

文章编号:1001-9081(2008)03-0726-03

## 带兴趣度的序列概念格模型及其构造

李 云,徐 涛,田素方,李 拓

(扬州大学 信息工程学院,江苏 扬州 225009)

(liyun@yzu.edu.cn)

**摘要:**为了有效地挖掘用户所需的序列模式,在序列数据库中对每个项目引入了兴趣度值以示其不同的重要程度。在对概念格模型进行扩展的基础上,提出了基于兴趣度的序列概念格模型及其构建算法。实例表明算法能减少冗余序列的产生,挖掘出满足用户需求的序列模式。

**关键词:**数据挖掘;序列模式;概念格;兴趣度

中图分类号: TP311.13;TP18 文献标志码:A

### Sequence concept lattice model with interest measure and its construction

LI Yun, XU Tao, TIAN Su-fang, LI Tuo

(Institute of Information Engineering, Yangzhou University, Yangzhou Jiangsu 225009, China)

**Abstract:** In order to mine efficiently the sequential patterns according to user's requirements, the interest measure for every item was introduced to reveal the different importance in the sequence database. And on the basis of extending the concept lattice model, the sequence concept lattice model with interest measure and its construction algorithm were presented. The instance shows that the algorithm can reduce the redundant sequences and mine the sequential patterns that satisfy user's requirements.

**Key words:** data mining; sequential patterns; concept lattice; interest measure

### 0 引言

序列模式的概念最早是由文献[1,2]提出的,序列模式挖掘是数据挖掘中的一个重要方向,且有着非常广泛的应用前景,包括顾客购买行为的分析、网络访问模式分析、科学实验的分析、疾病治疗的早期诊断、自然灾害的预测、DNA 序列模式的分析等。

目前,国内外研究者已经提出了很多序列模式挖掘方法,其中主要的序列模式挖掘算法可以分为三类:第一类是基于 Apriori 的候选码生成—测试的方法,第二类是基于垂直格式的候选码生成—测试的方法,第三类是基于模式增长的方法。文献[3,4]对序列规定了时间限制、滑动时间窗口和用户规定的分类,这些算法都是基于 Apriori 的水平格式的序列模式挖掘或者与时间相关的频繁模式挖掘。文献[5]提出了一种基于垂直格式存储的序列模式挖掘方法 SPADE 算法。文献[6]提出了一种基于投影的模式增长的算法——Freespan 算法,这个算法改进后为 Prefixspan 算法。概念格<sup>[7]</sup>是一种基于二元关系构造的模型,利用概念格可提取关联规则、分类规则和序列模式<sup>[8]</sup>等。

现有序列挖掘算法都有这样的前提假设:数据库中各项目相同的性质和作用,即重要性相同。仅简单地基于支持度阈值和置信度阈值进行序列模式挖掘,其存在明显的不足。故不能有效地解决不同项目有着不同重要性的问题。针对这个问题,文献[9]提出一种新的框架来提取序列模式,它通过给不同的项目引入权值,然后结合支持度来进行剪枝。然而

该算法依然会产生大量的冗余序列模式。

在现实世界的交易数据库中不同项目的利润不尽相同,商家真正关心的并不仅仅是频繁的序列模式,而是哪些序列模式的利润更大,哪种类型的顾客购买行为能带来更大的利润等。为了反映各个项目的不同重要性,我们引入了项目兴趣值概念,并对概念格模型进行扩展,形成基于兴趣度的序列概念格,并给出其构建算法。概念格的每个节点本质上是一个最大项目集,非常有利于序列模式发现。通过扫描格节点就能生成商家期望的兴趣序列,而不存在很多冗余的频繁序列,从而避免一些大利润商品(如电脑、钻石首饰等)由于销售量低于用户提出的支持度阈值和可信度阈值而排除,一些小利润商品(如日用品、小食品等)却往往由于销售量大而被频繁提取,然而商家对此却并不太感兴趣等问题。

### 1 相关背景知识

#### 1.1 序列模式相关概念

给定一个序列集合,其中每一个序列由项集构成,然后给定由用户确定的最小支持度阈值,序列模式挖掘就是去发现所有的频繁子序列,也就是找出所有出现频率不小于给定的最小支持度的子序列。

给定一个顾客交易的序列数据库,如表 1 所示。

**定义 1** 把每个商品称为一个数据项(Item,简称项),令非空集合  $I = \{i_1, i_2, \dots, i_m\}$ ,其中,  $i_j (j = 1, 2, \dots, n)$  是不同的项,项的集合称为项目集合(Itemset,简称项集),其中每个  $i_k (1 \leq k \leq m)$  是一个项,长度为  $k$  的项集称为  $k$  项集。

收稿日期:2007-09-18;修回日期:2007-12-07。 基金项目:国家自然科学基金资助项目(60575035;60673060)。

作者简介:李云(1965-),男,安徽合肥人,教授,博士,主要研究方向:概念格、数据挖掘; 徐涛(1979-),男,山东临沂人,硕士研究生,主要研究方向:概念格、数据挖掘; 田素方(1981-),男,山东巨野人,硕士研究生,主要研究方向:概念格、信息检索; 李拓(1983-),男,江苏丰县人,硕士研究生,主要研究方向:概念格、本体。

例如,甲顾客购买的商品为:MP4、Mobile telephone、Icebox、Digital Camera,依次用a、b、c、d表示,则甲顾客的项集为:4-项集(abcd)。乙顾客购买的商品为:MP4、Icebox、Digital Camera,则乙顾客的项集为:3-项集(acd)。

表1 交易序列数据库

Cid	Sequence
10	$\langle (abc)(ac)d(cf) \rangle$
20	$\langle (ab)(df) \rangle$
30	$\langle (eg)(af)b \rangle$
40	$\langle (ab)(cd)(bg)h \rangle$

定义2 序列(Sequence)是项集的有序表,记为 $S = \langle s_1, s_2, \dots, s_n \rangle$ ,其中 $s_k = \{i_1, i_2, \dots, i_m\}$ ( $1 \leq k \leq n$ )是非空项集,也称为该序列的一个元素,则序列的长度为该序列中所有项的和,记为:

$$\text{Length} = \sum_{i=1}^{\text{size}(s_1)} |I_1| + \sum_{i=1}^{\text{size}(s_2)} |I_2| + \dots + \sum_{i=1}^{\text{size}(s_k)} |I_k|$$

例如:序列 $10 = \langle (abc)(ac)d(cf) \rangle$ 的长度 $\text{Length} = 8$ 。

定义3 序列数据库 $D$ 是元组 $\langle Cid, S \rangle$ 的集合, $Cid$ 为序列标识号(顾客号或交易号),如果序列 $T$ 是 $S$ 的子序列(即 $T \subseteq S$ )称元组 $\langle Cid, S \rangle$ 包含序列 $T$ ;序列 $T$ 在序列数据库 $D$ 中的支持数是数据库中包含 $T$ 的元组数,即 $\text{Support}_D(T) = |\{\langle Cid, S \rangle \mid \langle Cid, S \rangle \in D \wedge T \subseteq S\}|$ ,记作 $\text{Support}(T)$ 。

定义4 给定一个最小支持度阈值 $minsup$ ,如果序列 $S$ 的支持度不小于 $minsup$ ,则称序列 $S$ 为频繁序列模式,所有的频繁序列模式集合记作 $FS$ 。

例如: $\langle (af) \rangle$ 这个序列分别在10和20序列出现2次,所以其支持数为2。假设用户规定 $minsup = 2$ ,可知 $\langle (af) \rangle$ 序列是频繁序列模式。

## 1.2 概念格相关知识

对于给定的数据信息表 $K = (G, M, I)$ ,在形式概念分析中称为形式背景,其中 $G$ 是对象集合, $M$ 为属性集合, $I$ 是 $G$ 和 $M$ 间的二元关系。对于一个对象 $g \in G$ ,属性 $m \in M$ ,那么 $gIm$ 就表示对象 $g$ 具有属性 $m$ 。

形式背景的对象集 $A \in P(G)$ ,属性集 $B \in P(M)$ 之间按如下关系连接:

$$\begin{aligned} f(A) &= \{m \in M \mid \forall g \in A, gIm\} \\ g(B) &= \{g \in G \mid \forall m \in B, gIm\} \end{aligned} \quad (1)$$

则称从形式背景中得到的每一个满足 $A = g(B)$ , $B = f(A)$ 的二元组 $(A, B)$ 为一个形式概念。其中 $A$ 是对象密集 $P(G)$ 的元素,称为概念 $(A, B)$ 的外延, $B$ 是属性密集 $P(M)$ 的元素,称为概念 $(A, B)$ 的内涵。

若概念 $C_1 = (A_1, B_1)$ 和 $C_2 = (A_2, B_2)$ ,满足 $A_1 \subseteq A_2$ ,则称 $(A_1, B_1)$ 为子概念(或亚概念), $(A_2, B_2)$ 为父概念(或超概念),记为: $(A_1, B_1) \leqslant (A_2, B_2)$ 。这种由形式背景中所有形式概念的超概念—亚概念的偏序关系(也称泛化—特化关系)所诱导出的格称为概念格,记为 $L(K)$ 。

## 2 序列概念格模型及构造

### 2.1 项目兴趣度

在一个交易数据库中,商家真正关心的可能并不是哪种商品卖的最多,而是何种商品能带来最大的利润,每个顾客的购买行为是如何的。但是,数据库中的各个项目是平等对待

的,只是简单地基于支持度阈值和置信度阈值进行挖掘,在对普通的交易数据库进行序列模式挖掘出的频繁序列并不一定能满足商家的要求。不同商品的利润和其销量的差异是很大的,如表2所示的商场交易表。

表2 一个商场的交易利润表

I-ID	Item	Profits(¥)	Support (frequency)
a	MP4	120	10 000
b	Mobile telephone	380	1 200
c	Ice box	420	500
d	Digital Camera	550	1 000
e	TV SET	680	400
f	DVD	250	800
g	Air condition	720	300
h	Computer	1 080	200

根据各个项目(商品)的利润的不同而赋予不同的兴趣值,假设给定项目集合 $I = \{i_1, i_2, \dots, i_m\}$ ,为表征项目的重要性,我们为每一个项目 $i_n$ 建立下面的简单的兴趣值映射函数,其中, $W(i_n)$ 表示项目 $i_n$ 在其利润为 $x$ 下的映射兴趣值。

$$w(i_n) = f(x_n) = \begin{cases} \vdots & \vdots \\ 0.9, & x \in [800, 900) \\ 0.8, & x \in [700, 800) \\ \vdots & \vdots \\ 0.2, & x \in [100, 200) \\ 0.1, & x \in (0, 100) \end{cases} \quad (2)$$

利用式(2)对表2的交易数据库的项目集进行映射得出表3。

表3 交易数据库项目兴趣值

Item	W( $i_n$ )	Item	W( $i_n$ )
a	0.2	e	0.7
b	0.4	f	0.3
c	0.5	g	0.8
d	0.6	h	1.1

定义5<sup>[8]</sup> 给定一个序列 $S = \langle s_1, s_2, \dots, s_n \rangle$ ,其中 $s_j$ 是项集 $(i_{j1}, i_{j2}, \dots, i_{jm})$ ,则其序列的兴趣度值定义为:

$$W(S) = \frac{\sum_{i=1}^{\text{Length}(s_1)} W(x_{1i}) + \sum_{i=1}^{\text{Length}(s_2)} W(x_{2i}) + \dots + \sum_{i=1}^{\text{Length}(s_m)} W(x_{mi})}{\text{Length}(s_1) + \text{Length}(s_2) + \dots + \text{Length}(s_m)} \quad (3)$$

则,该序列在该序列数据库中的兴趣度值定义为:

$$Weight(S) = W(S) \cdot Support(S) \quad (4)$$

例如:以表1的交易序列数据库为例,并以表3为其项的兴趣度值,则序列 $10 = \langle (abc)(ac)d(cf) \rangle$ 的兴趣度值为:

$$Weight(10) = \frac{(0.2 + 0.4 + 0.5) + (0.2 + 0.5) + 0.6 + (0.5 + 0.3)}{3 + 2 + 1 + 2} = 0.4 \quad (5)$$

因其序列10在表1的交易数据库中支持数为1,故, $Weight(10) = 0.38 \times 1 = 0.38$ ,序列模式 $\langle (ab) d \rangle$ 的兴趣度值 $W(\langle (ab) d \rangle) = \frac{(0.2 + 0.4) + 0.6}{2 + 1} = 0.4$ ,序列模式 $\langle (ab) d \rangle$ 在表1的交易数据库中支持数为3,在交易数据库中的 $Weight(\langle (ab) d \rangle) = 0.4 \times 3 = 1.2$ 。

## 2.2 序列概念格模型

给定一个由顾客交易 (Transaction) 组成的交易数据库 DB 中, 顾客的交易序列是由事件 (Event) (即: 顾客每次的交易子序列) 按交易时间的次序升序排列, 每个交易描述一位顾客的所有交易行为(或者说每位顾客的每一笔交易都是该顾客交易序列的一个子集), 该行为由顾客号 (Cid) 和交易序列 (Sequence) 组成。为了表示每个交易序列中的事件的先后序列关系, 为序列中每个事件分配唯一的有序标识符 Eid, 那么在交易数据库中的每个交易序列可以用客户号与事件有序标识符 ( $Cid, Eid$ ) 集合表示。例如表 1 所示序列  $10 = \langle a(abc)(ac)d(cf) \rangle$  中的每个事件描述为:  $(10,1) = (a)$ ,  $(10,2) = (abc)$ ,  $(10,3) = (ac)$ ,  $(10,4) = (d)$ ,  $(10,5) = (cf)$ 。

**定义 6** 一个形式背景表示为  $K = ((Cid, Eid), Event, W)$ , 其中,  $(Cid, Eid)$  为交易顾客集  $(Cid, Eid) = \{(Cid_i, Eid_i), (Cid_i, Eid_2), \dots, (Cid_i, Eid_j) | N\}$ ,  $Event$  为交易事件集  $Event = \{Event_1, Event_2, \dots, Event_M\}$ ,  $W$  为兴趣度值, 它表示  $(Cid, Eid)$  与  $Event$  (即交易顾客的交易序列) 的兴趣度值。

**定义 7** 在形式背景  $K = ((Cid, Eid), Event, W)$  中,  $t \subseteq (Cid, Eid)$ ,  $e \subseteq Event$ , 在  $(Cid, Eid)$  和  $Event$  之间可定义两个映射  $f$  和  $g$ :

$$\begin{aligned} f(t) &= \{e \mid \forall t \in (Cid, Eid), e \in t\} \\ g(e) &= \{(Cid, Eid) \mid \forall e \in E, e \in t\} \end{aligned} \quad (6)$$

**定义 8** 如果二元组  $(t_1, e_1)$  ( $t_1 \subseteq (Cid, Eid)$ ,  $e_1 \subseteq Event$ ) 满足:  $t_1 = g(e_1)$ ,  $e_1 = f(t_1)$ , 则称作  $K$  的一个感兴趣序列概念,  $t_1$ ,  $e_1$  是其概念的外延和内涵。 $K$  的所有感兴趣序列概念集合记为  $WCS(K)$ ,  $WCS(K)$  上的结构是通过泛化例化关系产生的, 定义为  $t_1 \sqsubseteq t_2$ , 则  $(t_2, e_1) \leq (t_1, e_1)$ 。通过此关系得到的有序集  $WCS(K) = (WCS(K), \leq)$ , 称作  $K$  的基于兴趣度的序列概念格。

## 2.3 序列概念格的构造

传统形式背景的概念格构造算法主要分为两类: 批处理算法和渐进式构造算法。考虑到交易序列数据库的是不断更新的, 本文在渐进式生成算法思想的基础上, 提出了一种基于兴趣度的序列概念格构造算法。

首先, 对原始的交易数据库进行预处理: 根据函数计算出每个项的兴趣度值, 并把序列数据库转化为序列概念格的形式背景。

算法: 基于兴趣度的序列概念格构造

输入: 已构建好的序列概念格 SCL, 新追加的对象  $x^*$ , 形成三元组  $CX = (x^*, f(\{x^*\}), w(f(\{x^*\})))$ , 序列兴趣度的阈值:  $min-weight$  (专家给出)。

输出: 更新后的序列概念格 SCL。

BEGIN

```

Mark := ∅;           /* Mark 存放更新格节点集合 */
For SCL 中的每个格节点 CM = ((Cid, Eid), Event, W
(Event)) 按! Event !升序字典序排序 Do
If Event ⊆ f({x*}) Then          /* 更新概念 */
  extent(c) ∪ {x*};
  Mark := Mark ∪ {CX};
If Event = f({x*}) Then
end for
Else
Call Cnew ();

```

```

If Newinter = ∅ Then
  Mark := Mark ∪ {CX};
End if
End for
End
Procedure Cnew ()
Newinter := Event ∩ f({x*});      /* 可能是产生子概念 */
If Newinter ≠ ∅ Then
  If 不存在 CK ∈ Mark 使得 Event K := Newinter Then
    /* CK 是 CX 的祖先节点 */
    NFL = (extent(c) ∪ {x*}, intent(c) ∩ f(x*),
    w((intent(c) ∩ f(x*))));
    /* NFL 是新增节点 */
    If w((intent(c) ∩ f(x*))) > min-weight Then
      Mark := Mark ∪ {NFL};
      Add edge CX ← NFL
    Else
      Delete NFL;
    End if
    For 对 Mark 中的每个格节点 CN = ((Cid, Eid)', Event',
    W(Event)') 按! Event !字典序降序排序 Do
      If 存在 CN ∈ Mark 使得 Event' ⊂ ; Newinter Then
        Add edge NFL ← CN;
      If 存在 CN 是 CX 的双亲 Then
        Delete edge CX ← CN;
      End for
    End if
  End Cnew
  把表 1 的交易数据库由表 3 的项目兴趣值转化为序列形式背景, 如表 4 所示。

```

表 4 由表 1、表 3 转化的序列形式背景

(Cid, Eid)	a	b	c	d	e	f	g	h
(10, 1)	0.2	0.4	0.5					
(10, 2)	0.2		0.5					
(10, 3)				0.6				
(10, 4)				0.5		0.3		
(20, 1)	0.2	0.4						
(20, 2)				0.6		0.3		
(30, 1)					0.7		0.8	
(30, 2)	0.2					0.3		
(30, 3)			0.4					
(40, 1)	0.2	0.4						
(40, 2)				0.5	0.6			
(40, 3)			0.4				0.8	
(40, 4)								1.1

根据表 4 的序列形式背景应用本文提出的序列概念格构造算法生成的序列概念格如图 1 所示, 设定其  $min-weight = 0.6$ 。

在图 1 的格节点中每个节点由三元组  $\{(obj), < attr>, Weight(S)\}$  构成, 如节点  $\{<(10,2)(20,2), \langle a,b \rangle, 0.9>\}$  表示 ID 号为 10, 20 的顾客在交易过程中都在第 2 次交易中购买了商品项目  $\langle a,b \rangle$ , 并且商品的兴趣值为 0.9。由格图可以清晰的看出能给商家带来最大利润的是  $\{<(10,1)(10,2)(10,4)(40,2), \langle c \rangle, 2.0>\}$  和  $\{<(10,1)(20,1)(30,3)(40,1)(40,3), \langle b \rangle, 2.0>\}$  节点, 项目  $c$  的兴趣值为 2.0, ID 号为 10 的顾客分别在  $Eid = 1, 2, 4$  的事件

(下转第 741 页)

(b) 中可以看出,由模式识别所得该时段的交通状况为顺畅交通状态,与所分析的结果一致。时间段 7:52~11:15 时间段,在 A20SVD10087 检测器采集的交通流数据分布图中,交通流参数分布与其他时间段的分布有很大的差别,主要表现在该时间段内速度骤然降低,而占有率上升很大的幅值,在早高峰时间段,该路段经常会有拥堵现象发生。图 4(b) 中所得交通状况为拥挤交通状态和堵塞交通状态相互交替,与图 4(a) 中的变化相一致。由模式识别分析检验可知,辨别所得的交通状态与实际的交通流状态变化相符合。说明了交通状态分级能够很好的反映出道路实际情况,由聚类分析所得交通状态的量化结论合理并具有较强的实际指导意义。

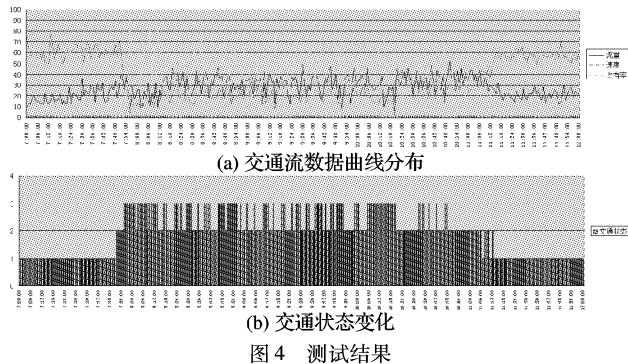


图 4 测试结果

## 4 结语

本文从检测器采集的交通信息集数据挖掘入手,详细介绍了整个信息挖掘过程的各阶段的要求和算法,即数据获取、数

(上接第 728 页)

中购买了该商品, ID 号为 40 的顾客在  $Eid = 2$  的事件中购买过该商品,由此得出的兴趣序列模式从高到低为:  $\{ \langle c \rangle, \langle b \rangle, \langle d \rangle, \langle g \rangle, \langle h \rangle, \langle a \rangle, \langle a, b \rangle, \langle f \rangle, \langle e, g \rangle, \langle a, c \rangle, \langle b, g \rangle \}$ 。它们对应的兴趣值为:  $\{ 2, 0.2, 0.1, 1.8, 1.6, 1.1, 1.0, 0.9, 0.9, 0.8, 0.7, 0.6 \}$ 。在本例中,序列模式  $\langle e, g \rangle$  的支持度仅为 1,但是其兴趣值为 0.60,大于最小感兴趣值阈值 0.6,故保留下来,而不会被剪枝掉,而如果按照普通的序列模式算法进行挖掘将会被剪枝掉。

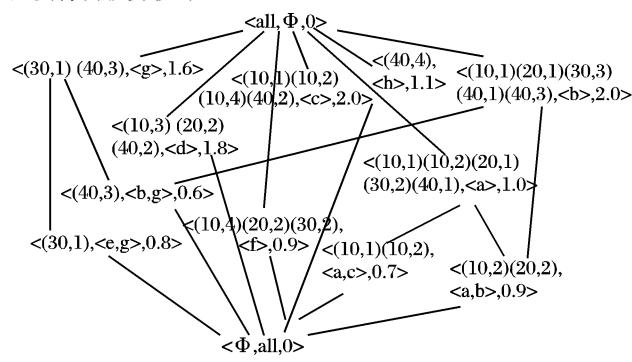


图 1 由表 4 的序列形式背景构造的格

## 3 结语

现实数据库中不同的项目往往有着不同重要性,为反映各个项目的不同的重要性,本文引入了项目兴趣值,从而扩展了现有的问题模型,提出了新的感兴趣序列模式发现问题。并对概念格模型进行扩展,形成基于兴趣度的序列概念格模型,给出了基于兴趣度的序列概念格构建算法。通过扩展概念格进行序列模式挖掘可以对该顾客购买行为进行分析,并

据预处理、挖掘方法、结果分析与评价、模式应用等。而且整个过程的部分阶段是一个不断反复的过程,直至得到满意的交通状态分类。该过程还可以对检测器采集的交通信息提取其他决策级的信息,如行程时间、事故判别等。同时也对智能交通系统其他海量信息挖掘的一般过程提供较强的参考价值。

### 参考文献:

- [1] HAND D, MANNILA H, SMYTH P. Principles of data mining [M]. Cambridge: Massachusetts London England, The MIT Press, 2001: 6~8.
- [2] KIRSCHFINK H. Basic tools for fuzzy modeling [C/OL]. [2007-09-15]. [http://www.erudit.de/erudit/events/tc-c/03\\_Fuzzy-Modeling.pdf](http://www.erudit.de/erudit/events/tc-c/03_Fuzzy-Modeling.pdf).
- [3] 郭桂蓉, 谢维信. 模糊模式识别 [M]. 长沙: 国防科技大学出版社, 1992: 169~178.
- [4] 耿彦斌. 城市道路交通流数据质量控制理论与模型 [D]. 北京: 北京交通大学, 2006: 2~8.
- [5] CHEN CHAO. Detecting errors and imputing missing data for single loop surveillance systems [C]// Proceedings of 82nd Transportation Research Board (TRB) Annual Meeting. Washington, DC: IEEE Press, 2003: 6~16.
- [6] ISHAK S. Quantifying uncertainties of freeway detector observations using fuzzy-clustering approach [C]// Transportation Research Record, Transportation Research Board 82nd Annual Meeting. Washington, DC: IEEE, 2003: 2~7.
- [7] 王沫然. Matlab 与科学计算 [M]. 北京: 电子工业出版社, 2004.
- [8] 周商吾. 交通工程 [M]. 上海: 同济大学出版社, 1987: 22~24.

能挖掘出期望的兴趣序列模式,减少冗余序列的产生。

### 参考文献:

- [1] AGRAWAL R, IMIELINSKI T, SWAMI A. Mining association rules between sets of items in large databases [C]// Proceedings of the ACM SIGMOD Conference on Management of Data. Washington, DC: ACM Press, 1993: 207~216.
- [2] AGRAWAL R, SRIKANT R. Fast algorithm for mining association rules [C]// Proceedings of International Conference on Very Large Data Bases. Santiago: IEEE Press, 1994: 487~499.
- [3] PARK J S, CHEN MINI-SYAN, YU P S. Efficient parallel data mining for association rules [C]// Proceedings of 4th International Conference on Knowledge and Data Engineering. New York: ACM Press, 1995: 31~36.
- [4] AGRAWAL R, SHAFER J. Parallel mining of association rules [J]. IEEE Trans on Knowledge and Data Engineering, 1996, 8(6): 962~969.
- [5] ZAKI M. SPADE: An efficient algorithm for mining frequent sequences [J]. Machine Learning, 2001, 41(1/2): 31~60.
- [6] PEI J, HAN J, PINTO H, et al. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth [C]// Proceedings of Int Conference on Data Engineering (ICDE01). Heidelberg: IEEE Press, 2001: 215~224.
- [7] WILLE R. Restructuring Lattice theory: An approach based on hierarchies of concepts [J]. Ordered Sets, 1982, 11(5): 445~470.
- [8] 孙莹, 胡学钢. 基于频繁概念格的序列模式发现研究 [J]. 计算机科学, 2004, 31(10A).
- [9] YUN U. A new framework for detecting weighted sequential patterns in large sequence databases [J/OL]. [2007-04-19]. <http://www.sciencedirect.com/>.