcup_{i=1}^{N}A_i=A$.

这要求 $A_i(i=1,2,\ldots,N)$ 中包括原公式集中的全部公式.

在以上条件满足的情况下，可以建立一个主问题和若干相对独立的子问题组成的分解计算模型，其基本思想是，对每个划分后的公式集建立一组局部约束形成局部问题，并各自独立(可并行)地求解，只要在保证求解过程满足原子公式相交公式集的局部解在公共的边缘上具有一致性，后者由一组全局约束进行控制. 因此所建立的计算模型采用一个主问题和若干个子问题的联合求解策略.

如果记主问题对应的所有主约束集合的系数矩阵为 M，右端向量为 B，记与公式子集 A_i 所确定的子问题对应的局部约束的系数矩阵为 M_i，右端向量为 B_i，变量列向量为 X_i，则可构成如下联合计算模型：

$$
\begin{array}{llll}
M & (X_1,X_2,\ldots,X_N)^T & = B & (0)\\
M_1 & X_1^T & = B_1 & (1)\\
\cdots\cdots & & \cdots\cdots & \\
 & M_i\ X_i^T & = B_i & (i)\\
 & \cdots\cdots & \cdots\cdots & \\
 & M_N\ X_N^T & = B_N & (N)
\end{array}
$$

在以上模型中，概率逻辑公式集在推理过程中的一致性由各组子问题约束(1)～(N)保证，而各相关子问题之间的一致性则由主问题约束(0)保证，这在形式上构成了大规模线性规划 dantzig-wolfe 分解算法的标准形式. 由于各子问题的规模已得到有效控制，单纯形方法可用于解各个子问题，而主问题是依靠各子问题的生成列的限制线性规划问题，规模也得到了有效控制. 主问题与各子问题可在并行计算环境中有条件地并行求解，以提高计算效率. 具体求解方法见文献[9～11]. 显然，对一般的概率逻辑知识库，若在结构上能进行某种类似的分解操作，则可应用联合求解算法(或改进后的联合求解算法)进行解算. 如何对知识库中的概率逻辑公式集进行分解是本文要解决的主要问题.

2　概率逻辑公式集分解的合并聚类算法

对概率逻辑公式集分解的目的是限制计算模型的规模从而可能用现实的计算设备进行解算. 由于概率逻辑的线性规划模型的规模一方面随公式集中的原子命题的数目呈指数式增长，另一方面随公式集中公式的数目呈线性增长，因此，控制局部计算模块中原子公式的数目是主要的，一般情况下可以不考虑公式的数目. 此外，在对公式集分解时所关心的只是每个公式子集中的公式数目和原子命题数目，而公式的逻辑结构与分解无关，这样可把公式集看作由原子命题的集合构成的集类，类中的元素是每一个公式对应的原子命题集合，分解的算法就是要找一个满足要求的集类的划分，每个划分子类中所有集合的并集构成一个原子命题的集合，称为子类原子集，则每个子类原子集的基数反映了局部问题的计算规模，而不同子类的交集的基数反映了主问题的规模. 在设计具体的分解算法之前，需要首先确定分

解的目标.目标可以分为2个层次来考虑,在基本层次,只考虑可行目标,即只要把公式集分解为若干个子集,使得每个子集不超过计算设备可解算的规模,所有子集的交集也不超出计算设备可解算的规模.在更高的层次上,可以进一步考虑当有多种分解的结果时,选择最优的分解结果使计算量最小.

设概率逻辑公式集 $Pr(A)$ 中的命题逻辑公式集为 $A=A_1,A_2,\ldots,A_N$,记相应的原子公式的集类为 $atom(A)=atom(A_1),atom(A_2),\ldots,atom(A_N)$,分解算法是要求一个公式集类 $ATOM(A)$ 的分解形式:

$$ATOM(A)=ATOM(A^1)\bigcup ATOM(A^2)\bigcup\ldots\bigcup ATOM(A^M);$$

其中 $ATOM(A^i)\bigcap ATOM(A^j)=\varnothing,(i,j=1,2,\ldots,M,i\neq j)$;

再记 $U(i)=\bigcup_{j\in A^i} atom(j),i=1,2,\ldots,M$,$U(i)$ 为第 i 个子类中集合的并集;使满足如下可行目标:

(1) $\max(2^{|U(i)|},i=1,2,\ldots,M)\leqslant MC$;

(2) $\Sigma_{i,j\in 1,2,\ldots,M,i\neq j,U(i)\cap U(j)\neq\varnothing}2^{|U(i)\cap U(j)|}\leqslant ML$;

式中 MC 与 ML 分别为计算设备所能承受的最大的线性规划系数矩阵的列数与行数,显然,列数约束只与每个子问题有关,而行数约束与每对相交子问题所构成的主问题有关.若追求分解的最优目标,则可增加优化指标如下:

(3) $\min(\Sigma_{i=1}^{M}2^{|U(i)|})$;

(4) $\min(\Sigma_{i,j\in 1,2,\ldots,M,i\neq j,U(i)\cap U(j)\neq\varnothing}2^{|U(i)\cap U(j)|})$;

优化指标(3)使得所有局部子问题系数矩阵的总列数最少,而指标(4)使主问题的行数最少,2个优化指标合在一起就使分解线性规划模型的总规模最小.

本文从解决一般问题的实际应用出发,只考虑可行目标,即算法只要找到公式集的一种划分使条件(1)和(2)满足则终止,并不继续寻找满足(3)(4)的最优划分.本文的分解算法是基于合并聚类方法.[12]该方法的关键是确定子类之间的距离从而给出子类合并标准,本文定义的2个子类之间的距离为相同原子命题数与相异原子命题数之差.即 $S(i,j)=2(|U(i)\bigcap U(j)|)-|U(i)\bigcup U(j)|$.下面是基于合并聚类的分解算法:

第1步:将每个公式视为一个子类,得到公式集的一个划分:

$ATOM(A^1),ATOM(A^2),\ldots,ATOM(A^M)$;

其中 $M=N,A^i=\{\{A_i\}\},U(i)=atom(A_i),i=1,2,\ldots,N$;并置记数器为 M.

第2步:对每两个不同的子类,计算它们之间的距离:

$S(i,j)=2(|U(i)\bigcap U(j)|)-|U(i)\bigcup U(j)|,i,j=1,2,\ldots,M,i\neq j$;

第3步:求出具有最小距离的子类对集合 $ATOM(A^{u_i})$ 和 $ATOM(A^{v_i})$:

$S(u_i,v_i)=\min(S(k,j),k,j=1,2,\ldots,M,k\neq j),i\in I$;

第4步:合并子类 $ATOM(A^{u_k})$ 和 $ATOM(A^{v_k})$,$k\in I$;

记 $H_i=|ATOM(A^{u_i})\bigcup ATOM(A^{v_i})|$,若对所有的 $i\in I$,均有 $2^{Hi}>MC$,则算法结束,无解;否则,选择 $k\in I$ 满足 $2^{Hk}\leqslant MC$ 进行合并:

删除子类 $ATOM(A^{u_k})$ 和 $ATOM(A^{v_k})$,并增加子类 $ATOM(A^t)=ATOM(A^{u_k})\bigcup ATOM(A^{v_k})$;重新排列各子类的序列标号为 $1,2,\ldots,M-1$,并使子类记数器减1,$(M-1)\to M$.

第 5 步:可行标准判断,当前划分若满足:

(1) $\max(2^{|U(i)|}, i=1,2,\ldots,M) \leqslant MC$;

(2) $\Sigma_{i,j\in 1,2,\ldots,M, i\neq j, U(i)\cap U(j)\neq\varnothing} 2^{|U(i)\cap U(j)|} \leqslant ML$;

则已找到可行解,算法结束;否则,计算新的子类与其它子类的距离:

$S(t,j)=2(|U(t)\cap U(j)|)-|U(t)\cup U(j)|, j=1,2,\ldots,M, j\neq t$;转第 3 步.

上述算法的第 1 步把每个公式单独地作为一个子类,这里的前提是构成公式的原子命题集的可能世界数不超过实际可解的线性规划的规模. 第 2～4 步为子类合并,这种合并过程使某些子类对应的可能世界数随原子集基数不断增加,同时使子类个数减少. 在并行计算环境中,合并过程则使某些计算节点上的计算量增加,使所使用的计算节点数减少. 合并过程的结束由第 4、5 步控制. 第 4 步的算法出口表明没有找到可行解,即对公式集的分解失败;第 5 步的算法出口表明已找到可行解,可以用所给出的分解结果利用联合解算模型进行概率逻辑推理. 该算法的时间复杂性上限为 $O(2N^3)$. 只为找可行解,因而算法没有回溯. 算法的可靠性由第 4 步的无解条件和第 5 步的可行标准判定予以保证. 如果要进一步寻找更好的解或最优解,将增加算法的复杂性. 在实际应用中,可考虑计算机辅助与人工分解相结合的工作方式,把问题背景特点充分反映到公式集结构中来. 以下算法实验采用了比文献[5]较宽的计算条件:每个逻辑公式由不超过 5 个原子命题组成;公式集中的原子命题数为 70;公式集中共有 2～8 个主题;每个主题涉及的原子命题数不超过 10;主问题和各子问题的限制规模均为 2^{10};对逻辑公式的结构不附加任何限制. 计算方案由计算机按所设条件随机产生.

表 1　算法求出可行解的百分率

主题数 \ 公式数	20	30	40	50	60	65	70	75	80	85	90	95	100
8	99	99	99	99	99	99	99	99	99	99	99	99	99
9	99	99	99	99	99	99	99	99	99	99	99	95	90
10	99	99	99	99	99	99	99	99	99	99	99	94	90

表 2　聚类得到可行解的平均合并次数

主题数 \ 公式数	20	30	40	50	60	65	70	75	80	85	90	95	100
2	0	3	12	22	31	37	43	48	56	60	69	74	75
3	0	3	13	23	33	38	43	48	53	58	62	69	73
4	0	2	13	23	32	37	42	47	53	56	62	68	72
5	0	2	12	22	30	36	41	45	51	55	60	66	70
6	0	0	8	18	28	32	38	43	48	53	58	63	68
7	0	0	7	17	27	32	37	42	47	52	57	62	67
8	0	0	6	17	26	31	36	41	46	51	57	62	67
9	0	0	5	17	26	31	36	41	46	51	57	62	67
10	0	0	4	17	26	31	36	41	46	51	57	62	67

以上结果是基于有结构限制的初始数据. 可以看出,对有主题划分和主题内原子公式数限制的公式集,即具有分层一递阶结构的概率逻辑知识库,分解算法成功的概率很高,联合

解算模型也很适用;而若针对完全没有结构限制的随机数据,则只能在公式数不超过40的情况下得到部分可行解,而公式数若超过50,则基本上得不到可行解.这样的结论基本符合对实际情况的分析.当公式集本身具有分层—递阶结构,它是可分解的.但能否分解到实际进行计算的程度,还与计算环境与设备的计算能力有关.只有当公式集可分解到满足实际计算规模的限制条件时,这样的分解结果才是有实际意义的.否则,即使公式集具有某种可分解的特性,由于计算环境的限制也会使分解过程以失败结束.对于本文给出的试验数据,由于原子公式集的基数只有几十,目前在一般的计算环境中对线性规划的求解规模为几百到几千个约束或变量.这使得各公式子集中原子数目可达到十几个,因此,若公式的长度再受到文中给出的限制,则算法的成功率一般是比较高的.若把原子集的基数扩大到上百至数百个,且不改变主题个数和计算环境,那么,算法的成功率会明显降低.在与算法成功与否关系最大的3个因素(即原子集基数、主题个数和计算环境)中,计算环境会受到各种客观条件的限制,而原子集基数和主题个数与知识库的结构有关,因此,发掘应用领域背景知识的结构特征是非常重要的,它对公式集的分解及算法设计影响很大.

3 结 论

概率逻辑公式集在具有分层—递阶结构的条件下,应用分解—协调的联合解算方法具有其它方法所没有的优越性.由于许多实际应用问题的领域知识都具有某种可分块的结构特征,因此,联合解算法把概率逻辑从理论研究和实验研究层次向实际应用层次推进了一大步.本文给出合并聚类算法的目的是希望这种基于联合解算方法的概率逻辑计算模型能被推广,以解决不同领域和应用背景的不确定性知识处理问题.今后研究问题有进一步放松联合解算方法的条件、寻找公式集最优分解方案(或满意分解方案)以及提高分解算法的效率.与本文有关的概率逻辑的一些基本问题还可参见文献[13,14].

参考文献

1 Nilsson Nils J. Probabilistic logic. Artificial Intelligence, 1986,**28**(**1**):71～87.

2 Genesareth M, Nilsson N. Logic foundations for AI. Morgan Kaufmann, 1987.

3 Andersen K A. Characterizing consistency in probabilistic logic for a class of Horn clauses. Mathematical Programming, 1994,**66**:257～271.

4 Anderser K A, Hooker J N. Bayesian logic. Decision Support Systems, 1994,**11**:191～210.

5 Kavvadias Dimitris, Papadimitriou Christos H. A linear programming approach to reasoning about probabilities. Annals of Mathematics and AI, 1990,**1**:189～205.

6 Bouchaffra D. A relation between isometrics and the relative consistency concept in probabilistic logic. In: Proceedings of the 13th IMACS World Congress on Computation and Applied Mathematics, Dublin, July 1991. 22～26.

7 Bouchaffra D. Consistent regions in probabilistic logic when using different norms. Proceedings of the Third Workshop on Artificial Intelligence and Statistics, Miami, Fort Lauderdale, FL, January 1991. 5.1～5.5.

8 高文豪.大系统最优化.水利电力出版社,1991.

9 Dantzig G B. Linear programming and extensions. Princeton University Press, 1963.

10 Dantzig G B, Wolfe P. Decomposition principle for linear programming. Oper. Res., 1960,**8**:101～111.

11 张勇传,瞿继恂.组合最优化——计算机算法和复杂性.武汉:华中理工大学出版社,1994.

12 陈尚勤,魏鸿骏.模式识别理论及应用.成都:成都电讯工程学院出版社,1985.

13 Halpern Joseph Y. An analysis of first-order logics of probability. Artificial Intelligence, 1990,46(3):311～350.
14 Nilsson Nils J. Probabilistic logic revisited. Artificial Intelligence, 1993,59(1):39～42.

A CLUSTERING-ALGORITHM FOR DECOMPOSITION OF PROBABILISTIC LOGIC FORMULA SET

ZHANG Chendong　CHEN Huowang　LIU Fenqi

(Department of Computer Science　Naitonal University of Defence Technology　Changsha　410073)

Abstract　In order that the probabilistic logic reasoning under uncertainty can be used for large scale knowledge-base, this paper presents a clustering-algorithm for decomposition of probabilistic logic formula set based on the general consistence assigning algorithm for probabilistic logic and the experience of developing the knowledge-base in an practical expert system. It ensures that the united-model of Dantzig-Wolfe decomposition can be used for probabilistic logic reasoning on probabilistic logic knowledge-base with different background, provided the formula set hold certain hierarchical structure. Experiments show that the algorithm performs successfully on instances with dozens of variables and clauses.

Key words　Probabilistic logic, consistency, decomposition algorithm, clustering, expert system, probabilistic reasoning.

Class number　TP18