ĐẠI HỌC QUỐC GIA TP. HCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÙI DANH HƯỜNG

PHÁT TRIỂN CÁC THUẬT TOÁN KHAI THÁC MẪU VÀ MẪU ĐÓNG TRÊN CƠ SỞ DỮ LIỆU ĐỊNH LƯỢNG

Chuyên ngành: Khoa học máy tính

Mã số: 62.48.01.01

TÓM TẮT LUẬN ÁN TIẾN SĨ

KHOA HỌC MÁY TÍNH

TP. Hồ Chí Minh - 2021

Công trình này được hoàn thành tại: **Trường Đại học Công nghệ Thông tin (UIT), Đại học Quốc gia TP.HCM**.

Người hướng dẫn khoa học 1: **PGS.TS. Võ Đình Bảy**

Người hướng dẫn khoa học 2: **PGS.TS. Nguyễn Hoàng Tú Anh**

Phản biện độc lập 1: ...

Phản biện độc lập 2: ...

Luận án đã được bảo vệ trước Hội đồng chấm luận án họp tại: ...

Vào lúc ... giờ ... ngày ... tháng ... năm

Có thể tìm luận án tại:

- Thư viện Quốc gia Việt Nam.
- Thư viện Trường Đại học Công nghệ Thông tin, ĐHQG-HCM.

CHƯƠNG 1. TỔNG QUAN

1.1. Dẫn nhập

Trong luận án này, nghiên cứu sinh nghiên cứu, đề xuất cấu trúc dữ liệu và phát triển một số thuật toán hiệu quả để giải quyết một số bài toán liên quan khai thác mẫu phổ biến trên cơ sở dữ liệu có trọng số. Cụ thể như sau:

1. Đề xuất cấu trúc dữ liệu WN-list và thuật toán NFWI tương ứng để khai thác mẫu phổ biến có trọng số (FWI – *frequent weighted itemset*) một cách hiệu quả. WN-list là một mở rộng của cấu trúc N-list (Deng et al. (2012)) để biểu diễn dữ liệu có trọng số. Sự vượt trội của cấu trúc WN-list đến từ những ưu điểm sau: (1) Dữ liệu được nén trên cây dạng FP-tree; (2) Quan hệ tổ tiên của các nốt trên cây được xác định bằng cách so sánh các giá trị *pre* và *pos* của các nốt; (3) Nội dung của cây được chuyển sang dạng tuyến tính theo dạng $\{(pre_i, pos_i, weight_i)\}$ tương ứng với WN-list của các tập danh mục một phần tử và quá trình khai thác chỉ thực hiện trên đó; (4) Độ hỗ trợ trọng số *ws* của một tập danh mục được tính dễ dàng bằng tổng của các giá trị $weight_i$ trong WN-list của nó; (5) Độ phức tạp của phép giao WN-list chỉ là $O(n)$ và kích thước của WN-list kết quả được giảm thiểu đáng kể nhờ vào việc kết hợp các phần tử chung (*pre*, *pos*) lại với nhau. Thuật toán NFWI đã tận dụng các ưu điểm của cấu trúc WN-list để nén dữ liệu trên cây, sau đó trích xuất và khai thác trên các WN-list của các tập danh mục một phần tử . Các ứng viên lớp k được tạo ra tương ứng từ lớp (k-1) bằng phương pháp chia để trị và phép giao WN-list với độ phức tạp tuyến tính. Một số định lý đã được đề xuất để tính độ phổ biến trọng số của một tập danh mục dựa trên WN-list, cũng như xác định nhanh giá trị của nó trong một số trường hợp mà không cần thực hiện phép giao WN-list.

Bảng 1.1 Các đóng góp khoa học của luận án

Công bố / Thách thức	CT 1 2018 (ESWA – Q1)		CT2 2020 (APIN – Q2)		CT3 2020 (KNOSYS – Q1)		CT4 2021 (IEEE Access – Q1)		Các đóng góp khoa học
Các phương pháp hiện tại trong khai thác FWI đều có điểm yếu	X	1	X	1	X	1			**1**. Cấu trúc cây WN-tree lưu trữ hiệu quả CSDL có trọng số
		2		2		2	X	2	**2**. Cấu trúc dữ liệu WN-list dùng để khai thác hiệu quả FWI
		3		3		3		3	**3**. Phép giao WN-list có độ phức tập $O(n)$
		4		4		4		4	**4**. Công thức tính độ phổ biến có trọng số ws (X) dựa trên WN-list của itemset X
		5							**5**. Định lý xác định nhanh ws của một số itemset đặc biệt
		6							**6**. Thuật toán NFWI khai thác hiệu quả FWI
Số mẫu phổ biến tìm được quá lớn			X	7					**7**. Quan hệ tổ tiên WN-list và định lý tỉa nhánh một số ứng viên
Vấn đề luật dư thừa			X	8					**8**. Thuật toán NFWCI khai thác hiệu quả FWCI
Khai thác mẫu theo định hướng người dùng					X	9			**9.** Mô hình hóa bài toán khai thác Top-rank-k FWI
						10			**10**. Ba thuật toán cơ sở khai thác Top-rank-k FWI

<table>
<tr><th>Công bố
Thách thức</th><th>CT 1 2018 (ESWA – Q1)</th><th>CT2 2020 (APIN – Q2)</th><th colspan="2">CT3 2020 (KNOSYS – Q1)</th><th colspan="2">CT4 2021 (IEEE Access – Q1)</th><th>Các đóng góp khoa học</th></tr>
<tr><td></td><td colspan="2"></td><td></td><td>11</td><td colspan="2"></td><td>11. Thuật toán TFWIN+ khai thác hiệu quả Top-rank-k FWI dựa trên chiến lược tăng ngưỡng và chiến lược tỉa nhánh</td></tr>
<tr><td rowspan="3">Trích xuất và xử lý dữ liệu theo thời gian thực</td><td colspan="4" rowspan="3"></td><td rowspan="3">X</td><td>12</td><td>12. Mô hình hóa bài toán khai thác FWI theo luồng dữ liệu có trọng số</td></tr>
<tr><td>13</td><td>13. Cấu trúc cây SWN-tree lưu trữ hiệu quả luồng dữ liệu có trọng số</td></tr>
<tr><td>14</td><td>14. Thuật toán FWPODS khai thác hiệu quả FWI theo luồng dữ liệu có trọng số</td></tr>
</table>

2. Đề xuất thuật toán mới NFWCI khai thác hiệu quả mẫu phổ biến đóng có trọng số (FWCI – *frequent weighted closed itemset*) dựa trên cấu trúc WN-list và chiến lược tỉa nhánh nhanh. Luận án đã giới thiệu khái niệm về quan hệ tổ tiên WN-list và đề xuất một định lý để loại bỏ nhanh các ứng viên không thỏa mãn dựa trên quan hệ tổ tiên WN-list. Thuật toán NFWCI được đề xuất cũng áp dụng tính chất của phép giao WN-list khi kết hợp hai ứng viên không thỏa mãn trên cùng một lớp tương đương để giảm kích thước WN-list của ứng viên kết hợp, từ đó tăng tốc tính toán trong các bước kế tiếp.
3. Mô hình hóa bài toán khai thác top-rank-k mẫu phổ biến có trọng số. Khai thác top-rank-k mẫu phổ biến có trọng số là xác định các mẫu phổ biến có độ phổ biến trọng số nằm trong *k* ngưỡng lớn nhất, nhằm thỏa mãn nhu cầu của người dùng mà không mất thời gian đi xem xét toàn bộ các

mẫu phổ biến có trọng số. Ba thuật toán cơ sở TFWIT, TFWID và TFWIN được đề xuất để giải quyết bài toán khai thác top-rank-k mẫu phổ biến có trọng số tương ứng dựa trên ba cấu trúc dữ liệu hiện hành là tidset, diffset và WN-list. Các chiến lược tăng ngưỡng và tỉa nhánh sớm đã được đề xuất để cải tiến thuật toán TFWIN, từ đó đề xuất thuật toán TFWIN$^+$ khai thác top-rank-k mẫu phổ biến có trọng số.

4. Mô hình hóa bài toán khai thác mẫu phổ biến có trọng số theo luồng dữ liệu sử dụng mô hình cửa sổ trượt. Luận án đã đề xuất cấu trúc cây SWN-tree để lưu trữ và duy trì hiệu quả các cửa sổ dữ liệu khi trượt theo luồng dữ liệu. Trên cơ sở đó, thuật toán FWPODS được đề xuất để khai thác hiệu quả mẫu phổ biến có trọng số theo luồng dữ liệu dựa trên mô hình cửa sổ trượt.

1.2. Động lực nghiên cứu

Hiện nay, cùng với sự phát triển nhanh chóng của kỷ nguyên số, các dạng dữ liệu thu được từ các ứng dụng IoT ngày càng đa dạng và luôn có một đặc trưng cơ bản đó là các đối tượng trong cơ sở dữ liệu thường có mức độ quan trọng khác nhau và cần được chú ý với mức độ khác nhau trong quá trình xử lý và khai thác dữ liệu. Một số ví dụ như là cảm biến trong các hệ thống IoT có tầm quan trọng khác nhau, các mặt hàng có giá cả hay lợi nhuận khác nhau, các trang web có trọng số khác nhau, các gen hay triệu chứng bệnh thì có ý nghĩa khác nhau trong phân tích gen và chữa bệnh hay các nốt đồ thị biểu diễn mạng xã hội sẽ có trọng số khác nhau dựa theo các liên kết của nốt đồ thị đó.

Hiện tại, các phương pháp hiện có để giải quyết bài toán khai thác mẫu phổ biến có trọng số đang tồn tại một số điểm yếu. Ngoài ra, việc khai thác mẫu phổ biến có trọng số giới hạn theo định hướng nhu cầu của người sử dụng như về số lượng, thời gian vẫn còn chưa được quan tâm nghiên cứu. Như vậy, nhu cầu về

khai thác mẫu phổ biến trên cơ sở dữ liệu trọng số đang đặt ra cấp thiết và cần phải giải quyết.

Bảng 1.2 Ưu nhược điểm của các thuật toán khai thác mẫu phổ biến có trọng số hiện có

Thuật toán khai thác mẫu phổ biến có trọng số	Nén dữ liệu	Quét cây nén dữ liệu	Quét cây tìm kiếm	Tỉa nhánh	Hiệu quả trên loại CSDL
WIT-FWIs-Tid (Vo et al. (2013))			X	X	All
WIT-FWIs-Diff (Vo et al. (2013))			X	X	All
IWS (Nguyen et al. (2016))			X	X	Sparse
FWI-WSD (Lee et al. (2017))	X	X			All
FWI-TCD (Lee et al. (2017))	X	X			All

1.3. Mục đích, đối tượng và phạm vi nghiên cứu

- **Mục đích nghiên cứu**: Nghiên cứu và phát triển cấu trúc dữ liệu và thuật toán giải quyết hiệu quả các bài toán khai thác mẫu phổ biến có trọng số.
- **Đối tượng nghiên cứu**: Các bài toán: khai thác mẫu phổ biến có trọng số, khai thác mẫu phổ biến đóng có trọng số, khai thác top-rank-k mẫu phổ biến có trọng số và khai thác mẫu phổ biến có trọng số theo luồng dữ liệu.
- **Phạm vi nghiên cứu**: Giới hạn trên các bộ dữ liệu có trọng số

1.4. Ý nghĩa khoa học và thực tiễn của đề tài

- **Ý nghĩa khoa học của đề tài**: Nội dung của luận án là nghiên cứu và đề xuất các phương pháp để giải quyết hiệu quả các bài toán: khai thác mẫu phổ biến có trọng số, khai thác mẫu phổ biến đóng có trọng số, khai thác top-rank-k mẫu phổ biến có trọng số và khai thác mẫu phổ biến có trọng số theo luồng

dữ liệu. Các đóng góp của luận án góp phần bổ sung nền tảng lý thuyết về khai thác mẫu phổ biến trên các cơ sở dữ liệu có trọng số nói riêng và bài toán khai thác mẫu nói chung.

- **Ý nghĩa thực tiễn của đề tài**: Luận án cung cấp cấu trúc dữ liệu và thuật toán cụ thể để giải quyết các bài toán liên quan khai thác mẫu phổ biến có trọng số, từ đó các nhà phát triển có thể áp dụng vào trong các ứng dụng.

1.5. Bố cục luận án

Nội dung của luận án được bố cục gồm 7 chương và tài liệu tham khảo:

- **Chương 1: Tổng quan:** Chương này bao gồm phần giới thiệu tóm tắt về công trình nghiên cứu, động lực nghiên cứu, mục đích, đối tượng và phạm vi nghiên cứu, ý nghĩa khoa học và thực tiễn của đề tài.
- **Chương 2: Cơ sở lý thuyết:** Nội dung chương 2 giới thiệu cụ thể về các bài toán được quan tâm giải quyết trong luận án, khảo sát các nghiên cứu liên quan, những vấn đề còn tồn tại. Từ đó đề ra mục tiêu của luận án.
- **Chương 3: Khai thác mẫu phổ biến có trọng số bằng cấu trúc WN-list:** Chương 3 trình bày về cấu trúc dữ liệu WN-list và thuật toán NFWI để giải quyết bài toán khai thác mẫu phổ biến có trọng số.
- **Chương 4: Khai thác mẫu phổ biến đóng có trọng số bằng cấu trúc WN-list và chiến lược tỉa nhánh sớm:** Trình bày về phương pháp đề xuất để giải quyết bài toán khai thác mẫu phổ biến đóng có trọng số.
- **Chương 5: Khai thác top-rank-k mẫu phổ biến có trọng số:** Giới thiệu mô hình và các thuật toán khai thác top-rank-k mẫu phổ biến có trọng số.
- **Chương 6: Khai thác mẫu phổ biến có trọng số theo dòng dữ liệu dựa trên mô hình cửa sổ trượt:** Giới thiệu và đưa ra mô hình và thuật toán khai thác mẫu phổ biến có trọng số theo dòng dữ liệu dựa trên cửa sổ trượt.

- **Chương 7: Kết luận và hướng phát triển:** Trình bày tóm tắt về những kết quả nghiên cứu đạt được, những ưu điểm và hạn chế của các thuật toán đề xuất và các hướng nghiên cứu tiếp theo.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu bài toán

2.1.1. Bài toán khai thác mẫu phổ biến có trọng số

Định nghĩa 2.1. Một cơ sở dữ liệu có trọng số (WD – *weighted database*) là một bộ $\langle T, I, W\rangle$, trong đó $I = \{ i_j \mid j \in [1, n]\}$ là tập các danh mục (item), $T = \{t_j \mid j \in [1, m] \wedge t_j \subset I\}$ là tập các dòng dữ liệu, và $W = \{ w_j \mid j \in [1, n] \wedge w_j$ là trọng số của item $i_j\}$.

Định nghĩa 2.2 (Trọng số dòng dữ liệu) (Tao et al. (2003)): Trọng số của dòng dữ liệu $t_k \in T$ ký hiệu là $tw(t_k)$ được tính toán như sau:

$$tw(t_k) = \frac{\sum_{i_j \in t_k} w_j}{|t_k|}$$

Định nghĩa 2.3 (Độ hỗ trợ trọng số) (Tao et al. (2003))**:** Cho tập danh mục (itemset) X, độ hỗ trợ trọng số của X ký hiệu là $ws(X)$ được tính toán như sau:

$$ws(X) = \frac{\sum_{t_k \in t(X)} tw(t_k)}{sumtw}$$

Trong đó $t(X)$ là tập các dòng dữ liệu chứa itemset X và *sumtw* là tổng giá trị *tw* của tất cả các dòng dữ liệu.

Định nghĩa 2.4 (Mẫu phổ biến có trọng số) (Tao et al. (2003)): Cho itemset $X \subseteq I$ và một ngưỡng hỗ trợ trọng số tối thiểu *minws*, X được gọi là mẫu phổ biến có trọng số (FWI – *frequent weighted itemset*) nếu $ws(X) \geq minws$.

<u>Bài toán 1 (Khai thác FWI)</u>: Cho *WD* và ngưỡng tối thiểu *minws*. Khai thác mẫu phổ biến có trọng số là đi tìm tập *FWI* thỏa mãn:

$$\boldsymbol{FWI = \{X \in WD \mid ws(X) \geq minws\}}$$

2.1.2. Bài toán khai thác mẫu phổ biến đóng có trọng số

Định nghĩa 2.9 (Mẫu phổ biến đóng có trọng số) (Vo (2017))**:** Cho một itemset $X \subseteq I$ là FWI, X mẫu phổ biến đóng có trọng số (FWCI - *frequent weighted closed itemset*) nếu và chỉ nếu không tồn tại itemset Y sao cho $X \subset Y$ và $ws(X) = ws(Y)$.

<u>Bài toán 2 (Khai thác FWCI)</u>: Cho *WD* và ngưỡng tối thiểu *minws*. Bài toán khai thác mẫu phổ biến đóng có trọng số là đi tìm tập *FWCI* thỏa mãn:

$$\boldsymbol{FWCI = \{X \in WD \mid ws(X) \geq minws \wedge (\nexists\, Y \mid Y \supset X \wedge ws(X) = ws(Y))\}}$$

2.1.3. Bài toán khai thác top-rank-k mẫu phổ biến có trọng số

Định nghĩa 2.10 (Rank của mẫu phổ biến có trọng số): Cho một cơ sở dữ liệu có trọng số WD, rank của một mẫu phổ biến có trọng số X ký hiệu là $r(X)$ được xác định như sau:

$$r(X) = |\{ws(Y) | Y \subseteq I \wedge ws(Y) \geq ws(X)\}|$$

Định nghĩa 2.11 (Top-rank-k mẫu phổ biến có trọng số): Một mẫu X ($\subseteq I$) được gọi là top-rank-k mẫu phổ biến có trọng số nếu và chỉ nếu $r(X) \leq k$.

<u>Bài toán 3 (Khai thác Top-rank-k FWI)</u>: Cho *WD* và ngưỡng *k*, bài toán khai thác Top-rank-k mẫu phổ biến có trọng số là bài toán đi tìm tập *TR* thỏa mãn:

$$\boldsymbol{TR = \{X \in WD \mid r(X) \leq k\}}$$

2.2.4. Bài toán khai thác mẫu phổ biến có trọng số theo dòng dữ liệu

<u>Bài toán 4 (Khai thác FWI theo luồng dữ liệu)</u>: Cho *WD* là luồng dữ liệu cập nhật theo thời gian với bước nhảy là *p* giao dịch và ngưỡng tối thiểu *minws*. *I* là tập các item trong *WD*. $\boldsymbol{WD_i = \{t_k \mid t_k \subseteq I \wedge k \in [p * (i-1) + i, p * (i-1) + s]\}}$ **là cửa sổ trượt có kích thước cố định *s* của lần cập nhật thứ *i*. Bài toán khai thác FWI theo luồng dữ liệu là là đi tìm các tập** $\boldsymbol{FWI_i}$ **thỏa mãn:**

$$\boldsymbol{FWI_i = \{X \in WD_i \mid ws(X) \geq minws\}}$$

2.2. Các nghiên cứu liên quan

2.2.1. Khai thác mẫu phổ biến có trọng số

Năm 2003 Tao và đồng sự giới thiệu một mô hình nền tảng để khai thác mẫu phổ biến có trọng số dựa trên hai khái niệm trọng số giao dịch và độ phổ biến trọng số. Sau đó, Vo và đồng sự (2013) đề xuất một thuật toán quét dữ liệu một lần sử

dụng cấu trúc WIT-tree để khai thác hiệu quả mẫu phổ biến có trọng số. Nguyen và đồng sự (2016) đề xuất cấu trúc IWS để lưu trữ và xử lý tidset và tăng cường hiệu quả của việc khai thác mẫu phổ biến có trọng số trên các bộ dữ liệu thưa. Lee và đồng sự (2017) đề xuất các cấu trúc dựa trên các cấu trúc dựa trên FP-tree để khai thác mẫu phổ biến có trọng số bằng hai thuật toán là FWI-WSD và FWI-TCD.

2.2.2. Khai thác mẫu phổ biến đóng có trọng số

Hiện chỉ có nghiên cứu của Vo (2017) đưa ra phương pháp khai thác mẫu phổ biến đóng có trọng số với hai thuật toán là WIT-FWCI-Tid và WIT-FWCI-Diff tương ứng với các cấu trúc tidset và diffset trên cây WIT-tree.

2.2.3. Khai thác top-rank-k mẫu phổ biến có trọng số

Hiện không có nghiên cứu nào liên quan đến khai thác top-rank-k mẫu phổ biến có trọng số.

2.2.4. Khai thác mẫu phổ biến có trọng số theo luồng dữ liệu

Hiện không có nghiên cứu nào liên quan đến khai thác mẫu phổ biến có trọng số theo luồng dữ liệu.

2.3. Những vấn đề còn tồn tại

Qua quá trình tìm hiểu, phân tích và tổng hợp các công trình nghiên cứu liên quan, nghiên cứu sinh nhận thấy có nhiều vấn đề tồn tại cần được giải quyết. Trong phạm vi của luận án, nghiên cứu sinh tập trung một số vấn đề quan trọng sau:

- Tính đáp ứng phổ quát của thuật toán khai thác FWI trên các loại cơ sở dữ liệu khác nhau.
- Tính đáp ứng khả năng mở rộng của các thuật toán khai thác FWI khi kích thước các bộ dữ liệu tăng trưởng theo chiều ngang (số item tăng) và theo chiều dọc (số dòng dữ liệu tăng).
- Vấn đề khai thác mẫu phổ biến có trọng số theo định hướng người sử dụng như giới hạn về số lượng, giới hạn theo thời gian.

- Sự ít ỏi của các phương pháp khai thác FWCI hiện có.
- Sự khuyết thiếu của các phương pháp khai thác top-rank-k FWI hay khai thác FWI theo luồng dữ liệu để đáp ứng sát hơn nhu cầu của người sử dụng.

2.4. Mục tiêu của luận án

Nghiên cứu sinh đề ra mục tiêu chính của luận án là đề xuất một cấu trúc dữ liệu để lưu trữ hiệu quả cơ sở dữ liệu có trọng số, từ đó phát triển các thuật toán hiệu quả để khai thác FWI, FWCI, top-rank-k FWI và khai thác FWI theo luồng dữ liệu. Để đạt được mục tiêu này, luận án có hướng tiếp cận như sau:

- ***Hướng tiếp cận 1***: Khảo sát các phương pháp khai thác mẫu phổ biến cũng như khai thác FWI hiện có, kết hợp ưu điểm của các phương pháp tốt nhất hiện có.
- ***Hướng tiếp cận 2***: Khảo sát các dạng cây tìm kiếm, từ đó nghiên cứu và phát triển các chiến lược tỉa nhánh hiệu quả trên không gian tìm kiếm ứng viên, từ đó tăng hiệu quả về mặt thời gian khai thác.
- ***Hướng tiếp cận 3***: Tìm cách mô hình hóa về mặt toán học các đặc trưng riêng của cơ sở dữ liệu có trọng số, từ đó vận dụng vào các bài toán khai thác FWI, FWCI, top-rank-k FWI cũng như khai thác FWI theo luồng dữ liệu nói trên.

CHƯƠNG 3. KHAI THÁC MẪU PHỔ BIẾN CÓ TRỌNG SỐ BẰNG CẤU TRÚC WN-LIST

3.1. Giới thiệu

Trong nội dung chương này, luận án đề xuất cấu trúc dữ liệu WN-list (*weighted node-list*), một biến thể mở rộng của cấu trúc N-list (Deng et al. (2012)), để biểu diễn hiệu quả cơ sở dữ liệu có trọng số, từ đó đưa ra thuật toán NFWI để khai thác hiệu quả FWI. Đầu tiên, cơ sở dữ liệu có trọng số được đọc và nén trên cây WN-tree (biến thể của cây FP-tree (Grahne & Zhu (2005))), sau đó được trích xuất và khai thác trên các WN-list của các 1-FWI (FWI có 1 item). Các ứng viên của lớp *k* sẽ được tạo ra từ các ứng viên tương ứng của lớp *k*-1 bằng phương pháp chia để trị và phép giao WN-list có độ phức tạp chỉ là $O(n)$. Thuật toán NFWI dựa trên cấu trúc WN-list kết hợp được cả hai ưu điểm chính của các thuật toán hiện có là khả năng nén dữ liệu trên cây tốt và khả năng khai thác ứng viên trên không gian tìm kiếm linh hoạt, hiệu quả.

3.2. Cấu trúc dữ liệu WN-list

Định nghĩa 3.1 (WN-Tree): Cây WN-Tree là một cấu trúc bao gồm một nốt cha là "null" và các nốt con, trong đó mỗi nốt con là một bộ⟨*item-name, child-list, pre, pos, weight*⟩:

- *item-name* là tên của nốt, chính là tên của item đại diện cho nốt.
- *child-list* là danh sách các nốt con của nốt hiện tại.
- *pre* là thứ tự của nốt khi duyệt cây từ trên xuống trái sang.
- *pos* là thứ tự của nốt khi duyệt cây từ trên xuống phải sang.
- *weight* là khối lượng của nốt tính bằng tổng *tw* của các giao dịch đi qua nốt.

Định nghĩa 3.2 (WN-code): Một WN-code của một nốt trên cây WN-Tree là một bộ ba giá trị (*pre*, *pos*, *weight*) của nốt đó.

Định lý 3.1 (Deng et al. (2012)): Cho hai nốt khác nhau trên cây WN-Tree N_1 và N_2, N_1 là một tổ tiên của N_2 nếu và chỉ nếu $N_1.pre < N_2.pre$ và $N_1.pos > N_2.pos$.

Định lý 3.2 (Deng et al. (2012)): Cho hai WN-code $C_1(x_1, y_1, w_1)$ và $C_2(x_2, y_2, w_2)$, C_1 là một tổ tiên của C_2 nếu và chỉ nếu $x_1 < x_2$ và $y_1 > y_2$.

Định nghĩa 3.3 (WN-list của một item): Cho một cây WN-Tree, WN-list của một item *X*, ký hiệu là *WL*(*X*), là dãy có thứ tự các WN-code của các nốt đại diện cho item *X* đó trên cây WN-Tree, trong đó các WN-code trong dãy được sắp xếp tăng dần theo giá trị *pre* của chúng.

Định lý 3.3: Cho A là một item với WN-list của *A* là $WL(A) = \{(x_1, y_1, w_1), (x_2, y_2, w_2), \dots, (x_n, y_n, w_n)\}$. Độ phổ biến trọng số của *A* là $ws(A)$ được tính như sau:

$$ws(A) = \frac{\sum_{i=1}^{n} w_i}{\sum_{t_k \in T} tw(t_k)} \quad (3.1)$$

Định nghĩa 3.4 (WN-list của một *k*-itemset): Cho *XA* và *XB* là hai (*k*-1)-itemset có cùng phần tiền tố là *X*, trong đó *A* xếp sau *B* theo thứ tự trong I_1. *WL*(*XA*) và *WL*(*XB*) là hai WN-list tương ứng của *XA* và *XB*. Ta có *WL*(*XAB*) là WN-list của *k*-itemset *XAB* được xác định như sau:

1. Với mỗi cặp $C_i(x_i, y_i, w_i) \in WL(XA)$ và $C_j(x_j, y_j, w_j) \in WL(XB)$, nếu C_j là tổ tiên của C_i thì ta thêm (x_j, y_j, w_i) vào *WL*(*XAB*).

2. Duyệt *WL*(*XAB*) và tổ hợp các WN-code có cùng (*pre*, *pos*) thành một WN-code mới có weight là tổng của các giá trị weight trong các WN-code đang xét.

Định lý 3.4: Cho *X* là một itemset với WN-list của *X* là $WL(X) = \{(x_1, y_1, w_1), (x_2, y_2, w_2), \dots, (x_n, y_n, w_n)\}$. Độ phổ biến trọng số của *X* là $ws(X)$ được tính như sau:

$$ws(X) = \frac{\sum_{i=1}^{n} w_i}{\sum_{t_k \in T} tw(t_k)} \quad (3.2)$$

Định lý 3.6: Cho itemset *P* và item *i* ($i \notin P$), nếu $ws(P) = ws(P \cup \{i\})$ thì với mọi itemset *A* thỏa mãn điều kiện ($A \cap P = \emptyset \wedge i \notin A$, ta có: $ws(A \cup P) = ws(A \cup P \cup \{i\})$.

3.3. Thuật toán khai thác mẫu phổ biến có trọng số dựa trên WN-list

Thuật toán 3.3: Thuật toán NFWI

Input: The weighted database *WD* and a threhold *minws*
Output:*FWI*, the set of all frequent weighted itemsets.
Method name: ***NFWI***(*WD*, *minws*)

1. Call ***Construction_WN_Tree***(*WD*, *minws*) to generate *Tree* and I_1
2. Scan *Tree* to generate WN-lists of items in I_1
3. Let $FWI \leftarrow I_1$
4. Call ***Find_FWI***(I_1, ∅, ∅)
5. return *FWI*

Procedure *Find_FWI*(L_k, Pre_k, S_k)

1. $Pre_{next} = Pre_k$
2. for $i = L_k.size$ downto 2 do
3. $Pre_{next} \leftarrow L_i$
4. $L_{next} = \emptyset$
5. $S_{next} = S_k$
6. for $j = i$-1 downto 1 do
7. NS_{ij} = ***WL_Intersection***(WL_i, WL_j)
8. if $ws(WL_{ij}) \geq minws$ then
9. if $ws(WL_{ij}) = ws(WL_i)$ then $S_{next} \leftarrow L_j$
10. else $FWI \leftarrow \{Pre_k \cup \{L_i\} \cup \{L_j\}\}$
11. $I_{next} \leftarrow L_i$
12. if $S_{next} \neq \emptyset$ then ***Find_FWI_notWL***(Pre_{next}, S_{next})
13. if $L_{next} \neq \emptyset$ then ***Find_FWI***(L_{next}, Pre_{next}, S_{next})
14. remove last item of Pre_{next}
15. $Pre_{next} \leftarrow L_1$
16. if $S_k \neq \emptyset$ then ***Find_FWI_notWL***(Pre_{next}, S_k)

Procedure *Find_FWI_notWL*(*Pre*, *S*)

1. *Childs* = all subset of *S*
2. for each $s \in Childs$ do
3. $FWI \leftarrow \{Pre \cup s\}$

3.4. Kết quả thực nghiệm

Thuật toán NFWI được so sánh với các thuật toán khai thác FWI hiện có bao gồm: WIT-FWIs-Diff (Vo et al. (2013)), IWS (Nguyen et al. (2016)), FWI-WSD

và FWI-TCD (Lee et al. (2017)) trên các mặt thời gian chạy, bộ nhớ sử dụng và khả năng mở rộng. Các bộ dữ liệu Chess, PAMP, BMS-POS, Retail, Kosarak, Chainstore và Sale_Fact_sync được sử dụng để thực nghiệm thời gian chạy và bộ nhớ sử dụng. Dữ liệu dùng để thực nghiệm khả năng mở rộng được trích xuất từ bộ dữ liệu Chainstore. Kết quả thực nghiệm cho thấy thuật toán đề xuất NFWI cho kết quả chạy nhanh nhất trên cả hai loại dữ liệu (thưa và dày). Thuật toán NFWI sử dụng ít bộ nhớ hơn ở hầu hết các bộ dữ liệu thực nghiệm như Chess, PAMP, BMS-POS và Sale_Fact_Sync. Thực nghiệm cũng cho thấy thuật toán NFWI cho thấy khả năng mở rộng tốt về mặt thời gian chạy, trong khi khả năng mở rộng về bộ nhớ sử dụng cũng khá tốt.

3.5. Đánh giá về phương pháp đề xuất

Trong chương này, luận án đã giới thiệu cấu trúc WN-list để biểu diễn CSDL có trọng số. Luận án cũng đề xuất một số định lý để tính toán độ phổ biến trọng số của itemset, cũng như xác định nhanh giá trị của nó trong một số trường hợp mà không cần thực hiện phép giao WN-list. Dựa vào đó để xây dựng thuật toán NFWI để khai thác nhanh mẫu phổ biến có trọng số. Sự hiệu quả của thuật toán NFWI thu được là nhờ các ưu điểm của cấu trúc WN-list. Các ưu điểm đó là: khả năng nén dữ liệu cao khiến kích thước WN-list nhỏ, dễ dàng tính độ phổ biến trọng số thông qua quét WN-list và thuật toán giao hai WN-list chỉ có độ phức tạp tuyến tính. Các thực nghiệm trên nhiều loại CSDL đã cho thấy NFWI hoạt động hiệu quả hơn các thuật toán khai thác FWI hiện có.

Kết quả nghiên cứu trong chương này được công bố trong các công trình [CT.1] và [CT.6]. Trong đó thuật toán NFWI được đề xuất trong [CT.1] hiện được xem là thuật toán tốt nhất hiện có trong khai thác mẫu phổ biến có trọng số.

CHƯƠNG 4. KHAI THÁC MẪU PHỔ BIẾN ĐÓNG CÓ TRỌNG SỐ BẰNG CẤU TRÚC WN-LIST VÀ CHIẾN LƯỢC TỈA NHÁNH SỚM

4.1. Giới thiệu

Trong nội dung chương này, luận án đề xuất thuật toán NFWCI để khai thác mẫu phổ biến đóng có trọng số dựa trên cấu trúc dữ liệu WN-list. Nghiên cứu sinh đã giới thiệu khái niệm về phép toán tổ tiên WN-list và đề xuất một định lý để loại bỏ nhanh các ứng viên không thỏa mãn dựa trên phép toán tổ tiên WN-list. Thuật toán NFWCI cũng áp dụng tính chất của phép giao WN-list khi kết hợp hai ứng viên không thỏa mãn trên cùng một lớp tương đương để giảm kích thước WN-list của ứng viên kết hợp, từ đó giúp cho việc tăng tốc tính toán.

4.2. Quan hệ tổ tiên WN-list và chiến lược tỉa nhánh sớm dựa trên quan hệ tổ tiên WN-list

Định nghĩa 4.1 (Quan hệ tổ tiên WN-list): Cho PA_1 và PA_2 là hai mẫu phổ biến có trọng số (A_1 đứng trước A_2 theo thứ tự trong I_1 và P có thể rỗng). Ta nói $WL(PA_1)$ là tổ tiên của $WL(PA_2)$, ký hiệu là $WL(PA_1) \rhd WL(PA_2)$ nếu và chỉ nếu $\forall C_i \in WL(PA_2)$, $\exists C_j \in WL(PA_1)$ sao cho C_j là một tổ tiên của C_i.

Định lý 4.1: Cho PA_1 và PA_2 là hai mẫu phổ biến có trọng số (A_1 đứng trước A_2 theo thứ tự trong I_1 và P có thể rỗng).

a. Nếu $WL(PA_1) \rhd WL(PA_2)$, thì PA_2 không phải là mẫu phổ biến đóng có trọng số.

b. Nếu $WL(PA_1) \rhd WL(PA_2)$ và $ws(PA_1) = ws(PA_2)$, thì PA_1 và PA_2 không phải là mẫu phổ biến đóng có trọng số.

4.3. Thuật toán khai thác mẫu phổ biến đóng có trọng số dựa trên WN-list

Thuật toán 4.1: Thuật toán NFWCI

Input: *WD* and *minws*
Output: *FWCIs*
Method name: ***NFWCI***(*WD, minws*)

1. ***Construction_WN_Tree***(*WD*, *minws*)
2. Scan *WN-Tree* to find {*SWL*(A_i), $A_i \in I_1$}
3. Set *FWCIs* ← ∅

4. ***Find_FWCI***(I_1)
5. Return *FWCIs*

Procedure *Find_FWCI*(*Is*)
6. for $i \leftarrow$ |*Is*|-1 down to 0 do
7. $I_{next} = \emptyset$
8. for $j \leftarrow i$ -1 to 0 do
9. $WL(X_iX_j) \leftarrow$ ***WL_Intersection***($WL(X_i)$, $WL(X_j)$)
10. if $WL(X_i) \subseteq WL(X_j)$ then
11. $X_i = X_i \cup X_j$
12. for each X_k in I_{next}
13. update $X_k = X_k \cup X_j$
14. end for
15. if $ws(X_i) = ws(X_j)$ then
16. Remove X_j
17. i -
18. end if
19. else
20. if $ws(X_iX_j) \geq minws$ and $X_iX_j \notin FWCIs$ then
21. $I_{next} \leftarrow X_iX_j$
22. end for
23. ***Find_FWCI***($FWCI_{next}$)
24. $FWCIs \leftarrow X_i$
25. end for

4.4. Kết quả thực nghiệm

Các thực nghiệm được tiến hành để so sánh sự hiệu quả của thuật toán đề xuất NFWCI với thuật toán hiện có WIT-FWCI-Diff (Vo (2017)), cũng như so sánh với thuật toán cơ sở DiffNFWCI, được áp dụng cùng phương pháp khai thác nhưng dựa trên cấu trúc dữ liệu DiffNodeset (Deng (2016)). Lí do để tạo ra và sử dụng thuật toán DiffNFWCI cho việc so sánh là bởi vì cấu trúc DiffNodeset là một cấu trúc dữ liệu mới nhất trong khai thác mẫu phổ biến.

Dữ liệu thực nghiệm về thời gian chạy và bộ nhớ sử dụng bao gồm 09 bộ dữ liệu là Accidents, Connect, PAMP, Chess, PowerC, OnlineRetail, Retail, Kosarak và Sale_Fact_Sync. Dữ liệu thực nghiệm về khả năng mở rộng được lấy từ hai bộ dữ liệu lớn là Chainstore và Susy.

Kết quả thực nghiệm cho thấy thuật toán NFWCI chạy nhanh hơn nhiều so với các thuật toán WIT-FWCI-Diff và DiffNFWCI trên các bộ dữ liệu Accidents, Connect, Retail, PALM, Sale_Fact_sync, PowerC và Kosarak. Đặc biệt NFWCI thể hiện sự vượt trội trên các cơ sở dữ liệu lớn như Kosarak và PAMP. Về bộ nhớ sử dụng, thuật toán NFWCI là thuật toán hiệu quả nhất về mặt bộ nhớ sử dụng khi so sánh với hai thuật toán còn lại là DiffNFWCI và WIT-FWCI-Diff. Về khả năng mở rộng, NFWCI tốt hơn WIT-FWCI-Diff khi số lượng giao dịch tăng lên, và kém hơn WIT-FWCI-Diff khi số danh mục tăng lên. Khả năng mở rộng về bộ nhớ sử dụng của NFWCI và DiffNFWCI là tương đương và cùng kém hơn so với WIT-FWCI-Diff.

4.5. Đánh giá phương pháp đề xuất

Đóng góp của nội dung chương 4 đó là đề xuất một thuật toán hiệu quả là NFWCI để khai thác mẫu phổ biến đóng có trọng số. Thuật toán NFWCI được xây dựng dựa trên cấu trúc WN-list. Dựa trên việc giới thiệu khái niệm phép toán tổ tiên WN-list, một định lý được đề xuất để loại bỏ nhanh chóng các ứng viên không thỏa mãn trong quá trình khai thác mẫu phổ biến đóng có trọng số. Kết quả thực nghiệm được thực hiện để so sánh hiệu quả giữa thuật toán đề xuất so với thuật toán hiện có WIT-FWCI-Diff (Vo (2017)) và một thuật toán áp dụng phương pháp tiếp cận tương tự nhưng dựa trên cấu trúc dữ liệu mới nhất trong khai thác mẫu phổ biến DiffNodeset (Deng (2016)). Kết quả thực nghiệm chỉ ra rằng thuật toán đề xuất có hiệu quả vượt trội so với các thuật toán còn lại trên các mặt như thời gian chạy, bộ nhớ sử dụng và khả năng mở rộng.

Kết quả nghiên cứu trong chương 4 đã được công bố ở các công trình [CT.2]. Thuật toán NFWCI được đề xuất hiện được xem là thuật toán tốt nhất hiện có trong khai thác mẫu phổ biến đóng có trọng số.

CHƯƠNG 5. KHAI THÁC TOP-RANK-K MẪU PHỔ BIẾN CÓ TRỌNG SỐ BẰNG CẤU TRÚC WN-LIST

5.1. Giới thiệu

Trong nội dung chương này, luận án giới thiệu và mô hình hóa bài toán khai thác top-rank-k mẫu phổ biến có trọng số. Khai thác top-rank-k mẫu phổ biến có trọng số là xác định các mẫu phổ biến có độ phổ biến trọng số nằm trong *k* ngưỡng lớn nhất, nhằm thỏa mãn nhu cầu của người dùng mà không mất thời gian đi xem xét toàn bộ các mẫu phổ biến có trọng số. Các chiến lược tăng ngưỡng và tỉa nhánh sớm đã được đề xuất đưa ra thuật toán TFWIN$^+$ khai thác top-rank-k mẫu phổ biến có trọng số hiệu quả nhất.

5.2. Phương pháp đề xuất

Chiến lược 5.1 (RTS - *raising threshold strategy*): Đầu tiên, tập top-rank-k ký hiệu là *TR* là rỗng, và như vậy ngưỡng khởi tạo được đặt là *minws* = 0. Khi một tập FWI ký hiệu là *I* được tìm thấy, nếu độ hỗ trợ trọng số của nó không nhỏ hơn ngưỡng *minws*, thì *I* được thêm vào *TR*. Nếu *TR* có nhiều hơn *k* rank, ngưỡng *minws* có thể được tăng lên tới độ hỗ trợ trọng số của mục cuối trong *TR*.

Định lý 5.1: Cho *XA* và *XB* là hai tập FWIs cùng lớp tương đương. Nếu *ws*(*XA*) < *minws* hoặc *ws*(*XB*) < *minws* thì *XAB* không thuộc về *TR*.

Chiến lược 5.2 (EPS - *early pruning strategy*)**:** Trong suốt thủ tục ***TFWIN_Plus_CandidateGen***, thuật toán TFWIN$^+$ sẽ không tạo ra ứng viên của hai FWI c_u và c_v với $c_u.ws$ < *minws* hoặc $c_v.ws$ < *minws* dựa trên Định lý 5.1. Bên cạnh đó, thuật toán cũng loại bỏ ứng viên có độ hỗ trợ trọng số nhỏ hơn ngưỡng trong thủ tục này.

Thuật toán 5.4. TFWIN$^+$ algorithm

Input: a weighted database *WD* and a threshold *k*
Output: the complete set of top-rank-*k* FWI (*TR*)

1 Let *TR* ← ∅ and C_k ← ∅
2 Build WN-tree from *WD*
3 Determine I_1 and its WN-lists

```
4   Sort I1 in weighted support ascending order
5   for j = 0 to m-1 do
6       if TR.count > 0 and TR.last_entry.ws = I1[j].ws then
7           Add I1[j] to TR.last_entry.list and I1[j] to Ck
8       else
9           Let R is new rank, R.ws = I1[j].ws and R.list.add(I1[j])
10          Add R to TR and I1[j] to Ck
11  end for
12  Let minws ←TR.last_entry.ws // using RTS
13  while Ck ≠ ∅ do
14      C ← TFWIN_Plus_CandidateGen(Ck)
15      Sort C in weighted support ascending order
16      Let Ck ← ∅, x ← 0 and l ← 0
17      while l < C.count and x < TR.count do
18          if TR[x].ws = C[l].ws then
19              Add C[l] to TR[x].list and C[l] to Ck
20              l++
21          else if TR[x].ws > C[l].ws then
22                  Let R be the new rank
23                  R.ws = C[l].ws and R.list.add(C[l])
24                  Insert R to TR at position x
25              if TR.count > k then
26                  Remove the last tuple in TR
27                  Update minws ←TR.last_entry.ws // using RTS
28                  Add C[l] to Ck
29                  l++
30              else
31                  x++
32      end while
33      if TR.count < k then
34          Let t ← min(k – TR.count, C.count - l + 1)
35          For (x = l; x < l + t; x++) do
36              R.ws = C[x].ws and R.list.add(C[x])
37              Add R to TR
38  end while
```

Procedure *TFWIN_Plus_CandidateGen*(*Ck*)

```
1   Let Cnext ← ∅
2   for each cu ∈ Ck do
3     for each cv ∈ Ck with u < v do
```

4 if c_u and c_v are in the same equivalence class then
5 if $c_u.ws < minws$ or $c_v.ws < minws$ then Continue
6 $c.WN\text{-}list = c_v.WN\text{-}list$ interest with $c_u.WN\text{-}list$
7 $c.ws = \frac{\sum_{i=1}^{n} w_i}{TTW}$
8
9 if $c.ws < minws$ then Continue // Using EPS
10 $c = c_u \cup c_v$
11 Add c to C_{next}
Return C_{next}

5.3. Kết quả thực nghiệm

Thực nghiệm so sánh thời gian chạy của các thuật toán TFWIT, TFWID, TFWIN và TFWIN$^+$ trên bốn bộ dữ liệu thực nghiệm Accident, Chess, Connect và Pumsb. Kết quả thực nghiệm cho thấy TFWIN$^+$ là thuật toán tốt nhất trong khai thác top-rank-k trên mặt thời gian chạy. TFWIN$^+$ cũng vượt trội TFWIT, TFWID và TFWIN về mặt bộ nhớ sử dụng.

5.4. Đánh giá phương pháp đề xuất

Nội dung chương 5 giới thiệu bài toán khai thác top-rank-k FWI, với một cách tiếp cận kết hợp hai pha khai thác và xếp hạng vào một pha mà không cần tìm tất cả FWI, từ đó tăng sự hiệu quả của các hệ thống thông minh. Các chiến lược tăng ngưỡng và tỉa nhánh sớm được áp dụng để tăng tính hiệu quả của quá trình khai thác top-rank-k FWI. Áp dụng các chiến lược đó với sự hỗ trợ của một định lý, từ đó phát triển thuật toán TFWIN$^+$ để khai thác hiệu quả hơn top-rank-k FWI. Cuối cùng, các thực nghiệm được thực hiện trên các bộ dữ liệu chuẩn để so sánh về mặt thời gian chạy và bộ nhớ sử dụng của các thuật toán được đề xuất nói trên. Kết quả thực nghiệm cho thấy thuật toán TFWIN$^+$ là thuật toán tốt nhất trong khai thác top-rank-k FWI về cả mặt thời gian chạy và bộ nhớ sử dụng.

Nội dung nghiên cứu trong chương 5 được công bố trong công trình [CT.3] và thuật toán TFWIN$^+$ hiện là thuật toán tốt nhất hiện có trong khai thác top-rank-k mẫu phổ biến có trọng số.

CHƯƠNG 6. KHAI THÁC MẪU PHỔ BIẾN CÓ TRỌNG SỐ THEO LUỒNG DỮ LIỆU BẰNG MÔ HÌNH CỬA SỔ TRƯỢT

6.1. Giới thiệu

Trong chương này, luận án đề xuất một thuật toán để khai thác mẫu phổ biến có trọng số trên dữ liệu tăng trưởng. Đầu tiên, một mô hình khai thác mẫu phổ biến có trọng số theo luồng dữ liệu được giới thiệu dựa trên mô hình cửa sổ trượt. Sau đó, cấu trúc cây SWN-tree, một phiên bản cải tiến của cây WN-tree [CT.1], được đề xuất để xây dựng cây dữ liệu trên cửa sổ ban đầu. Một thuật toán duy trì cây SWN-tree khi trượt trên các cửa sổ dữ liệu tiếp theo cũng được xây dựng, nhằm mục đích tối ưu hóa quá trình cập nhật dữ liệu theo luồng dữ liệu, hạn chế việc đọc lại toàn bộ dữ liệu của cửa sổ trước đó. Cuối cùng, một thuật toán tên là FWPDOS (*Frequent weighted patterns over data stream*) được đề xuất để khai thác mẫu phổ biến có trọng số theo luồng dữ liệu dựa trên mô hình cửa sổ trượt. Kết quả thực nghiệm trên các mặt như thời gian xây dựng và bảo trì cây, thời gian khai thác mẫu và khả năng mở rộng cho thấy thuật toán đề xuất có hiệu quả vượt trội so với thuật toán cơ sở NFWI [CT.1] khi áp dụng trên luồng dữ liệu.

6.2 Thuật toán khai thác mẫu phổ biến có trọng số theo luồng dữ liệu bằng mô hình cửa sổ trượt

Định nghĩa 6.1 (Cây SWN-tree): Cây SWN-tree là một cây bao gồm 6 thành phần ⟨*name*, *weight*, *pre*, *pos*, *child-list*, *parent*⟩. Trong đó *name* là định danh item, *weight* là tổng trọng số dòng dữ liệu chứa item, *pre* là thứ tự duyệt trước, *pos* là thứ tự duyệt sau, *child-list* là danh sách nốt con và *parent* là nốt cha.

Thuật toán 6.2. Thuậtt toán duy trì cây SWN-tree

Input: SWN-tree R and T – new inserted transactions

Output: Updated SWN-tree for the new window

Method name: ***Maintaining_SWN_Tree***(*R, T*)

```
1  for each T_i in T do
2      Call Insert_tree(T_i, R)
```

```
3  for i ← 1 to |T| do
4     Let l be the first element in TAIL, N = l.Parent and t = l.TID do
5        N.weight = N.weight – tw(t)
6        if N.weight = 0 then
7           Let N' = N.Parent
8           Delete N
9           Let N = N'
10       else
11       N = N.Parent
12    while (N is not the ROOT of R)
13    Delete l in TAIL
14 Return R
```

Thuật toán FWPODS được đề xuất dựa trên kết hợp giữa thuật toán NFWI và thuật toán ***Maintaining_SWN_Tree*** nói trên để khai thác FWI thông qua luồng dữ liệu bằng mô hình cửa sổ trượt.

6.3. Kết quả thực nghiệm

Thực nghiệm so sánh thời gian xử lý của thuật toán FWPODS và thuật toán NFWI [CT.1] trong việc xây dựng và duy trì cây trên 5 bộ dữ liệu Accidents, Connect, Kosarak, Pumsb và Retail. Tổng thời gian xử lý đối với lô cửa sổ trượt của FWPODS ít hơn nhiều so với tổng thời gian tương ứng của NFWI. FWPODS thể hiện khả năng mở rộng tốt trên các mặt thời gian chạy khi kích thước cửa sổ trượt và panel tăng lên.

6.4. Đánh giá phương pháp đề xuất

Trong chương này luận án đã phát triển thuật toán FWPODS [CT.4] để khai thác mẫu phổ biến có trọng số theo luồng dữ liệu. Đầu tiên, mô hình khai thác mẫu phổ biến có trọng số theo luồng dữ liệu được giới thiệu thông qua cách tiếp cận cửa sổ dữ liệu trượt. Sau đó, cấu trúc SWN-tree được đề xuất từ việc cải tiến cấu trúc WN-tree [CT.1] để có thể duy trì thông tin hiệu quả theo luồng dữ liệu. Thuật toán FWPODS được phát triển dựa trên cây SWN-tree này. Thực nghiệm được thực hiện và kết quả thực nghiệm cho thấy sự hiệu quả của phương pháp đề xuất khi khai thác mẫu phổ biến có trọng số theo mô hình cửa sổ trượt.

CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

7.1. Kết luận

Trong luận án này, nghiên cứu sinh nghiên cứu, đề xuất cấu trúc dữ liệu và phát triển một số thuật toán hiệu quả để giải quyết một số bài toán khai thác mẫu và mẫu đóng trên cơ sở dữ liệu có trọng số.

Đóng góp đầu tiên của luận án là đề xuất cấu trúc dữ liệu WN-list và thuật toán NFWI tương ứng để khai thác mẫu phổ biến có trọng số một cách hiệu quả. WN-list là một mở rộng của cấu trúc N-list để biểu diễn dữ liệu có trọng số. Sự vượt trội của cấu trúc WN-list đến từ những ưu điểm sau: (1) Dữ liệu được nén trên cây dạng FP-tree; (2) Quan hệ tổ tiên của các nốt trên cây được xác định bằng cách so sánh các giá trị *pre* và *pos* của các nốt; (3) Nội dung của cây được chuyển sang dạng tuyến tính theo dạng $\{(pre_i, pos_i, weight_i)\}$ tương ứng với WN-list của các tập danh mục một phần tử và quá trình khai thác chỉ thực hiện trên đó; (4) Độ hỗ trợ trọng số *ws* của một tập danh mục được tính dễ dàng bằng tổng của các giá trị $weight_i$ trong WN-list của nó; (5) Độ phức tạp của phép giao WN-list chỉ là $O(n)$ và kích thước của WN-list kết quả được giảm thiểu đáng kể nhờ vào việc kết hợp các phần tử chung (*pre*, *pos*) lại với nhau. Thuật toán NFWI đã tận dụng các ưu điểm của cấu trúc WN-list để nén dữ liệu trên cây, sau đó trích xuất và khai thác trên các WN-list của các 1-itemset phổ biến. Các ứng viên lớp k được tạo ra tương ứng từ lớp (k-1) bằng phương pháp chia để trị và phép giao WN-list với độ phức tạp tuyến tính. Một số định lý đã được đề xuất để tính độ phổ biến trọng số của một itemset dựa trên WN-list, cũng như xác định nhanh giá trị của nó trong một số trường hợp mà không cần thực hiện phép giao WN-list. Kết quả thực nghiệm trên nhiều loại cơ sở dữ liệu cho thấy thuật toán NFWI hoạt động hiệu quả hơn các thuật toán khai thác mẫu phổ biến có trọng số hiện có.

Đóng góp thứ hai của luận án là đề xuất thuật toán mới NFWCI khai thác hiệu quả mẫu phổ biến đóng có trọng số dựa trên cấu trúc WN-list và chiến lược

tỉa nhánh nhanh. Nghiên cứu sinh đã giới thiệu khái niệm về phép toán tổ tiên WN-list và đề xuất một định lý để loại bỏ nhanh các ứng viên không thỏa mãn dựa trên phép toán tổ tiên WN-list. Thuật toán NFWCI cũng áp dụng tính chất của phép giao WN-list khi kết hợp hai ứng viên không thỏa mãn trên cùng một lớp tương đương để giảm kích thước WN-list của ứng viên kết hợp, từ đó tăng tốc tính toán trong các bước kế tiếp. Thực nghiệm trên nhiều cơ sở dữ liệu thưa và dày cho thấy sự hiệu quả của thuật toán NFWCI so với thuật toán hiện có trong khai thác mẫu phổ biến đóng có trọng số.

Đóng góp thứ ba là luận án đã giới thiệu và mô hình hóa bài toán khai thác top-rank-k mẫu phổ biến có trọng số. Khai thác top-rank-k mẫu phổ biến có trọng số là xác định các mẫu phổ biến có độ phổ biến trọng số nằm trong *k* ngưỡng lớn nhất, nhằm mục đích thỏa mãn nhu cầu của người dùng mà không mất thời gian đi xem xét toàn bộ các mẫu phổ biến có trọng số. Ba thuật toán cơ sở TFWIT, TFWID và TFWIN được đề xuất để giải quyết bài toán khai thác top-rank-k mẫu phổ biến có trọng số tương ứng dựa trên ba cấu trúc dữ liệu hiện hành là tidset, diffset và WN-list. Các chiến lược tăng ngưỡng và tỉa nhánh sớm đã được đề xuất để cải tiến thuật toán TFWIN, từ đó đề xuất thuật toán TFWIN$^+$ khai thác top-rank-k mẫu phổ biến có trọng số. Kết quả thực nghiệm trên các mặt thời gian chạy và bộ nhớ sử dụng cho thấy thuật toán TFWIN$^+$ là thuật toán hiệu quả nhất trong khai thác top-rank-k mẫu phổ biến có trọng số.

Cuối cùng, luận án đã mô hình hóa bài toán khai thác mẫu phổ biến có trọng số theo luồng dữ liệu bằng cách tiếp cận cửa sổ dữ liệu trượt. Khai thác mẫu phổ biến có trọng số theo luồng dữ liệu giúp cho kết quả khai thác tiếp cận với nhu cầu của người dùng theo giới hạn về thời gian với chi phí bộ nhớ tối thiểu. Cấu trúc cây SWN-tree đã được đề xuất trên cơ sở cải tiến cấu trúc WN-tree, để có thể xây dựng và duy trì hiệu quả cây lưu dữ liệu khi trượt theo các cửa sổ dữ liệu. Thuật toán FWPODS được xây dựng trên nền tảng cây SWN-tree để có thể khai thác

hiệu quả mẫu phổ biến có trọng số theo luồng dữ liệu, và kết quả các thực nghiệm trên các bộ dữ liệu thưa và dày đã chứng minh tính hiệu quả của thuật toán đề xuất.

Các kết quả nghiên cứu liên quan các nội dung trên đã được công bố trên bốn tạp chí chuyên ngành uy tín ([CT.1], [CT.2], [CT.3] và [CT.4]) và các hội thảo chuyên ngành được phản biện độc lập ([CT.5], [CT.6] và [CT.7]).

7.2. Hướng phát triển

Hướng phát triển nghiên cứu trong tương lai sẽ tập trung giải quyết một số bài toán khai thác mẫu trên cơ sở dữ liệu có trọng số như là: khai thác tập đối đại có trọng số, khai thác mẫu phổ biến đóng có trọng số trên cơ sở dữ liệu tăng trưởng, khai thác mẫu phổ biến có trọng số trên cơ sở dữ liệu không chắc chắn và nghiên cứu triển khai các giải pháp khai thác mẫu phổ biến có trọng số trên hệ thống multicore và hệ thống phân tán. Nghiên cứu sinh sẽ nghiên cứu cải tiến và áp dụng cấu trúc WN-list để giải quyết các lớp bài toán khai thác mẫu trên cơ sở dữ liệu có trọng số, một dạng cơ sở dữ liệu khá thông dụng hiện hành trong các ứng dụng IoT. Bên cạnh đó, nghiên cứu sinh cũng sẽ nghiên cứu việc áp dụng nội dung nghiên cứu vào giải quyết các bài toán ứng dụng cụ thể như là khai thác dữ liệu văn bản và khai thác dữ liệu không gian. Cuối cùng, nghiên cứu sinh sẽ nghiên cứu triển khai các ứng dụng sử dụng nền tảng khai thác mẫu có trọng số như khai thác đồ thị, khai thác mạng xã hội, khai thác dữ liệu văn bản và khai thác dữ liệu IoT.

DANH MỤC CÔNG BỐ KHOA HỌC

Các công trình chính

[CT.1] **Bui, H.**, Vo, B., Nguyen, H., Nguyen-Hoang, T. A., & Hong, T. P. (2018). A weighted N-list-based method for mining frequent weighted itemsets. *Expert Systems with Applications*, vol. *96*, 388-405 (SCIE, 2018 IF= 4.292, Q1).

[CT.2] **Bui, H.**, Vo, B., Nguyen-Hoang, T. A., & Yun, U. (2020). Mining frequent weighted closed itemsets using the WN-list structure and an early pruning strategy. Applied Intelligence, vol. 51, no. 3, 1439-1459. (SCIE, 2018 IF= 3.325, Q2).

[CT.3] Vo, B., **Bui, H.**, Vo, T., & Le, T. (2020). Mining top-rank-k frequent weighted itemsets using WN-list structures and an early pruning strategy. Knowledge-Based Systems, vol. 201, 106064. (SCIE, 2019 IF= 5.921, Q1).

[CT.4] **Bui, H.**, Nguyen-Hoang, T. A., Vo, B., Nguyen, H., & Le, T. (2021). A Sliding Window-based Approach for Mining Frequent Weighted Patterns over Data Streams. *IEEE Access*, vol. 9, 56318-56329. (SCIE, 2019 IF=3.745, Q1).

Các công trình có liên quan

[CT.5] **Bui, H.**, Vo, B., & Nguyen, H. (2016). WUN-miner: A new method for mining frequent weighted utility itemsets. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 001365-001370). IEEE.

[CT.6] **Bùi Danh Hường**, Võ Đình Bảy, Nguyễn Duy Hàm. Khai thác tập phổ biến có trọng số dựa trên cấu trúc N-list. *Hội nghị Quốc gia lần thứ IX về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR). Cần Thơ, 2016.*

[CT.7] **Bùi Danh Hường**, Võ Đình Bảy, Nguyễn Hoàng Tú Anh. "SWUN-Miner: Phương pháp mới khai thác tập phổ biến có trọng số hữu ích". *Hội thảo quốc gia lần thứ XX: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – Quy Nhơn, 2017.*