ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

ĐỀ CƯƠNG ĐỀ TÀI LUẬN VĂN THẠC SĨ (PTII): 15TC

1. **Tên đề tài hoặc hướng nghiên cứu** (*gồm cả tiếng Việt và tiếng Anh*):
   Tiếng Việt: Ứng dụng N-list trong khai thác mẫu tuần tự phổ biến
   Tiếng Anh: Sequential pattern mining using N-list.

2. **Ngành và mã ngành**: Ngành khoa học máy tính, mã ngành 06.48.01.01

3. **Họ tên học viên**: BÙI VĂN BẰNG, khóa 10 - đợt 1

   Địa chỉ email, điện thoại liên lạc: vanbang0208@gmail.com - 0984219010

   Người hướng dẫn: PGS.TS VÕ ĐÌNH BẢY

   Địa chỉ email, điện thoại liên lạc: bayvodinh@gmail.com - 0937306858

4. **Tổng quan tình hình nghiên cứu**

Khai thác mẫu tuần tự phổ biến có vai trò quan trọng trong khai thác dữ liệu, bởi vì nó được ứng dụng trong nhiều lĩnh vực như phân tích thông tin giao dịch khách hàng, cách truy cập web, phân tích chuỗi DNA, … Xem xét một cơ sở dữ liệu giao dịch của một cửa hàng bán sách, với mỗi đối tượng đại diện cho một khách hàng, những thuộc tính đại diện cho tác tác giả hay tên sách. Cơ sở dữ liệu ghi nhận những quyển sách được mỗi khách hàng mua trong một khoảng thời gian nhất định. Câu hỏi đặt ra là chuỗi những quyển sách nào được khách hàng mua cùng nhau phổ biến trong một khoảng thời gian trên.

Một trang web nổi tiếng được nhiều người truy cập. Mỗi đối tượng là một người dùng, thuộc tính là một trang của web. Bài toán là tìm ra chuỗi các trang của web được truy cập phổ biến trong một khoảng thời gian nhất định.

Bài toán khai thác mẫu tuần tự phổ biến được phát biểu như sau:

Với I = { $i_1, i_2, \ldots, i_m$} là một tập m phần tử khác nhau, được gọi là những mẫu. Một sự kiện (trong thực tế một sự kiện có thể 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 ($s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_q$), với mỗi phần tử s_j 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 = ($s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_n$) được gọi là chuỗi con của chuỗi R = ($r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_m$), được ký hiệu S $\subseteq$ R, nếu tồn tại một số nguyên $i_1 < i_2 < \cdots < i_n$ sao cho $s_j \subseteq r_{i_j}$ với tất cả s_j.

Cho một cơ sở dữ liệu tuần tự D, với mỗi chuỗi có một id duy nhất, cho một chuỗi S = ($s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_n$), 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.

Bài toán khai thác mẫu tuần tự phổ biến được giới thiệu lần đầu bởi Agrawal và Srinkant vào năm 1995 [2]. Và từ đó đã có nhiều cách tiếp cận và thuật toán được đưa ra để giải quyết.

Các thuật toán để khai thác mẫu tuần tự phổ biến được để xuất trước đây như AproriAll [2], GSP [3], SPADE [5] cách tiếp cận này sinh ra tập ứng viên và kiểm tra độ phổ biến, yếu điểm của cách tiếp cận này là tốn bộ nhớ để lưu tập ứng viên và duyệt cơ sở dữ liệu nhiều lần. Các thuật toán PrefixSpan [6], FUSP [7] không sinh ra tập ứng viên, cải thiện được các khuyết điểm của các thuật toán trước đó nhưng cấu trúc cây phức tạp, tốn nhiều bộ nhớ tạm trong quá trình chiếu để tìm các chuỗi phổ biến tuần tự.

Cùng với việc đưa ra bài toán, Agrawal và Srinkant cũng đề xuất ba thuật toán để giả quyết vấn đề trên. Trong đó thuật toán AprioriAll [2] thực thi nhanh hơn hai thuật toán còn lại. AprioriAll gồm ba phần, đầu tiên là tìm tất cả các mẫu tuần tự phổ biến, sau đó loại các mẫu tuần tự không phổ biến, rồi khai thác mẫu tuần tự phổ biến. Vấn đề với cách tiếp cận này là tốn quá nhiều chi phí cho việc chuyển dữ liệu trong quá trình khai thác tập phổ biến và yêu cầu ổ đĩa gần như gấp đôi với kích thước cơ sở dữ liệu, điều này không thực tế với cơ sở dữ liệu lớn.

Thuật toán GSP của Agrawal và Srinkant 1996 [3], sau khi các mẫu đơn được đếm, từ những mẫu phổ biến đơn sinh ra mẫu ứng viên hai phần tử, sau đó tính độ phổ biến của những ứng viên để tìm ra mẫu phổ biến hai phần tử, từ đó sinh ra mẫu ứng viên ba phần tử. Qúa trình này được lặp lại cho đến khi không tìm ra mẫu phổ biến nào nữa. Thuật toán này cần yêu cầu quá nhiều lần quét toàn bộ cơ sở dữ liệu, mà việc này tốn rất nhiều thời gian khi cơ sở dữ liệu lớn dần.

Thuật toán SPADE [3] gồm việc xác định chuỗi phổ biến một phần tử, chuỗi phổ biến hai phần tử, sau đó chuyển vào dạng dữ liệu theo chiều dọc và xác định những chuỗi phổ biến khác bằng BFS hay DFS trong mỗi lớp. Thuật toán SPADE chỉ cần ba lần quét cơ sở dữ liệu để tìm được tất cả mẫu phổ biến. Qua thử nghiệm và thực tế cho thấy ưu việt hơn nhiều so với thuật toán GSP. Nhưng SPADE yêu cầu phải chuyển đổi cơ sở dữ liệu theo chiều ngang thành cơ sở dữ liệu theo chiều dọc điều này làm không gian lưu trữ lớn hơn không gian lưu trữ ban đầu.

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 tạo ra cây FP-tree. Nhưng khác nhau là liên kết giữ hai nút trên cây phân 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 những mẫu trong cùng một sự kiện.

Bithi và các đồng sự (2012) đưa ra “Tree Based 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 phổ 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.

Để giải quyết bài toán khai mẫu phổ biến từ cơ sở dữ liệu giao dịch [1] có nhiều phương pháp hiệu quả được đề xuất như: FP-growth [4], N-list [9].

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 những 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 trọng. Một là N-list của itemset có độ dài k+1 được tạo thành từ việc join N-list của những tập con có độ dài k. Hai là độ phổ biến của một itemset là tổng độ phổ biến của những 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.

Qua việc tìm hiểu các thuật toán cho việc giải quyết bài toán khai thác mẫu tuần tự phổ biến và bài toán khai thác mẫu phổ biến. Nhận thấy N-list là cấu trúc dữ liệu giúp nâng cao tốc độ giải quyết bài toán khai thác mẫu phổ biến. Cần tận dụng những ưu

điểm của cấu trúc này vào bài toán khai thác mẫu tuần tự. Nhưng việc hiệu chỉnh là không dễ dàng vì tính tuần tự của cơ sở dữ liệu tuần tự. Hiệu chỉnh N-list và sử dụng cây FUSP để khai thác mẫu tuần tự phổ biến có thể làm giảm thời gian đáng kể với việc tìm tất cả các mẫu tuần tự phổ biến từ cơ dữ liệu tuần tự.

5. Tính khoa học và tính mới của đề tài

Bài toán khai thác mẫu tuần tự phổ biến là một bài toán quan trọng của khai thác dữ liệu, có rất nhiều ứng dụng trong thực tế như phân tích thông tin giao dịch của khách hàng, cách truy cập web, phân tích chuỗi DNA. Những các thuật toán để giải quyết bài toán này thường tốn rất nhiều thời gian. Cấu trúc N-list được đề xuất trong vài năm trở lại đây được chứng minh rất hiệu quả cho việc khai thác mẫu phổ biến. Ứng dụng N-list vào bài toán khai thác mẫu tuần tự phổ biến để tận dụng những ưu điểm của cấu trúc này nhằm làm giảm thời gian thực thi của thuật toán. Nhưng việc hiệu chính cấu trúc này cho phù hợp với bài toán tìm mẫu tuần tự phổ biến là không dễ dàng vì tính thứ tự của mẫu dữ liệu, làm việc chia sẻ cấu trúc dữ liệu chung là một thách thức.

6. Mục tiêu, đối tượng và phạm vi nghiên cứu

6.1 Mục tiêu nghiên cứu

Mục tiêu tổng quan: Tìm hiểu về bài toán khai thác mẫu tuần tự phổ biến và các phương pháp giải quyết được đề xuất. Hiệu chỉnh lại cấu trúc N-list và cây FUSP để giải quyết bài toán khai thác mẫu tuần tự phổ biến.

Mục tiêu chi tiết:

+ Tìm hiểu về bài toán khai thác mẫu tuần tự phổ biến và các phương pháp giải quyết được đề xuất.
+ Hiệu chỉnh lại cấu trúc N-list và cây FUSP để giải quyết bài toán khai thác mẫu tuần tự phổ biến.
+ Kiểm tra kết quả, so sách với các phương pháp khác SPADE, PrefixSpan, FUSP.
+ Nhận xét ưu và khuyết điểm, đưa ra hướng phát triển.

6.2 Đối tượng và pham vi nghiên cứu

Đối tượng: Cơ sở dữ liệu dạng chuỗi.

Phạm vi: Các itemset là itemset đơn.

7. Nội dung, phương pháp dự định nghiên cứu

Nội dung 1: Tìm hiểu và cài đặt các thuật toán khai thác mẫu tuần tự phổ biến, nhận xét ưu và khuyết điểm.

+ Kết quả dự kiến: Báo cáo tổng quan về các thuật toán khai thác mẫu tuần tự phổ biến.

+ Phương pháp thực hiện: Tìm hiểu, nghiên cứu thông qua các bài báo, công trình, tài liệu trên các trang có uy tính, giảng viên hướng dẫn cung cấp, … về các thuật toán khai thác mẫu tuần tự phổ biến.

Nội dung 2: Nghiên cứu và cài đặt thuật toán khai thác mẫu tuần tự phổ biến dựa trên cấu trúc N-list.

+ Kết quả dự kiến: các lý thuyết về khai thác mẫu tuần tự phổ biến dựa trên cấu trúc N-list.

+ Phương pháp thực hiện: cài đặt thuật toán khai thác mẫu tuần tự phổ biến dựa trên cấu trúc N-list. Chạy thực nghiệm và so sánh với các thuật toán trước đó và rút ra kết luận.

8. Kế hoạch bố trí thời gian nghiên cứu:

Kế hoạch viết bản thảo luận văn

Thời gian	Công việc
Tháng 1	Nhận xét ưu và khuyết điểm của các thuật toán khai thác mẫu tuần tự phổ biến đã được đề xuất. Cài đặt các thuật toán khai thác mẫu tuần tự phổ biến đã có.
Tháng 2	Nghiên cứu hiệu chỉnh cấu trúc N-list và cây FUSP vào khai thác mẫu tuần tự phổ biến.
Tháng 3,4	Hiện thực trương trình và chỉnh sửa cho phù hợp thực tế.
Tháng 5	Viết và đăng bài báo khoa học.

Tháng 6	Viết, chỉnh sửa luận văn, thực hiện báo cáo luận văn trước hội đồng.

9.Tài liệu tham khảo

[1] Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami. "Mining association rules between sets of items in large databases." *Acm sigmod record*. Vol. 22. No. 2. ACM, 1993.

[2] Agrawal, Rakesh, and Ramakrishnan Srikant. "Mining sequential patterns." *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*. IEEE, 1995.

[3] Srikant, Ramakrishnan, and Rakesh Agrawal. "Mining sequential patterns: Generalizations and performance improvements." *International Conference on Extending Database Technology*. Springer Berlin Heidelberg, 1996.

[4] Han, Jiawei, Jian Pei, and Yiwen Yin. "Mining frequent patterns without candidate generation." *ACM Sigmod Record*. Vol. 29. No. 2. ACM, 2000.

[5] Zaki, Mohammed J. "SPADE: An efficient algorithm for mining frequent sequences." *Machine learning* 42.1-2 (2001): 31-60.

[6] Han, Jiawei, et al. "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth." *proceedings of the 17th international conference on data engineering*. 2001.

[7] Lin, Chun-Wei, et al. "An incremental fusp-tree maintenance algorithm." *2008 Eighth International Conference on Intelligent Systems Design and Applications*. Vol. 1. IEEE, 2008.

[8] Bithi, Ashin Ara, and Abu Ahmed Ferdaus. "Tree Based Sequential Pattern Mining." *IOSR Journal of Computer Engineering* 15.5 (2013): 79-89.

[9] Deng, ZhiHong, ZhongHui Wang, and JiaJian Jiang. "A new algorithm for fast mining frequent itemsets using N-lists." *Science China Information Sciences*_55.9 (2012): 2008-2030.

TP. HCM, ngày tháng năm 201..

NGƯỜI HƯỚNG DẪN	**HỌC VIÊN KÝ TÊN**
(*Họ tên và chữ ký*)	(*Họ tên và chữ ký*)
……..	……..