Trường Đại học Khoa học Tự nhiên
Khoa Công nghệ Thông tin

TÀI LIỆU LÝ THUYẾT KHAI THÁC WEB

Chủ đề 2

KHAI THÁC LUẬT KẾT HỢP & MẪU TUẦN TỰ (PHẦN 3)

Giảng viên: ThS. Nguyễn Ngọc Thảo
Email: nnthao@fit.hcmus.edu.vn

HKII/2011-2012

NỘI DUNG

- Khái niệm cơ bản về mẫu tuần tự
- Các thuật toán khai thác mẫu tuần tự
  - GSP
  - PrefixSpan
- Phát sinh luật từ mẫu tuần tự

KHÁI NIỆM CƠ BẢN VỀ MẠCH TUẦN TỰ

GIỚI THIỆU MẪU TUẦN TỰ

- Khai thác luật kết hợp không xét thứ tự của các giao dịch.
- Trong nhiều ứng dụng, thứ tự giao dịch là quan trọng $\Rightarrow$ mẫu tuần tự.
  - Phân tích giỏ bán hàng: khách hàng mua giường rồi sẽ mua gì tiếp theo, ví dụ ra trải giường.
  - Khai thác Web: tìm mẫu duyệt một Web site từ chuỗi các trang đã viếng thăm của người dùng.
  - Khai thác văn bản: xét thứ tự các từ trong câu để tìm mẫu ngôn ngữ.

ĐỊNH NGHĨA MẪU TUẦN TỰ

- Gọi $I = \{i_1, i_2, ..., i_m\}$ là tập hợp hạng mục.
- *Chuỗi là một danh sách các tập hạng mục được sắp xếp thứ tự*, kí hiệu $s = \langle a_1 a_2 ... a_r \rangle$.
  - a_i là tập hạng mục, gọi là phần tử của s.
- Phần tử của chuỗi (tập hạng mục) là tập $\{x_1, x_2, ..., x_k\}$, với $x_j \in I$ là một hạng mục.
  - Hạng mục trong một phần tử chuỗi phải sắp xếp theo thứ tự từ điển.

ĐỊNH NGHĨA MẪU TUẦN TỰ

- Một hạng mục xuất hiện một lần trong một phần tử chuỗi nhưng có thể xuất hiện trong nhiều phần tử khác nhau.
- Kích thước của chuỗi là số phần tử (tập hạng mục) có trong chuỗi.
- Độ dài của chuỗi là số hạng mục trong chuỗi. Chuỗi có độ dài *k* gọi là *k*-sequence.
  - Nếu một hạng mục xuất hiện trong nhiều phần tử, mỗi lần xuất hiện đều tính vào *k*.

ĐỊNH NGHĨA MẪU TUẦN TỰ

- Chuỗi $s_1 = \langle a_1 a_2 ... a_r \rangle$ là chuỗi con của chuỗi $s_2 = \langle b_1 b_2 ... b_v \rangle$ nếu tồn tại các số nguyên $1 \le j_1 < j_2 < ... < j_{r-1} < j_r \le v$ sao cho $a_1 \subseteq b_{j1}$, $a_2 \subseteq b_{j2}$, ..., $a_r \subseteq b_{jr}$.
- s_2 là chuỗi cha của s_1 hay s_2 chứa s_1.
- Ví dụ:
  - Cho I = {1, 2, 3, 4, 5, 6, 7, 8, 9}
  - ⟨{3} {4, 5} {8}⟩ là chuỗi con của ⟨{6} {3, 7} {9} {4, 5, 8} {3, 8}⟩, kích thước là 3, độ dài là 4.
  - ⟨{3} {8}⟩ không chứa trong ⟨{3, 8}⟩ và ngược lại.

KHAI THÁC MẪU TUẦN TỰ

- Cho tập *S* gồm các chuỗi dữ liệu đầu vào (hay cơ sở dữ liệu chuỗi).
- Bài toán khai thác mẫu tuần tự tìm mọi chuỗi thỏa giá trị độ hỗ trợ tối thiểu do người dùng định trước, gọi là chuỗi phổ biến hay mẫu tuần tự.
- Độ phổ biến của một chuỗi là tỉ lệ số chuỗi dữ liệu trong S chứa chuỗi.

CƠ SỞ DỮ LIỆU GIAO DỊCH

- Cơ sở dữ liệu giao dịch đã sắp theo ID khách hàng và thời gian giao dịch.

Customer ID	Transaction Time	Transaction (items bought)
1	July 20, 2005	30
1	July 25, 2005	90
2	July 9, 2005	10, 20
2	July 14, 2005	30
2	July 20, 2005	10, 40, 60, 70
3	July 25, 2005	30, 50, 70, 80
4	July 25, 2005	30
4	July 29, 2005	30, 40, 70, 80
4	August 2, 2005	90
5	July 12, 2005	90

CƠ SỞ DỮ LIỆU CHUỖI

- Chuỗi là danh sách các giao dịch của một khách hàng, sắp xếp theo thứ tự thời gian.
- Mỗi giao dịch là tập các món hàng mà khách đã mua tại một thời điểm giao dịch.

Customer ID	Data Sequence
1	⟨{30} {90}⟩
2	⟨{10, 20} {30} {10, 40, 60, 70}⟩
3	⟨{30, 50, 70, 80}⟩
4	⟨{30} {30, 40, 70, 80} {90}⟩
5	⟨{90}⟩

TẬP HỢP MẪU TUẦN TỰ

- Các mẫu tuần tự thu được khi áp dụng ngưỡng độ hỗ trợ tối thiểu 25%, tức là 2 khách hàng.

	Sequential Patterns with Support ≥ 25%
1-sequences	⟨{30}⟩, ⟨{40}⟩, ⟨{70}⟩, ⟨{80}⟩, ⟨{90}⟩
2-sequences	⟨{30} {40}⟩, ⟨{30} {70}⟩, ⟨{30} {90}⟩, ⟨{30, 70}⟩, ⟨{30, 80}⟩, ⟨{40, 70}⟩, ⟨{70, 80}⟩
3-sequences	⟨{30} {40, 70}⟩, ⟨{30, 70, 80}⟩

KHAI THÁC MẪU TUẦN TỰ GSP

THUẬT TOÁN GSP

- GSP hoạt động tương tự như Apriori, khác biệt chủ yếu ở hàm phát sinh ứng viên.

Algorithm GSP(S)

1 $C_1 \leftarrow$ init-pass(S); // the first pass over S
2 $F_1 \leftarrow \{\langle\{f\}\rangle | f \in C_1, f.count/n \geq minsup\}$; // n is the number of sequences in S
3 **for** ($k = 2$; $F_{k-1} \neq \emptyset$; k++) **do** // subsequent passes over S
4 $C_k \leftarrow$ candidate-gen-SPM(F_{k-1});
5 **for** each data sequence $s \in S$ **do** // scan the data once
6 **for** each candidate $c \in C_k$ **do**
7 **if** c is contained in s **then**
8 $c.count$++; // increment the support count
9 **endfor**
10 **endfor**
11 $F_k \leftarrow \{c \in C_k \mid c.count/n \geq minsup\}$
12 **endfor**
13 **return** $F \leftarrow \bigcup_k F_k$;

HÀM CANDIDATE-GEN-SPM

- **Funtion** candidate-gen-SPM(F_{k-1}) //SPM: Sequential Pattern Mining

  1. Join step: chọn cặp ứng viên từ F_{k-1}. Gọi s'_1 là chuỗi con khi bỏ hạng mục đầu tiên của s_1, s'_2 là chuỗi con khi bỏ hạng mục cuối cùng của s_2. Nếu s'_1 giống s'_2 thì tạo ứng viên $c = s_1$ nối với hạng mục cuối cùng của s_2.
     - Hạng mục thêm vào tạo thành phần tử riêng nếu như nó đã là phần tử riêng rẽ trong s_2, hoặc
     - Hạng mục thêm vào là một phần của phần tử cuối cùng trong s_1.

     Khi kết F_1 và F_1, ví dụ $\langle\{x\}\rangle$ và $\langle\{y\}\rangle$, cần tạo ứng viên cả $\langle\{x, y\}\rangle$ và $\langle\{x\}\{y\}\rangle$. Lưu ý, x và y trong $\{x, y\}$ có thứ tự.
  2. Prune step: ứng viên bị loại bỏ nếu tồn tại một chuỗi con độ dài (k-1) không phổ biến.

VÍ DỤ PHÁT SINH ỨNG VIÊN

- Bước gia nhập
  - ⟨{1, 2} {4}⟩ kết ⟨{2} {4, 5}⟩ tạo thành ⟨{1, 2} {4, 5}⟩
  - ⟨{1, 2} {4}⟩ kết ⟨{2} {4} {6}⟩ tạo thành ⟨{1, 2} {4} {6}⟩
  - ⟨{1} {4, 5}⟩ không kết với chuỗi nào vì không có chuỗi dạng ⟨{4, 5} {x}⟩ hoặc ⟨{4, 5, x}⟩.
- Bước tỉa nhánh
  - ⟨{1, 2} {4} {6}⟩

  bị loại bỏ. Vì sao?

Frequent 3-sequences	Candidate 4-sequences	
	after joining	after pruning
⟨{1, 2} {4}⟩	⟨{1, 2} {4, 5}⟩	⟨{1, 2} {4, 5}⟩
⟨{1, 2} {5}⟩	⟨{1, 2} {4} {6}⟩	
⟨{1} {4, 5}⟩		
⟨{1, 4} {6}⟩		
⟨{2} {4, 5}⟩		
⟨{2} {4} {6}⟩		

SỬ DỤNG NHIỀU MINSUP

- Tác vụ: khai thác các câu so sánh như "*the picture quality of camera X is better than that of camera Y*" từ những nhận xét sản phẩm, bài post trên forum và blog.
- Mục tiêu: phát hiện các mẫu ngôn ngữ liên quan đến tập các từ khóa so sánh, ví dụ better, more, less, ahead, win, superior,...
  - Một số từ (more, better) xuất hiện thường xyên hơn những từ khác (win, ahead,...).
  - Sử dụng minsup đơn sẽ không phù hợp nữa.

SỬ DỤNG NHIỀU MINSUP

- Gọi MIS(i) là giá trị MIS của hạng mục i.
- Độ hỗ trợ tối thiểu của mẫu tuần tự P là

  $\text{minsup}(P) = \min(\text{MIS}(i_1), \text{MIS}(i_2), ..., \text{MIS}(i_r))$
  - với i_j là hạng mục trong P.

THUẬT TOÁN MS-GSP

- MS-GSP tổng quát hóa thuật toán GSP, hoạt động tương tự MS-Apriori.
- Các điểm khác biệt
  - Hàm level2-candidate-gen-SPM() được thiết kế dựa trên level2-candidate-gen của MS-Apriori và join step của GSP.
  - Hàm MScandidate-gen-SPM() hoàn toàn khác (xem thêm trong tài liệu tham khảo).
  - Hạng mục có MIS nhỏ nhất có thể xuất hiện ở vị trí bất kỳ trong chuỗi, thay vì vị trí đầu tiên.
- Có thể tích hợp ràng buộc hiệu số hỗ trợ vào thuật toán MS-GSP.

THUẬT TOÁN MS-GSP

Algorithm MS-GSP(S, MS) // MS stores all MIS values
1 $M \leftarrow$ sort(I, MS); // according to MIS(i)'s stored in MS
2 $L \leftarrow$ init-pass(M, S); // make the first pass over S
3 $F_1 \leftarrow \{\langle\{l\}\rangle \mid l \in L, l.count/n \geq \text{MIS}(l)\}$; // n is the size of S
4 **for** ($k = 2$; $F_{k-1} \neq \emptyset$; k++) **do**
5 **if** $k = 2$ **then**
6 $C_k \leftarrow$ level2-candidate-gen-SPM(L)
7 **else** $C_k \leftarrow$ MScandidate-gen-SPM(F_{k-1})
8 **endif**
9 **for** each data sequence $s \in S$ **do**
10 **for** each candidate $c \in C_k$ **do**
11 **if** c is contained in s **then**
12 $c.count$++
13 **if** c' is contained in s, where c' is c after an occurrence of $c.minMISItem$ is removed from c **then**
14 $c.rest.count$++ // $c.rest$: c without $c.minMISItem$
15 **endfor**
16 **endfor**
17 $F_k \leftarrow \{c \in C_k \mid c.count/n \geq MIS(c.minMISItem)\}$
18 **endfor**
19 **return** $F \leftarrow \bigcup_k F_k$;

KHAI THÁC MẪU TUẦN TỰ PREFIX-SPAN

GIỚI THIỆU THUẬT TOÁN

- PrefixSpan khai thác mẫu tuần tự bằng tìm kiếm theo chiều sâu.
- Không cần phát sinh ứng viên.

THUẬT TOÁN PREFIX-SPAN

- Xét cơ sở dữ liệu chuỗi với minsup = 25%.

Customer ID	Data Sequence
1	⟨{30} {90}⟩
2	⟨{10, 20} {30} {10, 40, 60, 70}⟩
3	⟨{30, 50, 70, 80}⟩
4	⟨{30} {30, 40, 70, 80} {90}⟩
5	⟨{90}⟩

THUẬT TOÁN PREFIX-SPAN

- Đầu tiên, sắp xếp các hạng mục trong mỗi phần tử của chuỗi.
- Duyệt dữ liệu một lần để tìm mọi hạng mục phổ biến (30, 40, 70, 80 và 90)

  $\Rightarrow$ Mẫu tuần tự độ dài 1 là $\langle\{30\}\rangle$, $\langle\{40\}\rangle$, $\langle\{70\}\rangle$, $\langle\{80\}\rangle$ và $\langle\{90\}\rangle$.

THUẬT TOÁN PREFIX-SPAN

- Toàn bộ mẫu tuần tự có thể chia thành 5 tập con không giao nhau.
  - Tập con có tiền tố (mở đầu bằng) ⟨{30}⟩.
  - Tập con có tiền tố (mở đầu bằng) ⟨{40}⟩.
  - .....
  - Tập con có tiền tố (mở đầu bằng) ⟨{90}⟩.

- Xét các bước tìm mẫu tuần tự có tiền tố là⟨{30}⟩.

THUẬT TOÁN PREFIX-SPAN

- Xây dựng cơ sở dữ liệu chiếu-⟨{30}⟩
  - Nếu một chuỗi chứa hạng mục **30** thì phần chuỗi sau **30** đầu tiên được trích ra thành một chuỗi trong cơ sở dữ liệu chiếu.
  - Loại bỏ mọi hạng mục không phổ biến.

Customer ID	Data Sequence	⟨{30}⟩-projected database
1	⟨{30} {90}⟩	⟨{90}⟩
2	⟨{10, 20} {30} {10, 40, 60, 70}⟩	⟨{40, 70}⟩
3	⟨{30, 50, 70, 80}⟩	⟨{_, 70, 80}⟩
4	⟨{30} {30, 40, 70, 80} {90}⟩	⟨{30, 40, 70, 80} {90}⟩
5	⟨{90}⟩	

THUẬT TOÁN PREFIX-SPAN

- Xây dựng cơ sở dữ liệu chiếu-⟨{30}⟩
  - Nếu một chuỗi chứa hạng mục **30** thì phần chuỗi sau **30** đầu tiên được trích ra thành một chuỗi trong cơ sở dữ liệu chiếu.
  - Loại bỏ mọi hạng mục không phổ biến.

Custo…	Data Sequence	⟨{30}⟩-projected database
		⟨{90}⟩
	70}⟩	⟨{40, 70}⟩
3	⟨{30, 50, 70, 80}⟩	⟨{_, 70, 80}⟩
4	⟨{30} {30, 40, 70, 80} {90}⟩	⟨{30, 40, 70, 80} {90}⟩
5	⟨{90}⟩	

Xảy ra khi các hạng mục trong tập hạng mục cuối cùng của tiền tố được chứa trong một phần tử nhiều hạng mục của chuỗi.

THUẬT TOÁN PREFIX-SPAN

- PrefixSpan tìm cách mở rộng một hạng mục vào tiền tố. Có hai trường hợp:
  1. Mẫu phổ biến có dạng ⟨{30, x}⟩: sử dụng khuôn {_, x} và {30, x} so khớp mọi chuỗi đã chiếu để tính đếm hỗ trợ cho x.
     - Nếu có nhiều điểm khớp trong một chuỗi, tính là 1 lần.
     - Một cách tổng quát, khuôn thứ hai ({30, x}) sử dụng toàn bộ tập hạng mục cuối thay vì chỉ hạng mục cuối.
  2. Mẫu phổ biến có dạng ⟨{30} {x}⟩: x là hạng mục phổ biến trong CSDL chiếu và không nằm cùng tập hạng mục với hạng mục cuối của tiền tố.

THUẬT TOÁN PREFIX-SPAN

Customer ID	⟨{30}⟩-projected database
1	⟨{90}⟩
2	⟨{40, 70}⟩
3	⟨{_, 70, 80}⟩
4	⟨{30, 40, 70, 80} {90}⟩

- ⟨{30, 70}⟩:2 là chuỗi phổ biến.
  - Từ ⟨{_, 70, 80}⟩ và từ ⟨{30, 40, 70, 80} {90}⟩
- Tương tự ⟨{30, 80}⟩:2 là chuỗi phổ biến.
- ⟨{30} {40}⟩, ⟨{30} {70}⟩ và ⟨{30} {90}⟩ là mẫu tuần tự.
  - 40, 70 và 90 phổ biến nhưng không nằm cùng tập hạng mục với 30.

THUẬT TOÁN PREFIX-SPAN

- Tiếp tục tìm kiếm đệ qui trên các tập con cấp thấp hơn bằng cách xây dựng cơ sở dữ liệu chiếu tương ứng.
  - Ví dụ: tìm mẫu tuần tự có tiền tố ⟨{30} {40}⟩, CSDL chiếu-⟨{30} {40}⟩ gồm ⟨{_, 70}⟩ và ⟨{_, 70, 80} {90}⟩. ⟨{30} {40, 70}⟩ là mẫu tuần tự.
- Kết thúc quá trình khai thác trên CSDL chiếu-⟨{30}⟩, thu được: ⟨{30}⟩, ⟨{30} {40}⟩, ⟨{30} {40, 70}⟩, ⟨{30} {70}⟩, ⟨{30} {90}⟩, ⟨{30, 70}⟩, ⟨{30, 80}⟩ và ⟨{30, 70, 80}⟩.

SỬ DỤNG NHIỀU MINSUP

- Gọi MIS(*i*) là độ hỗ trợ hạng mục tối thiểu của hạng mục *i*, do người dùng chỉ định.
- Gọi φ là ngưỡng trong ràng buộc hiệu số hỗ trợ, $|\text{sup}(i) - \text{sup}(j)| \leq \varphi$.
  - *i* và *j* là hạng mục trong cùng một mẫu tuần tự
  - sup(*x*) là độ hỗ trợ thật sự của hạng mục *x* trong cơ sở dữ liệu chuỗi *S*.

THUẬT TOÁN MS-PS

1. Tìm mọi hạng mục i trong S thỏa MIS(i). i gọi là hạng mục phổ biến.
2. Sắp xếp các hạng mục phổ biến tăng dần theo giá trị MIS, kí hiệu i_1,..., i_u.
3. Với mỗi hạng mục i_k theo thứ tự trên
   i. Nhận diện mọi chuỗi trong S chứa i_k và xóa mọi hạng mục j không thỏa $|\text{sup}(i) - \text{sup}(j)| \leq \varphi$ trong các chuỗi này. Kí hiệu tập chuỗi kết quả là S_k.
   ii. Gọi hàm r-PrefixSpan tìm mẫu tuần tự chứa i_k trên S_k. count(MIS(i_k)) là độ hỗ trợ tối thiểu duy nhất. Sau khi tìm mẫu tuần tự xong, loại mọi xuất hiện i_k ra khỏi S.
      - count(MIS(i_k)) là đếm hỗ trợ tối thiểu tính theo số lượng chuỗi

HÀM r-PREFIXSPAN

- Hàm r-PrefixSpan gần giống như PrefixSpan ngoại trừ các điểm sau
  - Trong mỗi lượt gọi đệ quy, hoặc tiền tố hoặc mọi chuỗi trong cơ sở dữ liệu chiếu phải chứa i_k.
  - Ràng buộc hiệu số hỗ trợ cần được kiểm tra trong mỗi lượt chiếu vì sup(i_k) có thể không nhỏ nhất trong mẫu.

VÍ DỤ THUẬT TOÁN MS-PS

- Xét cơ sở dữ liệu chuỗi sau

Sequence ID	Data Sequence
1	⟨{20, 50}⟩
2	⟨{40}{30}{40, 60}⟩
3	⟨{40, 90, 120}⟩
4	⟨{30}{20, 40}{40, 100}⟩
5	⟨{20, 40}{10}⟩
6	⟨{40}{30}{110}⟩
7	⟨{20}{80}{70}⟩

- MIS(20) = 30% (3 chuỗi), MIS(30) = 20% (2 chuỗi), MIS(40) = 30% (3 chuỗi), MIS cho các hạng mục còn lại = 15% (2 chuỗi).
- Bỏ qua ràng buộc hiệu số hỗ trợ.

VÍ DỤ THUẬT TOÁN MS-PS

Sequence ID	Data Sequence
1	⟨{20, 50}⟩
2	⟨{40}{30}{40, 60}⟩
3	⟨{40, 90, 120}⟩
4	⟨{30}{20, 40}{40, 100}⟩
5	⟨{20, 40}{10}⟩
6	⟨{40}{30}{110}⟩
7	⟨{20}{80}{70}⟩

MIS(20) = 30% (3 chuỗi),

MIS(30) = 20% (2 chuỗi)

MIS(40) = 30% (3 chuỗi),

MIS hạng mục khác = 15% (2 chuỗi).

- Bước 1: 20, 30 và 40 là hạng mục phổ biến.
- Bước 2: danh sách sắp xếp (30, 20, 40).

VÍ DỤ THUẬT TOÁN MS-PS

- Bước 3 - Vòng lặp thứ nhất: $i_1 = 30$:
  - S_1 = {⟨{40} {30} {40, 60}⟩, ⟨{30} {20, 40} {40, 100}⟩, ⟨{40} {30} {110}⟩}
  - Gọi hàm r-PrefixSpan(30, S_1, 2). Hạng mục phổ biến trong S_1 là 30 và 40 $\Rightarrow$ Chuỗi phổ biến độ dài 1 là ⟨{30}⟩.
  - Tìm mẫu phổ biến có tiền tố ⟨{30}⟩
    - Cơ sở dữ liệu chiếu-⟨{30}⟩: {⟨{40}⟩, ⟨{40} {40}⟩}
    - Mẫu phổ biến độ dài 2: ⟨{30} {40}⟩
  - Tìm mẫu phổ biến có tiền tố ⟨{40}⟩.
    - Cơ sở dữ liệu chiếu-⟨{40}⟩: {⟨{30} {40}⟩, ⟨{30}⟩}
    - Mẫu phổ biến độ dài 2: ⟨{40} {30}⟩

VÍ DỤ THUẬT TOÁN MS-PS

- Bước 3 - Vòng lặp thứ hai: $i_2 = 20$:
  - S_2 = {⟨{20, 50}⟩, ⟨{20, 40} {40, 100}⟩, ⟨{20, 40} {10}⟩, ⟨{20} {80} {70}⟩}
  - Gọi hàm r-PrefixSpan(20, S_2, 3). Hạng mục phổ biến trong S_2 là 20 $\Rightarrow$ Chuỗi phổ biến độ dài 1 là ⟨{20}⟩.
- Bước 3 - Vòng lặp thứ ba: $i_3 = 40$:
  - S_3 = {⟨{40} {30} {40, 60}⟩, ⟨{40, 90, 120}⟩, ⟨{30} {20, 40} {40, 100}⟩, ⟨{20, 40} {10}⟩, ⟨{40, 30} {110}⟩}
  - Gọi hàm r-PrefixSpan(40, S_3, 3). Hạng mục phổ biến trong S_3 là 40 $\Rightarrow$ Chuỗi phổ biến độ dài 1 là ⟨{40}⟩.
- Tập mẫu tuần tự: {⟨{30}⟩, ⟨{20}⟩, ⟨{40}⟩, ⟨{40} {30}⟩, ⟨{30} {40}⟩}.

PHÁT SINH LUẬT TỪ MẪU TUẦN TỰ

LUẬT TUẦN TỰ

- Luật tuần tự (sequential rule – SR) có dạng $X \rightarrow Y$ trong đó Y là chuỗi và X là chuỗi con có độ dài nhỏ hơn Y.
- Độ hỗ trợ của luật $X \rightarrow Y$ trong cơ sở dữ liệu chuỗi S là tỉ lệ chuỗi trong S chứa Y.
- Độ tin cậy của luật $X \rightarrow Y$ trong S là tỉ lệ chuỗi trong S chứa X thì cũng chứa Y.
- Phát sinh luật từ chuỗi phổ biến.

VÍ DỤ LUẬT TUẦN TỰ

- minsup = 30% và minconf = 60%

	Data Sequence
1	⟨{1}{3}{5}{7, 8, 9}⟩
2	⟨{1}{3}{6}{7, 8}⟩
3	⟨{1, 6}{7}⟩
4	⟨{1}{3}{5, 6}⟩
5	⟨{1}{3}{4}⟩

- Một luật tuần tự có thể có là

⟨{1} {7}⟩ → ⟨{1} {3} {7, 8}⟩ [sup = 2/5, conf = 2/3]

LUẬT TUẦN TỰ NHÃN

- Luật tuần tự nhãn (label sequential rule – LSR) có dạng $X \rightarrow Y$ trong đó Y là chuỗi và X là chuỗi tạo từ Y bằng cách thay một vài hạng mục bằng wildcard.
  - Một wildcard "*" có thể khớp với hạng mục bất kì.
- Hạng mục bị thế thường rất quan trọng và được gọi là nhãn (label). Nhãn là tập con nhỏ trong số các hạng mục của dữ liệu.

VÍ DỤ LUẬT TUẦN TỰ NHÃN

- minsup = 30% và minconf = 60%

	Data Sequence
1	⟨{1}{3}{5}{7, 8, 9}⟩
2	⟨{1}{3}{6}{7, 8}⟩
3	⟨{1, 6}{7}⟩
4	⟨{1}{3}{5, 6}⟩
5	⟨{1}{3}{4}⟩

- Một luật tuần tự có thể có là

⟨{1}{*} {7,*}⟩ → ⟨{1} {3} {7, 8}⟩ [sup = 2/5, conf = 2/2]

LUẬT TUẦN TỰ NHÃN

- LSR dùng để dự đoán các nhãn trong một chuỗi đầu vào.
  - Ví dụ: hạng mục 3 và 8 trong ví dụ trước.
- Độ tin cậy luật ước lượng xác suất để "*" là 3 và 8, cho trước chuỗi ⟨{1}{*} {7,*}⟩.
- Ứng dụng: dự đoán một từ trong câu so sánh là một thực thể (ví dụ: tên sản phẩm).

LUẬT TUẦN TỰ NHÃN

- Độ tin cậy luật có thể được định nghĩa theo nhiều cách khác nhau tùy vào ứng dụng.
  - Để tính độ tin cậy, chuỗi phổ biến chưa đủ, còn phải duyệt dữ liệu.
  - Mẫu có thể xuất hiện trong một chuỗi dữ liệu nhiều lần.
- Ta có thể giới hạn wildcard trong phạm vi một số loại hạng mục để việc dự đoán nhãn có ý nghĩa và không nhập nhằng.

LUẬT TUẦN TỰ LỚP

- Gọi S là tập các chuỗi dữ liệu, mỗi chuỗi mang lớp y. I là tập các hạng mục trong S. Y là tập các nhãn lớp, $I \cap Y = \varnothing$.
- Luật tuần tự lớp (class sequential rule – CSR) có dạng $X \rightarrow y$ trong đó X là chuỗi và $y_i \in Y$.
- Một bộ (s_i, y_i) gọi là
  - phủ luật $X \rightarrow y$ nếu X là chuỗi con của s_i,
  - thỏa mãn luật $X \rightarrow y$ nếu X là chuỗi con của s_i và $y_i = y$.

VÍ DỤ LUẬT TUẦN TỰ LỚP

- minsup = 30% và minconf = 60%

	Data Sequence	Class
1	⟨{1}{3}{5}{7, 8, 9}⟩	c_1
2	⟨{1}{3}{6}{7, 8}⟩	c_1
3	⟨{1, 6}{9}⟩	c_2
4	⟨{3}{5, 6}⟩	c_2
5	⟨{1}{3}{4}{7, 8}⟩	c_2

- Một luật tuần tự có thể có là

$\langle\{1\}\{3\}\ \{7,8\}\rangle \rightarrow c_1$ [sup = 2/5, conf = 2/3]

TÀI LIỆU THAM KHẢO

- Tài liệu bài giảng môn học
- Chapter 2. B. Liu, *Web Data Mining-Exploring Hyperlinks, Contents, and Usage Data*, Springer Series on Data-Centric Systems and Applications, 2007.
- Chapter 5. J.Han, M.Kamber, Data Mining: Concepts & Technique, 2nd edition, Morgan Kauffman, 2006.

TỔNG KẾT

- Các thuật ngữ cơ bản về cơ sở dữ liệu chuỗi và mẫu tuần tự.
- Thuật toán khai thác mẫu tuần tự: PrefixSpan và GSP.
  - Cài đặt chương trình chạy thuật toán GSP và PrefixSpan.
  - Thực hiện chạy tay thuật toán GSP
- Các loại luật kết hợp xây dựng trên dữ liệu chuỗi tuần tự.

KẾT THÚC CHỦ ĐỀ