Abstract

Đề xuất một cách tiếp cận mới, đưa N-list vào giải quyết mẫu tuần tự phổ biến.

Bài toán khai thác mẫu tuần tự phổ biến có vai trò quan trọng trong việc khai thác dữ liệu, được ứng dụng trong nhiều lĩnh vực được giới thiệu lần đầu bởi Agrawal and Srinkant vào năm 1995 [..]

Để tìm mẫu tuần tự phổ biến, những thuật toán hiệu quả như ApriopriAll [],GSP [], SPADE [].

Gần đây Lin (2008) đề xuất FUSP-tree [],

+ *với bài báo khoa học*:

[2]. Hai Bang T., Ngoc Thanh N., Phi Khu N. (2011), “*Fuzzy Ontology Building and Integration for Fuzzy Inference Systems in Weather Forecast Domain*”. LNAI 6592, the 3rd Intl’ Conference, ACIIDS 2011, Korea. Springer, ISSN 0302-9743, ISBN 978-3-642-20042-7.

PrefixSpan [6] thuật toán này mở rộng từ FP-growth, trong cách tiếp cận PrefixSpan cơ sở dữ liệu tuần tự được chiếu bằng cách sau; (1) quét qua cơ sở dữ liệu tìm tất cả các mẫu phổ biến trong chuỗi. Mỗi mẫu phổ biến là một mẫu tuần tự phổ biến một phần tử. (2) Sau đó chia không gian tìm kiếm, tập hợp tất các chuỗi tuần tự phổ biến có thể chia theo các mẫu phổ biến được tìm thấy ở bước 1. (3) Tìm những chuỗi con bằng cách xây dựng các cơ sở dữ liệu được chiếu tương ứng, quá trình này được đệ quy.

Vào năm 2008, Lin đã đưa ra FUSP-tree [7] và thuật toán để cập nhật cây một cách hiệu quả. Cây FUSP-tree gồm một nút gốc (root) và một tập các cây con như là con của nút gốc. Mỗi nút trong cây con gồm **item-name** thể hiện cho nút chứa mẫu, **count** số chuỗi có đường đi qua nút này, **node-links** liên kết tới nút kế tiếp của cùng mẫu của nhánh tiếp theo. FUSP-tree chứa Header-Table đây là bảng chứa những mẫu phổ biến, độ phổ biến của mẫu và liên kết tới nút đầu tiên của một mẫu.Việc tạo ra cấu trúc cây này từ cơ sở dữ liệu ban đầu cũng tương tự như việc tao ra cây FP-tree. Nhưng khác nhau là liên kết giữ hai nút trên cây phần làm hai loại là ‘s’ và ‘i’. Liên kết ‘s’ là liên kết giữa hai sự kiện khác trong chuỗi. Liên kết ‘i’ là liên kết giữ nhưng mẫu trong cùng một sự kiện. Để làm rõ việc xây dựng cây ta xem xét ví dụ sau.

Cho cơ sở dữ liệu tuần tự như sau:

|  |  |
| --- | --- |
| Sequence ID | Sequence ID |
| 10 | <(a)(abc)(ac)(d)(cf)> |
| 20 | <(ad)(c)(bc)(ae)> |
| 30 | <(ef)(ab)(df)(c)(b)> |
| 40 | <(e)g)(af)(c)(b)(c)> |

Bithi và các đồng sự (2012) đưa ra “Tree Based Sequential Pattern Mining” [8] phát triển từ FUSP-tree. Sau khi cây FUSP-tree được xây dựng từ cơ sở dữ liệu ban đầu, việc khai thác mẫu phổ biến diễn ra từ cây này. Cây FUSP-tree điều kiện được sinh ra tương ứng với mỗi mẫu tuần tự trong Header-table. Quá trình này đệ quy đến khi tất cả các chuỗi tuần tự phổ biến được sinh ra. Thuật toán này khai thác được tất cả các mẫu tuần tự phô biến không sinh ra mẫu ứng viên, và giảm số lần phải quét toàn bộ cơ sở dữ liệu. Và qua thực nghiệm cho thấy kết quả tốt hơn các thuật toán trước đó như GSP, PrefixSpan.

Trong những năm gần đây, Deng đề xuất N-list và thuật toán PrePost [9]. PrePost được phát triển trên cấu trúc N-list, N-list là một cấu trúc dữ liệu mới để thể hiện itemset. Cấu trúc dữ liệu này lưu những thông tin cần thiết về itemset. Cũng với việc kết hợp cách tìm kiếm sinh ra tập ứng viên và cách tìm kiếm không sinh ra tập ứng viên PrePost đã giải quyết rất hiệu quả bài toán khai thác mẫu phổ biến.

Cấu trúc cây PPC-tree gồm một nút gốc với một tập nhưng nút con, mỗi nút con gồm các thành phần là: item-name, count, children-list, pre-order, post-order. **item-name** thể hiện cho nút chứa mẫu, **count** số giao dịch có đường đi qua nút này, **children-list** ghi nhận tất cả các nút con của nút này, **pre-order** là số thứ tự có được từ việc duyệt cây theo chiều từ trên xuống, **post-order** là số thứ tự có được từ việc duyệt cây theo chiều từ dưới lên. Cây PPC-tree khác cây FP-tree ở những điểm sau: PPC-tree không có header table để duy trì sự kết nối giữ các nút. Mỗi nút của cây PPC-tree có thêm thuộc tính là pre-order, post-order. FP-tree đươc sử dụng trong toàn bộ quá trình của thuật toán FP-growth trong khi đó PPC-tree chỉ để sinh ra Pre-Post của mỗi nút.

N-list là một tập có thứ tự của những nút sinh ra từ cây PPC-tree. N-list có hai thuộc tính quan trong. Một là N-list của itemset có độ dài k+1 được tạo thành từ việc join N-list của nhưng tập con có độ dài k. Hai là độ phổ biến của một itemset là count của nhưng nút trên N-list. PrePost cho thấy hiệu quả hơn FP-growth nhờ N-list, nhưng cần đánh chỉ số mỗi nút của PPC-tree với các chỉ số pre-order và post-order dẫn đến tiêu hao bộ nhớ.

PrePost hiệu quả bởi ba đặc điểm sau: N-list có cùng cách nén dữ liệu như FP-tree nên làm dữ liệu nhỏ gọn hơn các cấu trúc theo chiều dọc được đề xuất trước đó. Việc đếm độ phổ biến được chuyển vào việc tìm giao giữa các N-list. PrePost có thể tìm những mẫu phổ biến không cần sinh ra tập ứng viên trong một số trường hợp bởi việc sử dụng một đường đơn của N-list.

Với I = { , , …, } là một tập m phần tử khác nhau, được gọi là những mẫu. Một sự kiện (trong thực tế một sự kiện có thế có một hoặc nhiều phần tử) là một tập hợp không có thứ tự và khác rỗng của những mẫu (Xem sự kiện chỉ có một phần tử). Một chuỗi là một danh sách có thứ tự của các sự kiện. Một chuỗi S được ký hiệu ( ,,), với mỗi phần tử là một sự kiện. Số sự kiện trong chuỗi được gọi là kích thước của chuỗi và tổng số mẫu của chuỗi được gọi là độ dài của chuỗi. Một chuỗi có độ dài k được ký hiệu là k-sequence.

Một chuỗi S = ( → → ··· → ) được gọi là chuỗi con của chuỗi R = ( → → ··· → ), được ký hiệu S ⊆ R, nếu tồn tại một số nguyên < < ··· < sao cho ⊆ với tất cả .

Cho một cơ sở dữ liệu tuần tự D, với mỗi chuỗi có một id duy nhất, cho một chuỗi S = ( → → ··· → ), với độ phổ biến của S trong D được tính bằng số lần chuỗi S xuất hiện trong D. Với một ngưỡng phổ biến cho trước (ký hiệu là minsup), một chuỗi được gọi là mẫu phổ biến nếu nó có độ phổ biến thỏa mãn ngưỡng minsup.

Cho một cơ sở dữ liệu tuần tự D, và ngưỡng minsup, bài toán là tìm tất cả những chuỗi tuần tự phổ biến trong D.

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Với I = { , , …, } là một tập m phần tử khác nhau, được gọi là những mẫu. Một sự kiện (trong thực tế một sự kiện có thế có một hoặc nhiều phần tử) là một tập hợp không có thứ tự và khác rỗng của những mẫu (Xem sự kiện chỉ có một phần tử). Một chuỗi là một danh sách có thứ tự của các sự kiện. Một chuỗi S được ký hiệu ( …), với mỗi phần tử là một sự kiện. Số sự kiện trong chuỗi được gọi là kích thước của chuỗi và tổng số mẫu của chuỗi được gọi là độ dài của chuỗi. Một chuỗi có độ dài k được ký hiệu là k-sequence.

Một chuỗi α = ( …) được gọi là chuỗi con của chuỗi β = ( ··· ), được ký hiệu α ⊆ β, nếu tồn tại một số nguyên 1 ≤ < < ··· < ≤ m sao cho ⊆ .

Cho một cơ sở dữ liệu tuần tự D, với mỗi chuỗi có một id duy nhất, cho một chuỗi S = ( → → ··· → ), với độ phổ biến của S trong D được tính bằng số lần chuỗi S xuất hiện trong D. Với một ngưỡng phổ biến cho trước (ký hiệu là minsup), một chuỗi được gọi là mẫu phổ biến nếu nó có độ phổ biến thỏa mãn ngưỡng minsup.

Cho một cơ sở dữ liệu tuần tự D, và ngưỡng minsup, bài toán là tìm tất cả những chuỗi tuần tự phổ biến trong D.