LUẬT KẾT HỢP

* Giới thiệu

Cho tập hạng mục (itemset) I = {i1, i2,…, in}; ii ∈ I được gọi là một hạng mục (item).

Cho CSDL giao dịch (transaction) T = {t1, t2,…, tm}; trong đó mỗi ti là một giao dịch và là một tập con của I.

Số lượng các giao dịch T ký hiệu là |T| hay card(T).

Tập k hạng mục (k - itemset) = {i1, i2,…, ik}; gồm k item trong I.

* Độ phổ biến:
* Supp(X) = |X|/|D|
* Supp(X⇒Y)=|{T⊆D:X∪Y⊆T}|/|D|
* Độ tin cậy
* 𝐶𝑜𝑛𝑓 𝑋⇒Y =
* Ví dụ: |D|=10;|X∪Y|=5;

|X|=7;|Y|=6 Supp(X⇒Y) = 5/10 = 0,5

Conf(X⇒Y) = 5/7 ≈ 0,71

Luật kết hợp X→Y được coi là một mẫu có giá trị nếu xảy ra đồng thời supp(X→Y) ≥ minsup và conf(X→Y) ≥ minconf( minsup và minconf là hai giá trị ngưỡng cho trước).

* Tập phổ biến (frequent itemsets - FIs): một tập hạng mục X có độ hỗ trợ vượt qua ngưỡng minsup được gọi là tập phổ biến
* Tập phổ biến tối đại (Max Pattern/Max Frequent Itemsets) X nếu thỏa:

1. Supp(X) ≥ minsupp và

2. ∄ 𝑋′ ⊃ 𝑋 mà X’ cũng phổ biến.

* Tập phổ biến đóng (Closed Pattern/Frequent Closed Itemsets) X nếu thỏa:

1. Supp(X) ≥ minsupp và

1. ∄ 𝑋′ ⊃ 𝑋 mà supp(X’) = supp(X).

* Qui trình khai thác luật kết hợp:
* Bước 1: Tìm các tập phổ biến thỏa ngưỡng minsupp.
* Bước 2: Xây dựng luật từ các tập phổ biến:
* Đối với mỗi tập phổ biến S, tạo mọi tập con khác rỗng của S.
* Đối với mọi tập con khác rỗng A của S. Luật A  (S –A) là luật kết hợp cần tìm nếu conf(A  (S –A)) ≥ minconf.

Bài toán khai thác luật kết hợp được chia làm hai giai đoạn:

1. Khai thác tập phổ biến (FIs – Frequent Itemsets):

2. Khai thác luật từ các tập phổ biến (ARs – Association Rules)

* Bài toán khai thác tập phổ biến (frequent itemset) là bài toán rất quan trọng trong lĩnh vực data mining.

Bài toán khai thác tập phổ biến là bài toán tìm tất cả tập các hạng mục (itemset) S có độ phổ biến (support) thỏa mãn độ phổ biến tối thiểu minsupp: supp(S) ≥ minsupp

* Một số thuật toán khai thác tập phổ biến:
* Thuật toánApriori (Agrawal et al).
* Thuật toán IT-tree (Zaki – et al).
* Thuật toán FP-Tree (Han et al).
* Các thuật toán khác như LCM, DCI, PrePost, …

1. Thuật toán Apriori

* Thuật toán Apriori là một thuật toán điển hình áp dụng trong khai phá luật kết hợp.
* Thuật toán dựa trên nguyên lý Apriori “tập con bất kỳ của một tập phổ biến cũng là một tập phổ biến và mọi tập cha của tập không phổ biến là không phổ biến”.
* Mục đích của thuật toán Apriori là tìm ra được tất cả các tập phổ biến có thể có trong cơ sở dữ liệu giao dịch D.
* Thuật toán hoạt động theo nguyên tắc quy hoạch động, nghĩa là từ các tập Fi = { ci | ci là tập phổ biến, |ci| = i} gồm mọi tập phổ biến có độ dài i (1 ≤ i ≤ k), đi tìm tập Fk+1 gồm mọi tập phổ biến có độ dài k+1. Các hạng mục i1, i2,…, in trong thuật toán được sắp xếp theo một thứ tự cố định.
* Ý nghĩa nguyên lýApriori:
* Mọi tập con của tập phổ biến đều phổ biến, nghĩa là ∀ 𝑋 ⊆ 𝑌 , nếu supp(Y) ≥ minsupp thì supp(X) ≥ minsupp.
* Mọi tập cha của tập không phổ biến đều không phổ biến. nghĩa là ∀ 𝑋 ⊆ 𝑌, nếu supp(X) ≤ minsupp thì supp(Y) ≤ minsupp.
* Thuật toánApriori Input: CSDL giao dịch T và ngưỡng phổ biến minsupp. Output: FIs chưa tất cả các tập phổ biến của T.
* Ý tưởng thuật toánApriori

1. Tìm tất cả các tập phổ biến 1 hạng mục (L1)
2. Tạo các tập ứng viên kích thước k hạng mục Ck từ các tập phổ biến Lk-1 hạng mục (nguyên lý Apriori). Ví dụ, tạo ứng viên C2 từ tập phổ biến L1.
3. Kiểm tra độ phổ biến của các ứng viên trên CSDL và loại các ứng viên không phổ biến ta được Li(i=1,2,..., k).
4. Dừng khi không tạo được tập phổ biến hay tập ứng viên

* Thuật toán Apriori

◦ Mã giả: Gọi Ck là tập các ứng viên có kích thước k Lk các tập phổ biến có kích thước k L1= { i ∈ I: supp(i) ≥ minsupp} for (k = 1; Lk !=∅; k++) do Ck+1 =apriori\_gen(Lk)

//tập ứng viên được tạo từ Lk-1 for each t ∈ T do for each c ∈ Ck do if c ⊆ t then c.count++ Lk = {c ∈ Ck| c.count ≥ minSup} FIs = ∪𝑘Lk;

Tạo tập ứng viên Ck+1 hạng mục từ tập Lk hạng mục: Gồm 2 bước: ◦ kết và loại bỏ ◦ Giả sử rằng Lk được xếp theo thứ tự từ điển Hàm apriori\_gen: ◦ Input: Lk tập phổ biến kích thước k ◦ Output: Ck+1 tập ứng viên kích thước k+1

◦ Mã giả:

Procedure apriori\_gen(Lk) { For mỗi l1 ∈ Lk For mỗi l2 ∈ Lk If (l1[1]=l2[1] ⋀ … ⋀ l1[k-1]=l2[k-1] ⋀ l1[k]≠l2[k] ) then C = l1 ⋈ l2 If (có tập con của C ∉ không thuộc Lk ) then xóa C else them C vào Ck+1 Return Ck+1 }

◦ Ví dụ: Cho CSDL sau và minsupp = 50%.

Mã giao dịch (TID)

Nội dung giao dịch 1 A, C, D 2 B, C, E 3 A, B, C, E 4 B, E

Itemset {A} {B} {C} {D} {E}

Itemset Supp {A} 2 {B} 2 {C} 3 {E} 3

C1 L1

Itemset Supp {A} 2 {B} 2 {C} 3 {E} 3

C2 L2 Itemset {A, B} {A, C} {A, E} {B, C} {B, E} {C, E}

Itemset Supp {A, C} 2 {B, C} 2 {B, E} 3 {C, E} 2

◦ Chú ý: {A, C, B} không có trong C3 vì {A, B} không có trong L2.

C3 L3 Itemset Supp {B, C, E} 2

Itemset Supp {A, C} 2 {B, C} 2 {B, E} 3 {C, E} 2

Itemset {B, C, E}

◦ Các tập phổ biến tìm được là: FIs = {{A}, {B}, {C}, {E}, {A, C}, {B, C}, {B, E}, {C, E}, {B, C, E}}.

35

◦ Thuật toán Apriori là thuật toán dễ hiểu, dễ cài đặt. Tuy nhiên có các nhược điểm sau:

◦ Phải duyệt CSDL nhiều lần.

◦ Số lượng tập ứng viên rất lớn (=2|T|- 1).

◦ Thực hiện việc tính độ phổ biến nhiều, đơn điệu.

◦ Cải tiến Apriori: Ý tưởng chung:

◦ Giảm số lần duyệt CSDL.

◦ Giảm số lượng (không sinh) tập ứng viên.

◦ Tìm qui trình tính độ phổ biến hiệu quả hơn.