序列模糊概念格模型及其渐进式构造

李 云 袁运浩 盛 艳 陈 峻
(扬州大学信息工程学院 扬州 225009)

摘 要 传统的序列模式挖掘主要是挖掘满足最小支持度的频繁序列,没有考虑序列在实际中的重要度。为了能够有效地挖掘重要的序列模式,提出了一种序列模糊概念格模型,对所有序列的项目引入了重要度权值,定义了序列的重要度及可以动态调整最小支持度 minsup 的自适应系数;扩展了模糊形式背景,使其能够方便地表示序列,定义了概念的 Galois 闭包连接、序列模糊概念及序列模糊格结构,并给出了序列模糊概念格的渐进式构造算法 SeqFuzCL。实验表明,序列模糊概念格模型可以方便有效地组织自适应序列模式,在时间与空间上都具有良好的性能,并为进一步挖掘自适应序列模式提供了理论支持。

关键词 序列模式,模糊形式背景,模糊概念格,渐进式构造

中图法分类号 TP18 **文献标识码** A

Sequence Fuzzy Concept Lattice Model and Incremental Construction

LI Yun YUAN Yun-hao SHENG Yan CHEN Ling
(School of Information Engineering, Yangzhou University, Yangzhou 225009, China)

Abstract Traditional algorithms in mining sequence patterns only discover the frequent sequences satisfying the minimum support threshold minsup; however, these methods don't consider the importance of actual sequences. In order to mine the important sequence patterns satisfying users' many demands, this paper presented a new sequence fuzzy concept lattice model. The model firstly introduced the weight for each item in sequences. On the basis of the weight, we defined the weight of every sequence and the self-adaptive coefficient that may dynamically adjust the minimum support threshold minsup. And then the fuzzy formal context was extended to express sequences in brief. Making use of the sequence fuzzy formal context, Galois connection, sequence fuzzy conception and sequence fuzzy concept lattice were defined in the paper. At last, this article presented the incremental construction algorithm SeqFuzCL of the sequence fuzzy concept lattice. The experimental results show that the algorithm SeqFuzCL can effectively express self-adaptive sequence patterns in the lattice, and has excellent performance on the time-spatial complexity. Simultaneously, the model provides theoretic support for mining self-adaptive sequence patterns.

Keywords Sequence pattern, Fuzzy formal context, Fuzzy concept lattice, Incremental construction

1 引言

自从 R. Srikant 与 R. Agrawal 提出频繁序列挖掘[1]以来,序列模式挖掘已经成为数据挖掘研究的主要内容之一,并在顾客购买行为分析、网络访问模式分析、科学实验分析、疾病治疗的早期诊断、自然灾害的预测以及 DNA 序列模式分析等领域中有着广泛的应用。

目前,已经提出了许多序列模式的挖掘算法,大体可将序列模式挖掘算法分为两类。一类是基于候选码生成-测试的 AprioriAll 算法[1]、GSP 算法[2]、SPADE 算法[3]等,这类算法需要生成大量的候选序列集,并在此基础上生成序列模式;一类是基于模式增长的 FreeSpan 算法[4]、PrifixSpan 算法[5]等,该类算法在数据库上采用了投影技术,挖掘频繁序列时不需要生成候选序列集,而是直接在前缀对应的投影数据库上挖掘序列模式。

通过上述两类算法,可以挖掘出满足用户指定的最小支持度的序列模式,但这些算法并没有考虑序列模式的重要性,即虽算法挖掘出的所有序列模式都满足用户指定的最小支持度,但用户可能更关注比较重要的序列模式它们虽然不能满足用户指定最小支持度,但是这些序列对用户来说比较有价值,例如在入侵检测中,多数用户访问的时序模式属于正常模式,但检测者更期望得到异常的访问时序模式;在 DNA 序列中,对于某些特殊的疾病,可能取决于一些异常的基因序列等等;或者有些序列模式对用户来说重要程度不大,并不需要挖

到稿日期:2010-04-22 返修日期:2010-10-03 本文受国家自然科学基金(61070047,61070133,61003180),江苏省自然科学基金(BK2008206,BK2010311),江苏省教育厅自然科学基金(08KJB520012)资助。

李 云(1965-),男,博士,教授,主要研究方向为概念格、数据挖掘等;袁运浩(1983-),男,博士生,主要研究方向为概念格、模式识别、并行计算等;盛 艳(1985-),女,硕士生,主要研究方向为概念格、信息检索等;陈 峻(1951-),男,教授,博士生导师,主要研究方向为并行计算、系统优化等。

掘，这就需要算法能够自适应地调整以挖掘出符合用户需求的序列模式，但已提出的挖掘算法没有考虑这种特征，无法挖掘出这样的序列模式。

此外，文献[6]提出了一种加权的序列模式挖掘算法WSpan。WSpan算法通过对每个项目引入不同的权值，并结合用户指定的最小支持度 minsup 来挖掘频繁序列。但 WSpan算法需要多次扫描数据库，并会产生大量冗余序列模式，同时 WSpan算法也没有解决上述序列模式挖掘问题。

而形式概念分析[7]中的核心数据结构——概念格模型由于只需访问一次数据库就可构建成功，并且它的知识与层次表达能力强，将序列引入概念格中，只需存储最大公共子序列，减少了冗余序列的产生。因此，为了解决这类序列模式挖掘问题，本文提出了一种序列模糊概念格模型，在该模型上利用本文定义的序列模糊形式背景可以方便地表示权重序列；在序列权重与序列重要度阈值的基础上，定义了序列自适应系数，它可以对序列模式的最小支持度 minsup 进行动态的调整。在此基础上，为了有效地寻找与组织有意义的自适应序列，本文将模糊形式背景在序列上进行了扩展；利用扩展的序列模糊形式背景，定义了概念的 Galois 闭包连接、序列模糊概念及其格结构；最后给出了序列模糊概念格的构建算法。实验表明，序列模糊概念格模型可以方便有效地组织自适应序列模式，在时间与空间上都具有良好的性能，为进一步挖掘自适应序列模式提供了理论支持。

2 基本概念

2.1 序列模式

在序列模式挖掘中，所有项目(item)构成的非空集合称为项目集(itemset)，记为 $I=(i_1,i_2,\cdots,i_n)$，其中 $i_j(j=1,2,\cdots,n)$代表项目，$i_j\in T$ 且 $T\subseteq I$，T 代表事务的集合；一个序列 S 是项目集的有序列表，记为 $S=\langle s_1,s_2,\cdots,s_m\rangle$，其中 $s_k(k=1,2,\cdots,m)$是一个项目集，且 $s_k\subseteq I$，一个序列 S 的长度是其包含所有项目的个数，即 $|S|=\sum_{1\leqslant k\leqslant m}|s_k|$。设序列 $S_a=\langle a_1,a_2,\cdots,a_n\rangle$，$S_b=\langle b_1,b_2,\cdots,b_m\rangle$，若存在 $i_1<i_2<\cdots<i_n$ 使得 $a_1\subseteq b_{i_1}$，$a_2\subseteq b_{i_2}$，$\cdots$，$a_n\subseteq b_{i_n}$，则称序列 S_a 是序列 S_b 的子序列，序列 S_b 是序列 S_a 的超序列，记为 $S_a\subseteq S_b$；若两序列长度不等，则构成真包含关系，记为 $S_a\subset S_b$。序列数据库 SDB 记为(Seq_ID,S)，其中 Seq_ID 是序列 id 号，S 代表一条序列，如表 1 所列；若序列 S 包含在一个数据序列中，则称该数据序列支持 S。

表 1 序列数据库 SDB

Seq_ID	Sequence
1	⟨(ac)(d)(e)(a)(f)⟩
2	⟨(e)(a)(b)⟩
3	⟨(e)(abc)(bde)(f)⟩
4	⟨(f)(a)(ef)(g)⟩

2.2 模糊概念格[8]

在模糊形式概念分析中，模糊形式背景通常使用一个三元组 $K=(G,M,R)$ 表示，其中，G 是对象集合，M 是属性集合，映射 R 是隶属度，满足 $R:G\times M\to[0,1]$，或 $R(g,m)=v$，其中 $g\in G,m\in M,v\in[0,1]$。对于模糊形式背景 K 中的每个属性，均有两个阈值 θ_m 和 φ_m 满足 $0\leqslant\theta_m<\varphi_m\leqslant1$，阈值 θ_m 和 φ_m 可以根据领域背景知识确定，也可以由用户依据应用的目的指定，模糊形式背景的示例如表 2 所列。在模糊形式背景 K 中，对象集 $A\in P(G)$和属性集 $B\in P(M)$两者之间存在着一种闭包运算，即定义 $A'=\{m\in M|\ \forall g\in A,\theta_m\leqslant R(g,m)\leqslant\varphi_m\}$，$B'=\{g\in G|\ \forall m\in B,\theta_m\leqslant R(g,m)\leqslant\varphi_m\}$，为了表示的方便性，本文将 g'表示为$\{g\}'$，将 m'表示为$\{m\}'$。

表 2 模糊形式背景 K 与阈值

R	a	b	c	d	e	f	g
1	1	0	0.6	0.1	0.6	0.6	0
2	0.9	0.1	0	0	1	0	0
3	0.5	0.1	0.5	0.1	1	0.5	0
4	0.5	0	0	0	0.5	1	0.1
θ_m	0.6	0.1	0.4	0.1	0.5	0.3	0.1
φ_m	1.0	0.6	0.7	0.5	0.9	1.0	0.8

此外，在模糊形式背景 K 中还有两个模糊参数 σ 与 λ。对于对象集 $A\in P(G)$和属性集 A'，模糊参数 σ 定义形式如下：$\sigma_m=\frac{1}{|A|}\sum_{g\in A}R(g,m)$，$\sigma=\frac{1}{|A'|}\sum_{m\in A'}\sigma_m$；模糊参数 λ 定义形式如下：$\lambda=\frac{1}{|A'|}\sum_{m\in A'}\sqrt{\frac{\sum_{g\in A}(R(g,m)-\sigma_m)^2}{|A|}}$，其中 $|A|\neq0$，$|A'|\neq0$。在模糊形式背景 K 上，对于 $A\in P(G)$和 $B\in P(M)$，若一个四元组 $C=(A,B,\sigma,\lambda)$满足 $A'=B$ 且 $B'=A$，则称 C 是模糊形式背景 K 上的一个模糊概念。其中，A 称作模糊概念 C 的外延，B 称作概念 C 的内涵。

特别地，对 $A=\Phi,A'=M,\sigma=\lambda=0$；对 $B=\Phi,B'=G,\sigma=\lambda=0$。

在模糊形式背景 K 上任意给定两个模糊概念 $C_1=(A_1,B_1,\sigma,\lambda)$与 $C_2=(A_2,B_2,\sigma,\lambda)$，若两者满足 $A_1\subseteq A_2$，则称 C_1 是 C_2 的子模糊概念，C_2 是 C_1 的超模糊概念。这种模糊概念之间的偏序关系，通常用符号“≤”表示，记为 $C_1\leqslant C_2$。在模糊形式背景 K 上，由所有模糊概念的超模糊概念-子模糊概念的偏序关系所诱导出的格结构称为模糊概念格，记为 $L(K)$。

3 序列模糊概念格模型

3.1 自适应序列模式

自适应序列模式(Self-adaptive Sequence Pattern,SASP)通过在序列的每个项目上附加其重要度来自适应地调整用户指定的最小支持度；并在项目相对重要度与项目在序列中位置信息的基础上将序列数据库 SDB 转变为模糊形式背景，在此基础上构建序列模糊概念格，以挖掘出满足用户需求的序列模式。

定义 1 在序列数据库 SDB 中，令所有项目的集合 $I=(i_1,i_2,\cdots,i_n)$，对于任意的项目 $i_j\in I(j=1,2,\cdots,n)$，称 $IW(i_j)$是项目 i_j 的权重，称 $IWV=(W(i_1),W(i_2),\cdots,W(i_n))$是项目集 I 对应的权重向量，其中 $0\leqslant W(i_j)\leqslant1$。

序列数据库 SDB 中的项目权重可由用户根据需求指定，也可由相关领域专家来确定，如在表 1 的序列数据库 SDB 中，权重向量 $IWV(a,b,c,d,e,f,g)=(0.9,0.1,1,0.25,1,1,0.2)$。

定义 2 在序列数据库 SDB 中，一个序列 $S=\langle s_1,s_2,\cdots,s_m\rangle$，对于任意的项目 $i_j\in I(j=1,2,\cdots,n)$在序列 S 中出现的次数称为项频，记为 $\mathrm{INumS}(i_j)$；在序列 S 中出现的位置集合称为项位集，记为 $\mathrm{IPosS}(i_j)$。

例如，在序列数据库 SDB 中，id 为 3 的序列$\langle(e)(abc)(bde)(f)\rangle$中每个项目的项位集分别如下：IPos3$(a)=\{2\}$，IPos3$(b)=\{2,3\}$，IPos3$(c)=\{2\}$，IPos3$(d)=\{3\}$，IPos3$(e)=\{1,3\}$，IPos3$(f)=\{4\}$。

定义 3 在序列数据库 SDB 中，一个序列 $S=\langle s_1,s_2,\cdots,s_m\rangle$，若对于 $\forall s_k\in S$，设项目集 $s_k=(i_{k1},i_{k2},\cdots,i_{k|s_k|})$，那么称 $SW(S)=\dfrac{\sum_{k=1}^{m}\sum_{j=1}^{|s_k|}IW(i_{kj})}{\sum_{k=1}^{m}|s_k|}$ 是序列 S 的权重。

引理 1 若一个序列 $S=\langle s_1,s_2,\cdots,s_m\rangle$，则其权重 $SW(S)$必满足 $0\leqslant SW(S)\leqslant 1$。

证明：根据定义 1，可知 $0\leqslant IW(i_{kj})\leqslant 1$，所以对于 $\forall s_k\in S$，$0\leqslant\sum_{j=1}^{|s_k|}IW(i_{kj})\leqslant|s_k|$，所以 $0\leqslant\sum_{k=1}^{m}\sum_{j=1}^{|s_k|}IW(i_{kj})\leqslant(|s_1|+|s_2|+\cdots+|s_m|)=\sum_{k=1}^{m}|s_k|$，故 $0\leqslant\dfrac{\sum_{k=1}^{m}\sum_{j=1}^{|s_k|}IW(i_{kj})}{\sum_{k=1}^{m}|s_k|}\leqslant 1$，原命题得证。

由于挖掘出的序列模式需要体现对用户的重要程度，因此，在用户指定最小的支持度与项目权重（也可以是专家确定）后，算法应该根据序列模式的重要度自适应地调整最小支持度，以挖掘出用户感兴趣的序列，而用户认为不重要的序列模式应该尽量减少。

定义 4 对于一个序列 $S=\langle s_1,s_2,\cdots,s_m\rangle$，若存在一个实数 β 且 $0\leqslant\beta\leqslant 1$，当 $SW(S)\geqslant\beta$ 时，序列 S 被认为是重要的，当 $SW(S)<\beta$ 时，序列 S 被认为是非重要的，那么称 β 为序列 S 的重要度阈值。

重要度阈值 β 可由用户根据实际需要确定，也可以根据领域背景知识由专家来确定。

定义 5 在序列数据库 SDB 中，设最小支持度为 minsup，重要度阈值为 β，对于序列 $S=\langle s_1,s_2,\cdots,s_m\rangle$，若 ρ 满足下列条件：

①当 $SW(S)\geqslant\beta$ 时，序列 S 的支持度 Support$(S)\geqslant\rho$ minsup 且 $0\leqslant\rho\leqslant 1$；

②当 $SW(S)<\beta$ 时，序列 S 的支持度 Support$(S)\geqslant\rho$ minsup 且 $1<\rho\leqslant\dfrac{1}{\text{minsup}}$；

则称 ρ 是序列 S 的最小支持度 minsup 的自适应系数，简称自适应系数；对于满足 Support$(S)\geqslant\rho$ minsup 的序列 S 称为自适应序列模式。

通过自适应系数 ρ 可以对序列模式的最小支持度进行动态的调整，从而有效地挖掘用户认为重要的序列模式。自适应系数 ρ 的取值范围为 $0\leqslant\rho\leqslant\dfrac{1}{\text{minsup}}$，可根据具体应用设定。对于序列 S，本文根据其重要度即权重来确定自适应系数 ρ 的取值，如下：

$$\rho=\begin{cases}SW(S), & SW(S)\geqslant\beta\\ \dfrac{\text{minsup}-1}{\beta\text{minsup}}SW(S)+\dfrac{1}{\text{minsup}}, & SW(S)<\beta\end{cases}\qquad(1)$$

由式(1)可知，当序列数据库 SDB 中的每个项目均被平等地对待，即每个项目的权重 $IW(ij)=1$ 时，自适应系数 $\rho=1$，此时，自适应序列模式的挖掘就是传统频繁序列的挖掘。

引理 2 若自适应系数：

$$\rho=\begin{cases}SW(S), & SW(S)\geqslant\beta\\ \dfrac{\text{minsup}-1}{\beta\text{minsup}}SW(S)+\dfrac{1}{\text{minsup}}, & SW(S)<\beta\end{cases}$$

则 $0\leqslant\rho\leqslant\dfrac{1}{\text{minsup}}$。

证明：对于序列 S，①当 $SW(S)\geqslant\beta$ 时，$0\leqslant\beta\leqslant\rho=SW(S)\leqslant 1$，显然成立；

②当 $SW(S)<\beta$ 时，由于 $\dfrac{\text{minsup}-1}{\beta\text{minsup}}<0$，所以 ρ 是关于 $SW(S)$的递减函数，故当 $SW(S)=0$ 时，ρ 取得最大值，即 $\rho_{\max}=\dfrac{1}{\text{minsup}}$；当 $SW(S)=\beta$ 时，ρ 取得最小值，即 $\rho_{\min}=1$，所以 $1<\rho\leqslant\dfrac{1}{\text{minsup}}$，综上所述，$0\leqslant\rho\leqslant\dfrac{1}{\text{minsup}}$，原命题得证。

3.2 序列模糊形式背景

自适应序列模式上的模糊形式背景与传统的模糊形式背景不同，它不仅需要表示对象与属性之间的隶属度，还需要能够表示序列中每个项目的位置信息，即项位集 IPos$S(i_j)$。

定义 6 在序列数据库 SDB 中，对于一个序列 $S=\langle s_1,s_2,\cdots,s_m\rangle$，设其包含的所有项目的集合为 $Item(S)$，则对于 $\forall i_j\in Item(S)$，称 $IW_{sr}(i_j)=\dfrac{IW(i_j)INum_S(i_j)}{\sum_{i\in Item(S)}IW(i)}$ 是项目 i_j 相对序列 S 中其它项目的权重，简称项目 i_j 的相对权重，其中 $j=1,2,\cdots,|Item(S)|$。

在定义 6 中，当一个序列 S 的长度比较大时，会造成序列 S 中每一个项目的相对权重比较小；并且当不同序列的同一个项目 m 与阈值 θ_m，φ_m 做比较时，不具有可比性；因此，需对序列中项目的相对权重进行标准化处理，如下。

定义 7 在序列数据库 SDB 中，对于一个序列 $S=\langle s_1,s_2,\cdots,s_m\rangle$，若对于 $\forall i_j\in Item(S)$，其相对权重为 $IW_{sr}(i_j)$，则称 $FIW_{sr}(i_j)=\dfrac{IW_{sr}(i_j)}{\max\limits_{1\leqslant k\leqslant|Item(S)|}(IW_{sr}(i_k))}$ 为项目 i_j 相对权重的标准权重，简称项目 i_j 的标准权重，其中 $j=1,2,\cdots,|Item(S)|$。

在项目标准权重与项位集的基础上，本文对模糊形式背景进行了扩展，以便表示权重序列模式。序列模糊形式背景(Sequence Fuzzy Formal Context，SFFC)是一个四元组 $SFFC=(G,M,R,IP)$，其中 G 代表序列的 id 集，即对象集；M 代表序列数据库 SDB 中所有项目的集合，即属性集，且 M 中的属性(项目)按照字典序排列；对于 $\forall g\in G$ 与 $\forall m\in M$，R 满足 $R(g,m)=FIW_{gr}(m)$，表示对于任意属于序列对象 g 的项目 m 相对序列 g 中其它项目的标准权重；IP 表示所有项目在每个序列中的项位集 $IPos\mathrm{S}(\mathrm{i_j})$，表 2 所列的模糊形式背景经过扩展后得到的序列模糊形式背景如表 3 所列。

表 3 序列模糊形式背景 $SFFC$

R	a	b	c	d	e	f	g
1	1{1,4}	0	0.6{1}	0.1{2}	0.6{3}	0.6{5}	0
2	0.9{2}	0.1{3}	0	0	1{1}	0	0
3	0.5{2}	0.1{2,3}	0.5{2}	0.1{3}	1{1,3}	0.5{4}	0
4	0.5{2}	0	0	0	0.5{3}	1{1,3}	0.1{4}
θ_m	0.3	0.1	0.4	0.1	0.5	0.3	0.3
φ_m	1.0	0.9	0.7	0.8	1.0	1.0	0.8

构造序列模糊形式背景 $SFFC$ 的算法如下。

算法 1 Procedure $SFFC(SDB,IWV,\theta,\varphi)$

输入：序列数据库 SDB，项目集的权重向量 IWV，每个属性的阈值 θ_m，φ_m

输出:序列模糊形式背景 SFFC

1:Begin

2: for k←1 to $|SDB|$ do

3: max←0

4: for j←1 to $|Item(S_k)|$ do

5: $IW[k][j]$←$(IW(i_j) * INums_{S_k}(i_j))/\sum_{i\in Item(S)} IW(i)$

6: if max<$IW[k][j]$ then

7: max←$IW[k][j]$

8: Endif

9: Endfor

10: for j←1 to $|Item(S_k)|$ do

11: $FIW[k][j]$←$IW[k][j]$/max

12: fuzzy_forcon$[k][j].iw$←$FIW[k][j]$

13: fuzzy_forcon$[k][j].ps$←$lPos_{S_k}(i_j)$

14: Endfor

15: Endfor

16:End

3.3 序列模糊概念

在模糊形式背景 $K=(G,M,R)$上,对于 $A\in P(G)$,$B\in P(M)$,模糊概念 $C=(A,B,\sigma,\lambda)$的外延 A 与内涵 B 应满足闭包运算 $A'=B,B'=A$。而在序列模糊形式背景 $SFFC=(G,M,R,IP)$上,也应该满足这两种闭包运算。

定义 8 在序列模糊形式背景 $SFFC=(G,M,R,IP)$上,任意给定 3 个序列 s,S_1,S_2 且对于 $\forall\alpha\in s,\theta\alpha\leqslant FIW_{s\tau}(\alpha)\leqslant\varphi_\alpha$;$\forall\delta\in S_1,\theta\delta\leqslant FIW_{S1\tau}(\delta)\leqslant\varphi_\delta$;$\forall\pi\in S_2,\theta\pi\leqslant FIW_{S2\tau}(\pi)\leqslant\varphi_\pi$;若 $s\subseteq S_1$ 且 $s\subseteq S_2$,则称序列 s 是序列 S_1 与 S_2 的公共子序列。

定义 9 在序列模糊形式背景 $SFFC=(G,M,R,IP)$上,设任意序列集 $Sset=\{id_{k_1},id_{k_2},\cdots,id_{k_t}\}\subseteq G$,其中 $\forall k_j<|G|(j=1,2,\cdots,t;t<|G|)$,$Sset$ 中所有元素的公共子序列集(Common Subsequence Set)记为 $ComSS(Sset)$;对于 $\forall subs\in ComSS(Sset)$的子序列,在 $ComSS(Sset)$中,$subs$ 所有的子序列与超序列集合称为 $subs$ 在 $Sset$ 上的扩减集,记为 $ExpRSset(subs)$。

在序列模糊形式背景 SFFC 上,定义 9 中,公共子序列集 $ComSS(Sset)$中的每个元素均已删除了其上不满足隶属度属性阈值区间$[\theta_m,\varphi_m]$的项目。如在表 3 中,对于$\langle(e)(a)\rangle\in ComSS(\{1,3\})$,则$\langle(e)(a)\rangle$中的每个项目 e,a 均满足对应的属性阈值区间;而 $ComSS(\{4\})$则为$\{\langle(f)(a)(ef)\rangle\}$,而不是$\{\langle(f)(a)(ef)(g)\rangle\}$,因为 $R(4,g)=0.1\notin[0.3,0.8]$。

定义 10 在序列模糊形式背景 $SFFC=(G,M,R,IP)$上,设 $\forall subs\in ComSS(Sset)$,对于一序列 $S_x\in ExpR_{Sset}(subs)$,若对于 $\forall S_y\in ExpR_{Sset}(subs)\backslash\{S_x\}$,都不存在 $S_x\subset S_y$,则称 S_x 是扩减集 $ExpRSset(subs)$上的最大公共子序列(Maximal Common Subsequence,简称 MaxCS);$ComSS(Sset)$上的所有 $MaxCS$ 构成的集合记为 $MCS(\{id_{k_1},id_{k_2},\cdots,id_{k_t}\})$。

例如,在表 3 的序列模糊形式背景 $SFFC=(G,M,R,IP)$上,由定义 10 可知,$MCS(\{1\})=\{\langle(ac)(d)(e)(a)(f)\rangle\}$,$MCS(\{1,3\})=\{\langle(e)(a)(f)\rangle,\langle(ac)(d)(f)\rangle,\langle(ac)(e)(f)\rangle\}$等。

定理 1 在序列数据库 SDB 中,设所有序列模式的集合为 SeqP(SDB),对于 $\forall s\in SeqP(SDB)$,在序列模糊形式背景 SFFC 上必存在一集合 $MCS(\{id_{k_1},id_{k_2},\cdots,id_{k_t}\})$,当每个项目属性 m 的 $\theta_m=0$ 且 $\varphi_m=1$ 时,使得在其上 $\exists S\in MCS(id_{k_1},id_{k_2},\cdots,id_{k_t})$且 $s\subseteq S$,其中 $\forall k_j<|G|(j=1,2,\cdots,t;t<|G|)$。

证明:当每个项目属性 m 的 $\theta_m=0$ 且 $\varphi_m=1$ 时,假设在序列模糊形式背景 SFFC 上不存在一集合 $MCS(\{id_{k_1},id_{k_2},\cdots,id_{k_t}\})$,其中 $\forall k_j<|G|(j=1,2,\cdots,t;t<|G|)$。对于 $\forall s\in SeqP(SDB)$,序列模式 s 在序列数据库 SDB 中必有支持它的顾客序列,设这些顾客序列的 id 分别为 $id_{k_1},id_{k_2},\cdots,id_{k_t}$。由假设可知,当每个项目属性 m 的 $\theta_m=0$ 且 $\varphi_m=1$ 时,序列中的任一个项目的标准权重均在阈值区间$[\theta_m,\varphi_m]$之内,都不会被过滤删除掉,因此,在 SFFC 上不存在 $MCS(\{id_{k_1},id_{k_2},\cdots,id_{k_t}\})$这样的集合,这显然与定义 9 和定义 10 矛盾。故在 SFFC 上必存在 $MCS(\{id_{k_1},id_{k_2},\cdots,id_{k_t}\})$,所以 $ExpR_{Sset}(s)\subseteq MCS(\{id_{k_1},id_{k_2},\cdots,id_{k_t}\})$,故存在 $S\in ExpR_{Sset}(s)$且 $S\in MCS(\{id_{k_1},id_{k_2},\cdots,id_{k_t}\})$,使得 $s\subseteq S$。

根据定理 1 可知,当每个项目属性 m 的 $\theta_m=0$ 且 $\varphi_m=1$ 时,序列数据库 SDB 中所有序列模式必包含于序列模糊形式背景 SFFC 上的最大公共子序列集中,从而保证了算法可以挖掘传统的序列模式。另外,由于对阈值 θ_m 与 φ_m 的指定不同,算法还可以快速地挖掘特殊的序列模式,如满足不同重要度的序列模式;并且在序列模糊形式背景 SFFC 上最大公共子序列包含了其扩减集上所有序列模式的信息,因此,在 SFFC 上挖掘自适应序列模式时可以降低序列模式的冗余性,提高算法的效率。

在序列模糊形式背景 SFFC 上,构建两个序列 S_1 与 S_2 的最大公共子序列集 TMCS 的算法 Max*ComSS* 如下。

算法 2 Procedure Max*ComSS*(S_1,S_2)

输入:SFFC 上的两个序列 S_1,S_2

输出:最大公共子序列集 $TMCS$

1:Begin

2:for each $\alpha\in Item(S_1)$ do

3: if $FIW_{s1r}(\alpha)\notin[\theta_\alpha,\varphi_\alpha]$ then

4: Delete α in S_1

5: Endif

6: Endfor

7: for each $\alpha\in Item(S_2)$ do

8: if $FIW_{s2r}(\alpha)\notin[\theta_\alpha,\varphi_\alpha]$ then

9: Delete α in S_2

10: Endif

11:Endfor

12:for each $s\subseteq S_1$ do

13: if $s\subseteq S_2$ then

14: $ComSS(\{S_1,S_2\})$←s

15: Endif

16: Endfor

17: for each $subs\in ComSS(\{S_1,S_2\})$ do

18: for each $rests\in ComSS(\{S_1,S_2\})\backslash\{subs\}$ do

19: if $rests\subseteq subs$ or $subs\subseteq rests$ then

20: $ExpR(subs)$←$rests$

21: Endif

22: Endfor

23: for each $s\in ExpR(subs)$ do

24: for each $x\in ExpR(subs)\backslash\{s\}$ do

25: if $s\subseteq x$ then

```
26:        ExpR(subs)←ExpR(subs)\{s}
27:            Break
28:          Endif
29:        Endfor
30:      Endfor
31:      TMCS←ExpR(subs)
32:  Endfor
33:End
```

定义 11 在序列模糊形式背景 $SFFC=(G,M,R,IP)$ 上，Galois 闭包运算定义如下：

①对于 $\forall A\subseteq P(G)$，$\Delta(A)=\{s\mid s\in MCS(A)\wedge(\forall g\in A,\forall m\in Item(s),\theta_m\leqslant R(g,m)\leqslant\varphi_m)\}$；

②对于序列集 T，$\Psi(T)=\{g\mid g\in G\wedge(\forall s\in T,s\subseteq g)\wedge(\forall m\in Item(s),\theta_m\leqslant R(g,m)\leqslant\varphi_m)\}$；

其中，$MCS(A)$ 代表序列对象 A 的最大公共子序列集，$Item(s)$ 代表序列 s 中所有项目的集合。

在定义 11 中，若 $A=\{1,3\}$，则 $\Delta(A)=\{\langle(e)(a)(f)\rangle,\langle(ac)(d)(f)\rangle,\langle(ac)(e)(f)\rangle\}$；若 $T=\{\langle(a)\rangle,\langle(e)(a)\rangle\}$，则 $\Psi(T)=\{1,2,3\}$。

定义 12 在序列模糊形式背景 $SFFC=(G,M,R,IP)$ 上，若一个四元组 $C=(A,T,\sigma,\lambda)$ 满足以下 3 个条件：

①序列对象 $A\in P(G)$，且 T 为 SFFC 上的序列集；

②$\Delta(A)=T,\Psi(T)=A$；

③模糊参数 σ 定义形式如下：$\sigma=\frac{1}{|T|}\sum_{s\in T}SW(s)$，模糊参数 λ 定义形式如下：$\lambda=\sqrt{\frac{\sum_{s\in T}(SW(s)-\sigma)^2}{|T|}}$，其中 $|T|\neq 0$；

则称 $C=(A,T,\sigma,\lambda)$ 是 SFFC 上的一个序列模糊概念。其中，A 称作序列模糊概念 C 的外延，T 称作序列模糊概念 C 的内涵；σ 表示序列模糊概念 C 的外延对应于最大公共子序列的平均隶属度，体现了 C 概念聚集子序列的程度；λ 是 C 概念中各内涵偏离平均值的平均程度，体现了 C 概念中内涵的发散程度。

定理 2 在序列模糊形式背景 $SFFC=(G,M,R,IP)$ 上，若对于 $\forall A\in P(G)$，一个四元组 $C=(\Psi(\Delta(A)),\Delta(A),\sigma,\lambda)$，则 C 必为一个序列模糊概念。

根据定义 11 中 Galois 闭包运算的定义以及定义 12，定理 2 显然是成立的。

4 序列模糊概念格及其构造算法

定义 13 在序列模糊形式背景 SFFC 上，任意给定两个序列模糊概念 $C_1=(A_1,T_1,\sigma_1,\lambda_1)$ 与 $C_2=(A_2,T_2,\sigma_2,\lambda_2)$，若两者满足对于 $\forall s_2\in T_2$，总存在 $s_1\in T_1$，使得 $s_2\subseteq s_1$ 或 $A_1\subseteq A_2$，则称 C_1 是 C_2 的序列模糊子概念，C_2 是 C_1 的序列模糊超概念。C_1 与 C_2 之间的这种偏序关系，使用符号“$\leqslant$”表示，记为 $C_1\leqslant C_2$。

在序列模糊形式背景 SFFC 上，由所有序列模糊概念的超概念-子概念的偏序关系所诱导出的格结构称为序列模糊概念格(Sequence Fuzzy Concept Lattice)，记为 FL(SFFC)。

在传统形式背景上，概念格构造算法主要分为两类：批处理算法[9,10]和渐进式构造算法[11,12]。实验证明，相对于批处理算法，渐进式生成算法具有更多的优越性。本文在模糊概念格渐进式构造算法思想[8]的基础上，提出了一种在序列模糊形式背景上构造序列模糊格的算法(Algorithm of constructing Sequence Fuzzy Concept Lattice from SFFC，简记为 SeqFuzCL 算法)。

算法 3 Procedure $SeqFuzCL(SFFC,IWV,\theta_m,\varphi_m)$

输入：序列模糊形式背景 $SFFC$，权重向量 IWV，阈值 θ_m,φ_m

输出：序列模糊概念格 $FL(SFFC)$

```
1:Begin
2:  for each object x*∈G do
3:    if FL(SFFC)=Φ then
4:      Add pair(x*,Δ(x*)) to FL(SFFC), and calculate fuzzy
        parameters σ,λ
5:    Else
6:      for each node C=(A,T,σ,λ)∈FL(SFFC) do {ordering ac-
        cording to |A| ascending}
7:        Scount←0,flag←False
8:        for each s∈T do
9:          if s⊆Δ(x*) then scount←scount+1 endif
10:       Endfor
11:   if scount=|T| then flag←True endif
12:       if flag then {modified notion}
13:         A←A∪{x*}
14:     Update fuzzy parameters σ,λ in C
15:           Add C to the set newcon {initialize newcon to be Φ}
16:         Else
17:           for each s∈T do
18:             newint←MaxComSS(s,Δ(x*)) {newint is a se-
                quence set}
19:           Endfor
20:           for each s∈newint do
21:             for each y∈newint\{s} do
22:               If s⊆y then
23:                 newint←newint\{s}
24:                 break
25:               Endif
26:             Endfor
27:           Endfor
28:           if ¬∃C1∈newcon satisfying intent(C1)=newint then
29:             Create new pair Cn←(A∪{x*},newint)
30:               Calculate fuzzy parameters σ,λ
31:             Add Cn to newcon
32:             Add edge Cn=(A∪{x*},newint,σ,λ)→C
33:             for each Ca in newcon do
34:               scount←0
35:             for each s∈intent(Ca) do
36:                   if exist y∈newint and s⊂y then
37:                   scount←scount+1
38:                 Endif
39:               Endfor
40:             If scount=|intent(Ca)| then
41:               parent←True
42:               for each Cc child node of Ca do
43:             if intent(Cc)⊂newint then parent←False;break endif
44:               Endfor
45:               if parent then
46:                 if Ca is a parent node of C then
47:                   Delete edge Ca→C
```

```
48:                    Endif
49:                  Endif
50:                Endif
51:              Endfor
52:            Endif
53:          Endif
54:        Endfor
55:      Endif
56:  Endfor
57:  Input sequence fuzzy lattice FL(SFFC)
58:End
```

利用算法 SeqFuzCL，表 3 所列的序列模糊形式背景所对应的序列模糊概念格如图 1 所示。

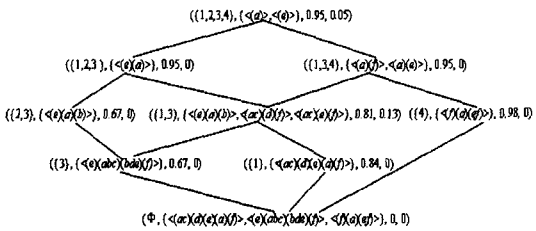

图 1　表 3 所对应的序列模糊概念格 SeqFCL

5　实验分析

为了验证本文提出的 SeqFuzCL 算法的有效性，我们利用 Python2.5.1 实现了 SeqFuzCL 算法，并在随机生成的序列数据集上进行了实验。实验的运行环境是：Linux 操作系统，256MB 内存，CPU2.8GHz。

图 2 和图 3 中项目参数 $|M|$ 为 20，每个序列对象平均拥有项目的概率为 0.5，每个项目属性 m 的阈值 θ_m，φ_m 与项目权重均在区间[0,1]内随机产生。

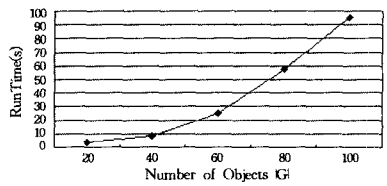

图 2　构格算法 SeqFuzCL 的时间复杂度

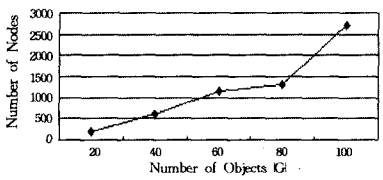

图 3　序列对象数与生成格节点数的关系

图 2 显示了在构造序列模糊概念格时算法 SeqFuzCL 随着序列对象数的增加其时间复杂度的变化情况。通过图 2 可以看出，算法 SeqFuzCL 的时间复杂度和经典的 Godin 算法基本一致。

通过图 3 可以看出随着序列对象数的增加，序列模糊概念格节点的变化情况。当序列对象数比较少时，格节点的数目变化比较平缓，但随着序列对象数的增大，由于序列对象组合的剧增，格节点的数目也较大。

图 4 和图 5 中项目参数 $|M|$ 为 20，每个序列对象平均拥有项目的概率为 0.4，项目权重在区间[0,1]内随机产生，其中 $\theta=0$ 表示每个属性 m 的阈值 θ_m 均为 0，$\varphi=1$ 表示每个属性 m 的阈值 φ_m 均为 1；$\theta=0.3$，$\varphi=0.9$ 与 $\theta=0.5$，$\varphi=0.6$ 表示的含义与上述相同。

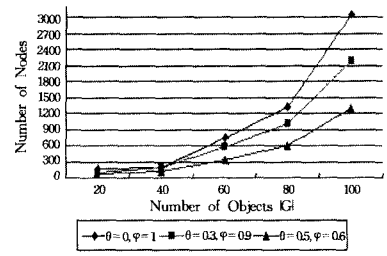

图 4　对象数与节点数在不同 θ_m，φ_m 下的关系

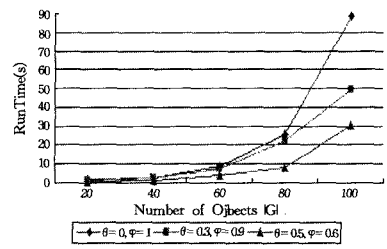

图 5　不同 θ_m，φ_m 下的 SeqFuzCL 的时间复杂度

图 4 表现了在不同序列对象数下每个属性阈值 θ_m，φ_m 的变化与格节点个数变化的关系。通过图 4 可以看出在相同的序列对象个数下，阈值 θ_m，φ_m 的取值不同，得到的序列模糊概念个数也不同。随着阈值 θ_m 与 φ_m 的接近，获得的序列模糊概念数也会越少，既可满足用户的需求，也可减少冗余的序列模糊概念，从而节省大量的存储空间。

图 5 表现了在不同序列对象数下每个属性阈值 θ_m，φ_m 的变化与构格算法 SeqFuzCL 的运行时间的关系。通过图 5 可以看出在相同的序列对象个数下，阈值 θ_m，φ_m 的取值不同，构格算法 SeqFuzCL 的运行时间也不同；阈值 θ_m 与 φ_m 越接近，算法的运行时间也会越少，这是由于算法 SeqFuzCL 在构格的过程中，通过阈值 θ_m 与 φ_m 已过滤掉许多冗余的项目属性，从而减少了序列长度，同时也减少了冗余的序列模糊概念。因此，算法 SeqFuzCL 在不同的用户需求下可以节省大量的时间。

结束语　本文定义了一种序列模糊概念格模型，并给出了该模型的渐进式构造算法 SeqFuzCL。该模型能够有效地组织自适应序列模式，通过设置不同的参数、阈值来满足用户不同的需求，通过实验表明算法在时空复杂度上具有良好的性能，为进一步挖掘自适应序列模式提供了理论支持。如何并行或分布式地构造序列模糊概念格以及从中挖掘自适应序列模式将是下一步的研究工作。

参 考 文 献

[1] Agrawal R, Srikant R. Mining Sequential Pattern[C]// Proc. of the 11st Int. Conf. on Data Engineering. Taipei, March 1995: 3-14

[2] Srikant R, Agrawal R. Mining Sequential Patterns: Generalizations and Performance Improvements[C]// Proc. of the Fifth Int. Conf. on Extending Database Technology (EDBT). Avignon, France, March 1996: 3-17

[3] Zak M. SPADE: An Efficient Algorithm for Mining Frequent Sequences[J]. Machine Learning, 2001, 42: 31-60

[4] Han J, Pei J, Mortazavi-Asi B, et al. FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining[C]// Proc. 2000 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'

00). Boston, Massachusetts, 2000: 355-359

[5] Pei J, Han J, Mortazavi-Asl B, et al. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth[C]// Proc. of 17th International Conf. on Data Engineering (ICDE'01). Heidelberg, Germany, 2001: 215-224

[6] Yun U. A New Framework for Detecting Weighted Sequential Patterns in Large Sequence Databases[J]. Knowledge-based systems, 2008, 21(2): 110-122

[7] Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts[J]. Ordered Sets, 1982: 445-470

[8] 刘宗田，强宇，周文，等. 一种模糊概念格模型及其渐进式构造算法[J]. 计算机学报，2007，30(2)：184-188

[9] Bordat J P. Calcul pratique du treillis de galois d'unc correspondence[J]. Math. Et Sci. Humaines, 1986, 96: 31-47

[10] Bettini C, Wang X S, Jajodia S. Mining Temporal Relationships with Multiple Granularities in Time Sequences[J]. Data Eng. Bull, 1998, 21: 32-38

[11] Godin R, Missaoui R, Alaoui H. Incremental concept formation algorithms based on Galois lattices[J]. Computational Intelligence, 1995, 11(2): 246-267

[12] 谢志鹏，刘宗田. 概念格的快速渐进式构造算法[J]. 计算机学报，2002，25(5)：490-495

(上接第 205 页)

提取系统中加入了投票功能，对用户反馈的信息加以收集。用户根据用户信息保护下的网络学习资源知识点内容自动提取平台的表现进行评分。评分标准：1～100 分。

本文收集了 4 个学习网站在 120 天中的用户评分信息，对比了用户信息保护下的网络学习资源知识点内容自动提取系统使用前后的平均分数，结果如表 2 所列。

表 2　4 个学习网站在系统使用前后的用户评分比较

	site1	site2	site3	site4
Before	63	67	71	61
After	76	79	84	83

分析

如图 3 所示，使用用户信息保护下的网络学习资源知识点内容自动提取系统后，4 家学习网站的用户评分较之前的均有提升，site4 增幅最大，达 36%；site2 增幅最小，为 17.91%；4 家网站用户评分增幅平均提高达 23.23%。真实数据集显示的实验结果证明了本文提出的方法的有效性。但是本文也注意到实验中用户评分的提高可能会受其他因素的影响。例如，site4 在使用本系统前，用户评价最低，但之后评分的迅速提高可能是受到其在实验过程中的广告促销的带动。实验中，至少 3 家网站的数据清晰地显示了用户信息保护下的网络学习资源知识点内容自动提取系统对加强用户信息保护及知识点自动提取方面具有显著的效果。本文用真实数据集支持了实验结论。

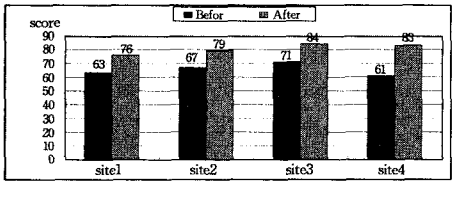

图 3　4 家学习网站用户评分比较图

结束语　本文重点研究用户信息保护下的知识点内容自动提取技术，通过计算用户信息保护状态、影响因素观察项、初始概率、转移概率、发射概率，构造了 HMM 模型来预测用户信息是否被侵犯，实现了保护状态失效时数据处理程序的自动退出。在用户信息得到保护的前提下，通过知识点内容将大量的课程信息用简洁、准确的形式呈现给用户。为了消除课程特有的冗余信息，本文提出了一种基于聚类算法的知识点自动提取模型，其通过信息融合技术合并提取出的知识点内容。

今后的研究主要围绕以下方面展开：

在协同式学习中，建立用户之间单向、双向以及多向等复杂信任关系的评价图。在保护用户隐私的条件下，通过获取用户社会网络联系信息来构建基于信任链的协同式知识点自动提取系统。

在原 OPTICS 算法中，有序队列总是按照可达距离升序排列，所以，算法总选择可达距离最小的点进行处理。这样会导致算法总是沿着稠密的区域进行扩张，只有处理完当前的稠密区域才会跳到下一个稠密区域。而跳入新的稠密区后，原先未处理的稀疏节点将无法得到处理，一直到所有稠密区域全部找出后才会处理稀疏节点。这种处理节点的方式会导致聚类结果的偏差，不能完全反映数据空间的真实结构。为了提高聚类的准确性，需要给出一种处理稀疏节点的方法：拟先将所有节点使用 OPTICS 算法进行聚类，找出基本的聚类簇；然后再对稀疏节点进行处理，计算其到每个簇的相似度，并将其加入相似度最大的簇进行聚类。将稀疏结点恰当地分到每个类簇中去，使聚类结果比原 OPTICS 算法能更加准确地反映学习资源文档集合空间的结构。

参 考 文 献

[1] Wu Z. Dynamic Trust Establishment with Privacy Protection for Web Services [A]// Proceedings of the IEEE International Conference on Web Services[C]. ACM Press, 2005: 811-812

[2] Liu Shuang, Liu Fang, Yu C, et al. An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases[A] // Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval[C]. ACM Press, 2004: 266-227

[3] Liu Shuang, Yu C, Meng Wei-yi. WordSense Disambiguation in Queries [A]// Proceedings of the 14th ACM International Conference on Information and Knowledge Management[C]. ACM-Press, 2005: 525-532

[4] Ankerst M, Breunig M M, Kriegel H-P, et al. OPTICS: ordering points to identify the clustering structure. International Conference on Management of Data [A]// Proceedings of the 1999 ACM SIGMOD international conference on Management of data [C]. ACM Press, 1999: 49-60

[5] Barzilay R, Elhadad N, McKeown K. Inferring Strategies for Sentence Ordering in Multidocument News Summarization[J]. Journal of Artifical Intelligence Research (JAIR), 2002, 17: 35-55

[6] Lin Chinyew, Hovy E. Automated Multi-document Summarization in NeATS[A]// Proceedings of the Human Language Technology Conference[C]. 2002