计算机科学1999Vol.26№.8

基于时序逻辑的时序数据库中知识发现方法*)

Approaches of Discovering Knowledge in Temporal Databases Based on Temporal Logic

王清毅 范 焱 蔡庆生
(中国科学技术大学计算机系 合肥230027)

Abstract This paper first gives the definitions of temporal database, temporal operator and temporal logic programming, and then discusses the temporal relations between the facts in the temporal databases. Some concepts such as fact sequence, class of temporal pattern are also defined. The approaches of discovering two classes of temporal patterns are described in detail.

Keywords Knowledge discovery in databases, Temporal databases, Temporal operator, Temporal logic programming, Temporal pattern

知识发现是一个众多学科诸如人工智能、机器学习、模式识别、统计学、数据库和知识库、数据可视化等相互交叉、融合所形成的一个新兴的且具有广阔应用前景的领域。目前国际上对知识发现的研究与开发进展很快[1]。众多现实世界(如天气预报、电信、金融)的数据库中数据都是随时间变化的,即数据具有时序性。目前,时序数据库中的知识发现问题正逐渐引起 KDD 研究者的兴趣。本文首先给出时序数据库、时序算子和时序逻辑程序的定义,然后讨论时序数据库中事实间的时序关系,定义了事实序列、时序模式的类等概念,详细描述时序数据库中两类时序模式的发现方法。为讨论方便,本文在同样的意义下不加区别地使用知识和模式这两个术语。

1 相关的概念

考虑和自然数集合 N 同态的时间域 T,一个关系由有限个非时序属性和一个或多个在时间域上变化的时序属性所构成。考虑数据库是由有限个谓词(即将一个元组视为一个谓词)构成的集合 ζ。

定义1 一个时序数据库 ζ 是一个函数,该函数将 ζ 中的每个谓词映射到 T 的一个子集(该子集是这个关系成立的时刻的集合),即 $B=(\zeta\to 2^T)$。

在经典一阶逻辑中加进时序算子,就构成了一阶时序逻辑(FOTL)。时序算子如下:

定义2 现在 always 算子□。□A 为真,如果 A 在将来每个时刻为真。现在 sometimes 算子◇。◇A 为真,如果 A 在将来某个时刻为真。next 算子○。○A 现在为真,如果 A 在下一个时刻为真。过去 always 算子■。■A 为真,如果 A 在过去每个时刻为真。过去 sometimes 算子◆。◆A 为真,如果 A 在过去某个时刻为真。previous 算子●。●A 为真,如果 A 在前一个时刻为真。until 算子▷。A▷B 为真,如果 A 一直为真,直到 B 出现,此时 A 为假。该算子是一个二元算子。

约定□,◇,○的运算优先级高于▷及∧,∨,→,↔,而▷的优先级又高于∧,∨,→,↔。我们仍然沿用一阶逻辑中的项、原子公式、文字、子句以及基项、基原子公式、基文字和基子句的概念。一个时序文字就是一个以时序算子为前缀的文字。如 next-文字就是以有限个 next 算子为前缀的文字,○○…○A=○ⁿA 表示在 n 个下一时刻 A 为真。一个时序规则具有 BODY→HEAD 的形式,其中 BODY 和 HEAD 都是时序文字的合取式,合取式中各个时序文字用逗号分开。

一个时序逻辑程序 TLP 是一个时序规则的集合。这里我们采用 Datalog$_{1s}$[2]来表示时序规则。在 Datalog$_{1s}$中,每个谓词至多包含一个时序参数。另外我们允许谓词的否定出现在时序规则的头和体中。

*)本文得到国家自然科学基金资助。王清毅 博士生,主要研究领域为机器学习、知识发现。范焱 博士生,主要研究领域为机器学习、知识发现。蔡庆生 教授、博士生导师,主要研究领域为人工智能、机器学习和知识发现。

允许时序变元之间的比较出现在时序规则的体中。将(…((0+1)+1…+1)(加n个1)记为n,将(…(T+1)+1…+1)(加n个1)记为T+n,例如,存在这样一条规则:对银行有欺诈行为的顾客将永远不允许在该银行存款。用时序算子可以表示为:

◇SAVING(bank, customer)∧SWINDLE(bank, customer)→□¬SAVING(bank, customer)。

在Datalog$_{1S}$中,可以利用谓词中的时序参数来表示顾客的存款时间,即用SAVING(bank,customer,time)来替换SAVING(bank, customer),用SWINDLE(bank, customer, time)来替换SWINDLE(bank, customer)。

2 两个知识发现方法

一般说来,时序数据库中的事实(即谓词的对象为常元)之间的时序关系可以形象地用有向非循环图如图1所示。

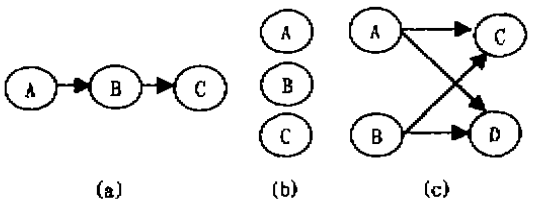

图1 时序数据库中的事实间的时序关系

图1(a)描述了事实A,B,C之间的时序串行关系,即A在B之前发生,B在C之前发生;(b)描述了事实A,B,C之间的时序并行关系,即A,B,C的发生没有时间顺序上的关系约束;在(c)中,事实C、D只有在事实A,B发生以后才会发生,但在A和B之间,C和D之间则没有时间顺序上的约束关系。下面我们来形式化地定义时序数据库中的几个概念。

定义3 给定一个基本事实类F,一个时间域T,一条事实$fact$是一个序偶(f,t),其中$f\in F$,$t\in T$。一个事实序列S是一个三元矢(T_1,T^n,s),其中T_1是开始时间,T^n是结束时间,s是一个事实的有序序列,即$s=(f_1,t_1),(f_2,t_2),(f_3,t_3),\cdots,(f_n,t_n)$,其中$f_i\in F$,$T_1\leqslant t_i\leqslant T^n$,$i=1,2,3,\cdots,n$,$t_i\leqslant t_{i+1}$,$i=1,2,3,\cdots,n-1$。

定义4 一个时序模式的类$P=(V,\leqslant,g)$,其中V是结点的集合,$\leqslant$是关于V的偏序,g是一个映射$g:V\to F$,该映射将每个结点和一条事实相关联。$g(V)$中的事实以$\leqslant$所确定的时间顺序发生。

定义5 称一个时序模式的类$(V,\leqslant,g)$出现在一个事实序列$s=(f_1,t_1),(f_2,t_2),(f_3,t_3),\cdots,(f_n,t_n)$,如果存在一个满射$h:V\to\{1,2,\cdots,n\}$,使得对所有的$v\in V$,$g(v)=e_{h(v)}$且对所有的$v,w\in V$,有$v\leqslant w$,$h(v)<h(w)$,称一个时序模式是频繁的,如果这个时序模式在事实序列中发生的次数大于给定的个频度阈值c。

从时序逻辑公式的角度来看,一个时序模式是一个基一阶时序逻辑公式。一个含有一个或多个变元的时序逻辑公式定义了一个时序模式的类。对时序逻辑公式中的变元用特定的基值例化定义了一个时序模式。

基于以上定义,时序数据库中的知识发现任务可描述为:给定一个事实序列,一个时序模式的类及一个频度阈值,从该事实序列中发现全部频繁的时序模式。下面我们讨论两类时序模式的发现方法。

2.1 一阶时序逻辑公式定义的时序模式类

我们先从如式(1)所示的时序模式类开始讨论发现方法,然后再推广到一般情形。

$$a(X)\to\Diamond b(Y)\wedge\bigcirc^{T+n}b(Y)\rhd c(Z) \qquad (1)$$

其中a,b,c是时序数据库中的时序谓词,X,Y,Z分别是a,b,c的属性向量。该公式的直观意义是:如果a现在为真,那末最终会出现b为真的状态,并且b将一直为真,直到c出现的时刻,此时b为假。问题是要发现全部X,Y,Z的例化即时序模式,且这些时序模式是频繁的。图2是发现时序模式类(1)的TLP程序。我们在$a(X),b(Y),c(Z)$中分别引进了时序参数T,还引进了一个时序识别谓词$d(X,Y,Z,T)$。

```
(1)simtime(0)
(2)simtime(T)→simtime(T+1),¬simtime(T)
(3)simtime(T),a(X,T),¬a(X,T-1)→flag1(X,T1)
(4)simtime(T),a(X,T),a(X,T-1)→flag2(X,T)
(5)simtime(T),b(Y,T),¬b(Y,T-1)→flag3(Y,T2),¬
   flag2(X,T)
(6)simtime(T),b(Y,T),b(Y,T-1)→flag4(Y,T)
(7)simtime(T),C(Z,T),flag3(Y,T2),flag4(Y,T-1),
   (T2≤T3≤T)→d(X,Y,Z,T3)
(8)simtime(T),¬b(Y,T),flag3(Y,T2),flag4(Y,T-1),
   (T2≤T3≤T)→¬flag4(Y,T3),¬flag3(Y,T2),
   ¬flag1(X,T1)
(9)C(Z,T)→d(X,Y,Z,T)
```

图2 发现式(2)所示时序模式的TLP程序

规则(1)以谓词simtime作为系统时钟来仿真时间;规则(2)表示时间以"1"为时间粒度向前推进;当谓词$a(X)$在第一个时间点为真时,规则(3)即设置了falg1来标记;规则(4)继续设置flag2来标记a

(X)为真的时间段；当$b(Y)$发生时，规则(5)设置了谓词flag3，改变了flag2；规则(6)继续设置flag4来标记$b(Y)$为真的时间段；当$c(Z)$发生时，规则(7)设置了识别谓词$d(X,Y,Z)$，$d(X,Y,Z)$在从flag3被设置的时刻起到$c(Z)$发生的时刻止这一时间段为真；规则(8)在$b(Y)$为真的时间段结束以后重新设置flag4，flag3，falg1；规则(9)在$c(Z)$为真时重新设置了$d(X,Y,Z)$。

我们可以在以上的时序逻辑程序中加进一个计数器，对$q(X,Y,Z)$为真的时间粒度进行计数，然后再将此计数器的值与所给的频度阈值进行比较，以确定出频繁的时序模式。将这一例子进行一般化推广便得到以下定理：

定理 在一个时序数据库D中，对任何一条一阶时序逻辑公式φ所确定的时序模式类，存在一个对应的时序逻辑程序TLP及一个识别谓词d，使得对任何D中的有限事实，$d\leftrightarrows\varphi$，即d与φ同时成立。

该定理可以通过对φ中的时序算子的个数进行归纳来证明，其过程是构造性的。只要D有限且时序谓词的参数也有限，TLP程序就会正常终止。

2.2 由next算子○和and算子∧定义的时序模式类

这类时序模式如式(2)所示：

$$P_0(X_0)\wedge\bigcirc^{k_1}P_1(X_1)\wedge\bigcirc^{k_2}P_2(X_2)\wedge\cdots\wedge\bigcirc^{k_n}P_n(X_n) \quad (2)$$

其中，$1\leqslant k_1\leqslant\cdots\leqslant k_n$，$P_0,P_1,\cdots,P_n$是不同的时序谓词，$X_1,X_2,\cdots,X_n$是谓词的属性向量。

不难看出，该时序模式的频度是式(2)中自然数n的单调递减函数，即随n的增大，时序模式的频度逐渐降低。这样就一定存在一个n的最大值N，此时的时序模式是非频繁的。我们的目标就是要找出这样的最大值N，并且发现$n<N$时的所有频繁模式。实现这一目标的一个简单的方法就是从$n=1$开始，一步步随n以1递增来发现各个n值对应的频繁模式，每一步各自独立，不使用前次步骤的发现结果。换句话说，对第$i(1\leqslant i<N)$步的所有时序模式，不论频度高低，都要被计数。这样势必效率不高。考虑文[5]中讨论的发现数据库中关联规则的方法，在发现关联规则的时候，下一次迭代只利用前次迭代过程中生成的大的数据项集来生成新的候选数据项集以供进一步的评价。借鉴这一方法，在第i步发现频繁时序模式的时候，只利用第i-1步的发现结果，这就避免了对那些频度低的时序模式的计算，提高了算法的效率。为讨论方便，不妨先将式(2)简化成式(3)所示的形式：

$$P(X)\wedge\bigcirc^{k_1}P(X)\wedge\bigcirc^{k_2}P(X)\cdots\wedge\bigcirc^{k_n}P(X) \quad (3)$$

发现该时序模式的TLP程序如图3所示：

```
n=1，freq0=所有频繁基谓词的P的集合
repeat
    通过时序逻辑程序 TLPn 计算 freqn
    n=n-1
until (freqn=φ).
其中 TLPn 描述如下.
(1)simtime(0)
(2)simtime(T),T≤T*→simtime(T+1)
(3)simtime(T),P(X,T),P(X,T+K1),…,P(X,T+
   Kn-1),freqn-1(X,K1,…,Kn-1),P(X,T+Kn),count
   (X,K1,…,Kn-1,Kn,VAL),→holds(T,X,K1,…,
   Kn-1,Kn)→holds(T,X,K1,…,Kn-1,Kn),count(X,
   K1,…,Kn-1,Kn,VAL+1)
(4)simtime(T*+1),count(X,K1,…,Kn-1,Kn,VAL),
   (VAL>c)→freqn(X,K1,…,Kn-1,Kn)
```

图3 发现式(3)所示时序模式的TLP程序

T^*是时间域的最大值，$freq_n(X,K_1,\cdots,K_{n-1},K_n)$是第$n$步所发现的全部频繁模式，$count(X,K_1,\cdots,K_{n-1},K_n,VAL)$是频度计算谓词，$c$是给定的频度阈值，规则(3)中的谓词$holds(T,X,K_1,\cdots,K_{n-1},K_n)$避免了对频度的重复计算，其初始值为假。此程序以递归形式写出。它对每一个t值，只利用$t-1$步的发现结果，再发现t步中的频繁时序模式。程序一直执行到$freq_n=\phi$结束。

显然，此TLP程序可以推广到式(3)的一般情形，不同的地方只是程序中的谓词标记要复杂些，限于篇幅，不再详细给出。式(3)中的谓词组合是任意的，并且谓词的个数也是有限任意的，因此此程序存在着组合爆炸的危险。另外，程序的每次迭代都对应着一个TLP程序，所以程序的复杂度不再是线性的。

结束语 时序数据库中的知识发现是当前KDD领域一个较为热门的研究课题，有关论文尚不多见。本文介绍的方法是这方面的一个初步尝试，以期起到抛砖引玉的作用。

参考文献

1 王清毅，陈恩红，蔡庆生. 知识发现的若干问题及应用研究. 计算机科学，1997(5)

2 Baudient M, et al. Temporal Deductive Databases. Ch13 in Temporal Databases Theory Design and Implementation. Benjamin/Cummungs, 1993

3 Abadi M, Manna Z. Temporal Logic Programming. J. of Symb. Computation, 1989, 8: 277～295

4 Tuzhilin A. Sim TL: A Simulation Language Based on

(下转第89页)