# MỞ ĐẦU

Khai phá luật kết hợp là một trong những kỹ thuật quan trọng nhất trong khai phá dữ liệu. Mục đích chính của khai phá luật kết hợp là tìm ra mối quan hệ giữa các phần tử khác nhau trong cơ sở dữ liệu. Bài toán khai phá tập luật kết hợp gồm hai bài toán con đó là khai phá tập phổ biến và sinh luật kết hợp. Trong đó, bài toán khai phá tập phổ biến đã thu hút được nhiều nhà nghiên cứu trong nước và thế giới quan tâm. Nhưng khai phá tập phổ biến truyền thống trong thực tế vẫn còn nhiều hạn chế, không đáp ứng được nhu cầu của người sử dụng như đánh giá sự quan trọng của từng phần tử trong từng giao dịch hay trong cơ sở dữ liệu. Để khắc phục những hạn chế của khai phá tập phổ biến truyền thống, nhiều nhà nghiên cứu đã đề xuất mô hình mở rộng trong đó có tính đến mức độ quan trọng khác nhau của các phần tử trong cơ sở dữ liệu như: khai phá tập phổ biến có trọng số - WFI; khai phá tập lợi ích cao - HUI.

Một trong những thách thức trong khai phá tập phổ biến có trọng số và tập lợi ích cao đó là tập phổ biến có trọng số, tập lợi ích cao không có tính chất đóng - tính chất làm giảm số lượng ứng viên được sinh ra và không gian tìm kiếm. Hầu hết các thuật toán khai phá tập lợi ích cao đều sử dụng tính chất đóng của lợi ích trọng số giao dịch – TWU do Liu và cộng sự công bố năm 2005. Tuy nhiên, ngưỡng TWU vẫn còn khá cao so với lợi ích thực tế của các tập phần tử, do đó vẫn còn phát sinh một số lượng lớn các ứng viên không cần thiết, do đó tiêu tốn thời gian và không gian tìm kiếm.

Trên cơ sở những nghiên cứu, nhận xét và đánh giá ở trên, nghiên cứu sinh đã chọn đề tài “***Nghiên cứu phát triển mô hình, thuật toán khai phá tập phần tử có trọng số và lợi ích cao***” làm đề tài nghiên cứu cho luận án tiến sĩ của mình.

**Mục tiêu nghiên cứu**

* Nghiên cứu các thuật toán khai phá tập phổ biến, tập phổ biến có trọng số và tập lợi ích cao.
* Xây dựng mô hình, điều kiện, cấu trúc dữ liệu nhằm giảm không gian tìm kiếm và dựa trên cơ sở đó để xây dựng các thuật toán khai phá tập phổ biến có trọng số và tập lợi ích cao.

# Tổng quan về khai phá tập phổ biến

## Giới thiệu chung

Khai phá tập phổ biến là tìm ra các tập phần tử có số lần xuất hiện lớn hơn một ngưỡng hỗ trợ tối thiểu (minsupp). Tuy nhiên, khai phá tập phổ biến có những hạn chế. Thứ nhất, nó xử lý tất cả các phần tử có tầm quan trọng như nhau. Thứ hai, trong một giao dịch mỗi phần tử chỉ có trạng thái xuất hiện hoặc không xuất hiện. Rõ ràng những hạn chế này làm cho bài toán khai phá tập phổ biến truyền thống không phù hợp với các cơ sở dữ liệu thực tế, ví dụ như trong cơ sở dữ liệu của siêu thị, mỗi mặt hàng có tầm quan trọng hay giá cả khác nhau, số lượng mua các mặt hàng trong mỗi giao dịch cũng khác nhau,… Vì vậy, mô hình khai phá tập phổ biến chỉ phản ánh mối tương quan giữa các phần tử xuất hiện trong cơ sở dữ liệu, nhưng không phản ánh ý nghĩa của từng phần tử dữ liệu. Để khắc phục những nhược điểm trên có hai mô hình được đưa ra: Tập phổ biến có trọng số - WFI và Tập lợi ích cao - HUI.

## Tập phổ biến

Khai phá tập phổ biến là quá trình tìm kiếm tập các phần tử có số lần xuất hiện cùng nhau lớn hơn một ngưỡng cho trước trong cơ sở dữ liệu lớn được R. Agrawal, T. Imielinski và A. Swami đề xuất năm 1993, xuất phát từ nhu cầu bài toán phân tích dữ liệu trong cơ sở dữ liệu giao dịch, để phát hiện các mối quan hệ giữa các tập hàng hóa đã bán tại siêu thị. Việc xác định này không phân biệt sự khác nhau giữa các hàng hóa mà chỉ dựa vào sự xuất hiện của chúng.

*Một số phương pháp khai phá tập phổ biến:*

- Phương pháp dựa trên quan hệ kết nối

- Phương pháp sử dụng cấu trúc cây

- Phương pháp tăng trưởng đệ quy dựa trên hậu tố

- Một số phương pháp song song

## Tập phổ biến có trọng số

Năm 1998, nhóm của Ramkumar đã đưa ra mô hình khai phá tập phổ biến có trọng số (*Weight Frequent Itemsets – WFI*). Trong đó, mỗi phần tử có một trọng số khác nhau như: lợi ích, giá cả, độ quan trọng hay số lượng,…Một tập các phần tử là phổ biến có trọng số khi giá trị có trọng số của chúng lớn hơn một ngưỡng cho trước. Dựa trên mô hình này đã có nhiều thuật toán khai phá tập phổ biến có trọng số được công bố.

*Một số phương pháp khai phá tập phổ biến có trọng số:*

- Thuật toán dựa trên khoảng trọng số

- Thuật toán sử dụng bảng băm

- Thuật toán dựa trên trọng số phổ biến xấp xỉ

- Thuật toán dựa trên cây WIT

## Đề xuất thuật toán khai phá mẫu phổ biến có trọng số theo chiều dọc

Dựa trên những ưu điểm của thuật toán VMDG khai phá tập phổ biến, đề xuất thuật toán khai phá tập phổ biến có trọng số với tên gọi VMWFP (Vertical Mining of Weighted Frequent Patterns Using Diffset Groups) sử dụng cấu trúc. Từ thuật toán VMWFP xây dựng thuật toán song song PVMWFP trên mô hình chia sẻ bộ nhớ. Kết quả thử nghiệm trên các cơ sở dữ liệu với 52 phần tử và 3984 giao dịch sinh ngẫu nhiên để tiến hành so sánh thuật toán song song PVMWFP với thuật toán tuần tự VMWFP được kết quả như *Hình 1.1*.

Hình .. Kết quả so sánh PVMWFP và VMWFP

## Tập lợi ích cao

Năm 2003 Chan và cộng sự đã đưa ra mô hình khai phá tập lợi ích cao (*High Utility Itemsets – HUI*), để khắc phục những hạn chế của mô hình khai phá tập phổ biến và tập phổ biến có trọng số. Trong mô hình này cho phép người sử dụng đánh giá được tầm quan trọng của từng phần tử qua hai trọng số khác nhau gọi là lợi ích trong và lợi ích ngoài.

Năm 2005, Ying Liu và cộng sự đưa ra khái niệm lợi ích giao dịch có trọng số của một tập phần tử X, ký hiệu là TWU(X) được tính bằng tổng lợi ích của các giao dịch có chứa tập phần tử X. Đây là giá trị có tính chất đóng, tính chất này đảm bảo rằng TWU(X) nhỏ hơn ngưỡng lợi ích tối thiểu thì tập X không có khả năng sinh ra tập lợi ích cao chứa tập X.

*Một trong những thách thức của khai phá tập lợi ích cao:*

- Tập lợi ích không có tính chất đóng, tính chất này đảm bảo một tập là tập lợi ích cao thì các tập con của nó cũng là tập lợi ích cao.

- Đa số các thuật toán khai phá tập lợi ích cao đều sử dụng ngưỡng TWU để cắt tỉa tập ứng viên. Đây là ngưỡng cao hơn rất nhiều so với giá trị lợi ích thực tế của một tập phần tử.

Do vậy, số lượng các ứng cử viên được sinh ra rất lớn dẫn đến không gian tìm kiếm và thời gian kiểm tra các ứng viên có chi phí cao.

Một số phương pháp khai phá tập lợi ích cao hiệu quả gần đây như: sử dụng danh sách lợi ích (utility-list) của Liu (2012); bảng chỉ số kết hợp bảng ứng viên của Guo (2013); ước tính lợi ích các cặp phần tử cùng xuất hiện của Philippe (2014); sử dụng dụng lợi ích cây con (utility sub-tree) và và lợi ích cục bộ (local utility) của Zida (2016).

# Thuật toán Khai phá tập lợi ích cao dựa trên mô hình CWU

## Mô hình hiệu quả khai phá tập lợi ích cao

#### Đặt vấn đề

Như chúng ta đã biết, đa số các thuật toán khai phá tập lợi ích cao được phân tích ở trên đều sử dụng mô hình TWU làm cơ sở để cắt tỉa các tập ứng viên. Với một phần tử a, một tập phần tử {X} và một tập phần tử có a là tiền tố {aX}, ta có TWU({aX}) là cận trên của AU({aX}). Tương tự, có TWU({X}) là cận trên của AU({X}). Ta thấy {X} ⊆ {aX} nên số giao dịch chứa {X} sẽ lớn hơn hoặc bằng số giao dịch chứa {aX}. Vậy, TWU({X}) là tổng lợi ích của các giao dịch chứa {X} sẽ lớn hơn hoặc bằng TWU({aX}) là tổng lợi ích của các giao dịch chứa {aX}.

Trong các thuật toán khai phá tập lợi ích cao theo chiều sâu. Giả sử, {aX} là tất cả các tập có tiền tố là phần tử a, {bX} là tất cả các tập có tiền tố là phần tử b. Khi khai phá các tập trong {bX} sẽ không còn chứa phần tử a. Nhưng khi tính TWU({bX}) có thể vẫn gồm giá trị lợi ích của phần tử a. Điều này làm TWU({bX}) là cận trên của AU({bX}) lớn hơn mức cần thiết và khi dùng TWU({bX}) để tỉa các tập ứng viên sẽ không hiệu quả.

Từ những phân tích ở trên, luận án đề xuất mô hình CWU (Candidate Weight Utility) và thuật toán HP khai phá tập lợi ích cao dựa trên mô hình này nhằm giảm số lượng tập ứng viên [II].

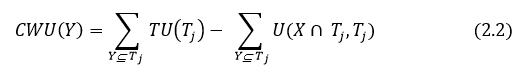
#### Đề xuất mô hình CWU

Từ những nhận xét trên, luận án đề xuất mô hình CWU để khắc phục nhược điểm của mô hình TWU.

**Định nghĩa 2.1.** [II] Tập tiền tố của một phần tử It là tập các phần tử trong tập I mà đứng trước phần tử It: SetPrefix(It) = {j I | j It}.

**Định nghĩa 2.2.** [II] Tiền tố của một tập phần tử có thứ tự Y là tập các phần tử trong I đứng trước phần tử đầu tiên y1 của tập Y, kí hiệu là SetPrefix(Y) và

SetPrefix(Y) = {j I | j y1} (2.1)

**Định nghĩa 2.3.** [II]Lợi ích ứng viên có trọng số (CWU – Candidate Weighted Utility) của tập phần tử Y, ký hiệu là CWU(Y) được xác định như sau:Đặt X = SetPrefix(Y), thì

Nếu X = ∅ thì .

**Định nghĩa 2.4.** [II]Khi CWU(Y) α với α là ngưỡng tối thiểu lợi ích ứng viên cho trước, ta gọi Y là tập lợi ích ứng viên có trọng số cao (HCWU- High Candidate Weighted Utility). Ngược lại, Y được gọi là tập lợi ích ứng viên có trọng số thấp (LCWU – Low Candidate Weighted Utility).

**Tính chất 2.1.** [II] Cho 3 tập phần tử có thứ tự I, Y­k-1,Yk thỏa mãn Y­k-1 ⊆ I, Y­k ⊆ Ivà Y­k-1 là tiền tố của Yk. Cụ thể: Yk-1 = {y1, y2,…, yk-1 | yi yi+1 với i=1..k-2} là tiền tố của tập Yk = {y1, y2,…, yk-1, yk | yi yi+1 với i=1..k-1} thì SetPrefix(Yk-1) = SetPrefix(Yk).

**Định lý 2.1.** [II] Xét 2 tập phần tử có thứ tự, Yk là tập k-phần tử, Yk-1 là tập (k-1)-phần tử và là tiền tố của Yk. Nếu Yk ∈ HCWUs thì Yk-1 ∈ HCWUs.

Đây là tính chất đóng của các tập phần tử theo mô hình CWU. Nghĩa là, nếu CWU(Yk-1) < thì CWU(Yk) <

**Định lý 2.2**. [II] Giả sử HCWUs gồm các tập Y có CWU(Y) α, HUs gồm các tập Y có AU(Y) α với α là ngưỡng lợi ích tối thiểu cho trước. Khi đó HUs ⊆ HCWUs.

Để khẳng định mô hình CWU có số ứng viên ít hơn mô hình TWU, luận án đưa ra hai bổ đề sau.

**Mệnh đề 2.1**. [II] Cho tập bất kỳ Y, ta luôn có CWU(Y) ≤ TWU(Y).

**Mệnh đề 2.2**. [II] Cho HCWUs gồm các tập Y có CWU(Y) α và HTWUs gồm các tập Y có TWU(Y) α, với α là các ngưỡng lợi ích tối thiểu cho trước, thì HCWUs ⊆ HTWUs.

## Thuật toán HP khai phá tập lợi ích cao dựa trên chỉ số hình chiếu và mô hình CWU

Trong phần này, luận án trình bày thuật toán HP được cải tiến từ thuật toán PB của Gou (2013) với một số cải tiến sau:

- Sử dụng kết hợp hai mô hình TWU và CWU;

- Sếp các phần tử trong từng giao dịch giảm dần theo AU sau khi đã loại các phần tử nhỏ hơn ngưỡng lợi ích tối thiểu.

#### Một số cấu trúc được sử dụng trong thuật toán:

- Bảng ứng viên TCk gồm: các tập k-phần tử, lợi ích ứng viên có trọng số - CWU và lợi ích thực tế của tập ứng viên - AU.

- Bảng chỉ số ITX của tập X gồm: các giao dịch Tj chứa tập X, vị trí p của phần tử cuối cùng của tập X xuất hiện trong giao dịch Tj và U(X,Tj). Từ bảng chỉ số ITX gồm k-phần tử có thể tính nhanh các tập ứng viên gồm (k+1)-phần tử với tiền tố là tập phần tử X.

- Bảng giao dịch lợi ích - UTi chứa giá trị lợi ích của phần tử i trong từng giao dịch gồm: giao dịch Tj chứa i và U(i, Tj). Sau khi tìm tất cả tập lợi ích cao với tiền tố là phần tử i thì dựa vào bảng UTi sẽ tính được CWU(Y) với phần tử i = ListItemPrefix(Y).

#### Kết quả thực nghiệm

Kết quả thử nghiệm, so sánh giữa thuật toán HP với các thuật toán Two Phase, PB trên bộ dữ liệu T30I4D100K và Mushroom.

|  |  |
| --- | --- |
| Hình .. Số lượng ứng viên được sinh ra trên T30I4D100K | Hình .. Thời gian thực hiện trên T30I4D100K |

|  |  |
| --- | --- |
| Hình 2.3. Số lượng ứng viên được sinh ra trên Mushroom | Hình 2.4. Thời gian thực hiện trên Mushroom |

## Thuật toán song song PPB khai phá tập lợi ích cao dựa trên chỉ số hình chiếu và danh sách lợi ích

Thuật toán song song PPB [V] khai phá tập lợi ích cao được công bố trong tạp chí Công nghệ Thông tin và Truyền thông: “Các công trình nghiên cứu, phát triển và ứng dụng CNTT-TT" với một số đóng góp sau:

- Dùng bảng chỉ số kết hợp với danh sách lợi ích để sinh tập ứng viên, tìm tập lợi ích cao, loại nhanh các ứng viên và độc lập xử lý các phần tử trên từng bộ xử lý.

- Giản lược thông tin lưu trữ trong danh sách lợi ích.

- Xây dựng thuật toán song song khai phá tập lợi ích cao trên mô hình chia sẻ bộ nhớ.

#### Một số cấu trúc được sử dụng trong thuật toán PPB gồm:

- Bảng TCk gồm: các tập k-phần tử, lợi ích thực tế - AU và lợi ích còn lại của ứng viên – RU. Các giá trị AU, RU trong bảng TC1 được tính trong cùng một lần duyệt để tính TWU, trong đó RU(X) = TWU(X) – AU(X).

- Bảng chỉ số ITX của tập X gồm: các giao dịch Tj chứa tập X; vị trí p của phần tử cuối cùng của tập X xuất hiện trong giao dịch Tj; itutil(X, Tj) – giá trị lợi ích của tập X trong giao dịch Tj; rutil(X, Tj) – giá trị lợi ích các phần tử còn lại sau tập X trong giao dịch Tj.

#### Kết quả thực nghiệm

Kết quả thử nghiệm, so sánh giữa thuật toán PPB-Miner với thuật toán HP [II] trên bộ dữ liệu T30I4D100K và Mushroom. *Hình 2.5* so sánh thời gian thực hiện khai phá tập lợi ích cao khi thay đổi ngưỡng lợi ích tối thiểu, *Hình 2.6* so sánh số lượng ứng viên được sinh ra tương ứng với các ngưỡng lợi ích tối thiểu khác nhau. *Hình 2.7* và *Hình 2.8* so sánh thời gian thực hiện khai phá tập lợi ích cao và số ứng viên sinh ra giữa hai thuật toán tương ứng với các ngưỡng lợi ích tối thiểu khác nhau trên bộ dữ liệu Mushroom.

|  |  |
| --- | --- |
| Hình .. Thời gian thực hiện trên T30I4D100K | Hình .. Số lượng ứng viên được sinh ra trên T30I4D100K |

|  |  |
| --- | --- |
| Hình .. Thời gian thực hiện trên Mushroom | Hình .. Số lượng ứng viên được sinh ra trên Mushroom |

## Thuật toán CTU-PRO+

Thuật toán CTU-PRO+ [III] cho khai phá tập lợi ích cao được cải tiến từ thuật toán CTU-PRO sử dụng mô hình CWU [II] được giới thiệu trong phần 2.2. Thuật toán CTU-PRO+ sử dụng cấu trúc cây mẫu lợi ích nén, các phần tử trong cây sắp xếp tăng dần theo lợi ích AU để các phần tử có lợi ích cao sẽ là tiền tố của các tập lợi ích và được khai phá trước. Sau đó, giá trị CWU sẽ được cập nhật lại bằng cách trừ đi lợi ích của các tiền tố đã được khai phá.

#### Một số cấu trúc

Các phần tử trong CSDL được đánh chỉ số 1, 2, 3,… theo thứ tự tăng dần theo AU.

* + - *Bảng phần tử chung* – GlobalItemTable gồm các phần tử ứng viên lợi ích có trọng số cao được sắp xếp tăng dần theo AU. Trong bảng này gồm: chỉ số (index), phần tử (item), lợi ích trên một đơn vị phần tử (utility), tổng số lượng của phần tử (quantity), lợi ích ứng viên có trọng số (CWU), lợi ích thực tế của phần tử (AU) và con trỏ trỏ đến gốc của nhánh trong cây mẫu lợi ích nén chung (GlobalCUP-Tree).
    - *Mỗi nút của GlobalCUP-Tree bao gồm:* chỉ số (index), mảng CWU tương ứng với giá trị lợi ích ứng viên có trọng số của 1 tập, mảng con trỏ chứa số lượng tương ứng của từng phần tử trong giao dịch, con trỏ trỏ đến nút anh em cùng mức, con trỏ trỏ đến nút cha.
    - *Mảng CWU[]* = {T0, T1,…, Tn}, trong đó: Ti là giá trị CWU của tập phần tử từ nút chỉ số i đến nút chứa Ti.
    - *Tập I =* {i1, i2,…, in} là tập hợp các phần tử HCWU trong giao dịch được ánh xạ tương ứng với các chỉ số trong GlobalItemTable sau đó chèn các chỉ số index vào cây mẫu lợi ích nén, bắt đầu từ nút gốc của nhánh cây được trỏ bởi con trỏ PST của phần tử i1 trong GlobalItemTable.

#### Kết quả thực nghiệm

Kết quả thử nghiệm, so sánh giữa thuật toán CTU-PRO+ với các thuật toán TwoPhase, CTU-PRO về so sánh thời gian thực hiện trên bộ dữ liệu T5N5D100K và T10N5D100K với ngưỡng lợi ích tối thiểu khác nhau.

|  |  |
| --- | --- |
| Hình .. Thời gian thực hiện trên T5N5D100K | Hình .. Thời gian thực hiện trên T10N5D100K |

# Thuật toán khai phá tập lợi ích cao trên cây danh sách lợi ích và điều kiện RTWU

## Cấu trúc dữ liệu hiệu quả cho khai phá tập lợi ích cao

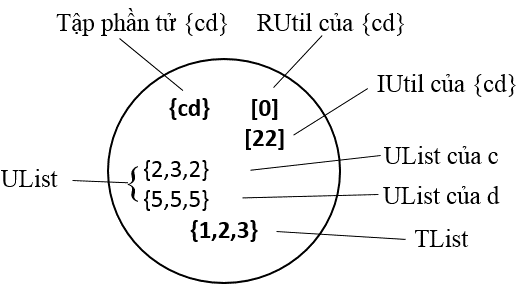
Trong thuật toán khai phá tập lợi ích cao sử dụng cấu trúc cây có những hạn chế như mỗi nút trên cây chỉ lưu trữ được một phần tử, dẫn đến khả năng nén không cao. Hơn nữa, các phần tử trong cây được sắp xếp giảm dần theo TWU nên số nút trong cây sẽ nhiều hơn sắp xếp giảm dần theo tần suất làm tốn không gian lưu trữ và tìm kiếm.

Năm 2012, Liu và cộng sự (2012) đã trình bày thuật toán khai phá tập lợi ích cao không sinh viên ứng viên. Trong thuật toán nhóm tác giả sử dụng cấu trúc danh sách lợi ích (utility list) để lưu trữ thông tin của tập phần tử và thông tin cắt tỉa không gian tìm kiếm.

Để khắc phục những hạn chế trong cấu trúc cây và tận dụng ưu điểm của danh sách lợi ích, trong phần này luận án trình bày một cấu trúc cây mẫu lợi ích nén (CUP) kết hợp danh sách lợi ích, trong đó mỗi nút chứa tập phần tử và danh sách lợi ích của nó. Cấu trúc này có thể cắt tỉa hiệu quả tập ứng viên làm giảm không gian tìm kiếm và lưu trữ. Trong cây các phần tử được sắp xếp giảm dần theo tần suất xuất hiện, làm giảm số nút xuất hiện trong cây so với việc sắp xếp theo TWU.

#### Mô tả cấu trúc cây CUP

Trong phần này, luận án sẽ trình bày khái niệm, cấu trúc cây CUP. Quá trình xây dựng cây CUP được mô tả chi tiết bằng thuật toán ở phần cuối.

Hình .. Ví dụ một nút trong cây CUP

Ví dụ như *Hình 3.1*, mô tả nút N trên cây CUP bao gồm: N.Itemset, N.IUtil, N.RUtil, N.TList, N.UList, N.Parent, N.Links và N.Childs. Trong đó, N.Itemsets là tập phần tử của nút, N.IUtil là giá trị lợi ích của N.Itemsets, N.RUTil là lợi ích còn lại của N.Itemsets, N.TList là danh sách các giao dịch chứa N.Itemsets, N.UList là một danh sách lợi ích của từng phần tử trong N.Itemsets tương ứng với N.TList, N.Parent là con trỏ trỏ đến cha của nút N, N.Links là danh sách con trỏ trỏ đến các nút có cùng các phần tử trong cây, N.Childs là danh sách con trỏ trỏ đến các nút con của nó.

#### Quá trình xây dựng cây CUP gồm các bước được mô tả như sau:

Để đơn giản luận án chỉ mô tả quá trình chèn các phần tử vào cây, còn các phần tính toán các giá trị RUtil, TList, UList sẽ được mô tả trong phần mô tả thuật toán.

*Bước 1,* duyệt dữ liệu lần 1 để đếm độ hỗ trợ (support) và tính TWU cho từng phần tử.

*Bước 2,* duyệt từng giao dịch, đưa các phần tử có TWU lớn hơn ngưỡng lợi ích tối thiểu vào danh sách. Sau đó sắp xếp các phần tử giảm dần theo tần suất.

*Bước 3,* xây dựng cây CUP.

Thực hiện chèn bằng cách lưu từng giao dịch vào danh sách phần tử và chèn danh sách phần tử này vào cây bắt đầu từ nút gốc như sau:

*Bước 3.1,* kiểm tra các nút con N của nút hiện tại và so sánh các phần tử trong N.Itemset với các phần tử trong danh sách chèn còn lại với các khả năng xảy như sau:

- Nếu tất cả các phần tử giống nhau thì chỉ thêm tid vào TList.

- Nếu không có 1 hoặc nhiều phần tử đầu tiên giống nhau thì tạo nút mới là con của nút hiện tại gồm: itemsets là các phần tử còn lại trong danh sách.

- Nếu có một hoặc nhiều phần tử đầu tiên giống nhau thì nút N chỉ gồm phần giống nhau, các phần tử khác nhau còn lại của nút N thành một nút con của nút N, các phần tử khác nhau của danh

#### Thuật toán khai phá tập lợi HUI-Growth

Sau khi xây dựng cây CUP thì các tập lợi ích cao được tìm ra bằng phương pháp đệ quy tương tự như thuật toán FP-Growth của Han (2000). Quá trình khai phá tập lợi ích cao trên cây CUP được duyệt từ dưới lên dựa vào bảng HeaderTable. Đầu tiên, lấy một phần tử ai cuối cùng trong bảng HeaderTable, dựa vào con trỏ liên kết của ai trỏ vào nút Ni để tìm các mẫu điều kiện với hậu tố ai. Chi tiết thuật toán được mô tả phía dưới.

#### Kết quả thực nghiệm

Trong phần này, luận án so sánh kết quả thực hiện thuật toán HUI-Growth [IV] với thuật toán: UP-Growth, HUI-Miner. Kết quả thử nghiệm, trong Hình 3.2 và Hình 3.3 so sánh thời gian thực hiện với các ngưỡng lợi ích khác nhau với hai bộ dữ liệu Mushroom và T40I4D100K.

|  |  |
| --- | --- |
| Hình 3.2. Thời gian thực hiện với dữ liệu Mushroom | Hình 3.3. Thời gian thực hiện với dữ liệu T40I4D100K |

## Điều kiện RTWU cho tỉa tập ứng viên

Thuật toán FHM do nhóm Fournier-Viger (2014) đã hạn chế các phép nối có chi phí cao của thuật toán HUI-Miner dựa trên tính chất đóng của TWU (Transaction-Weighted Utility). Đó là, không kết nối các tập sinh ra có chứa cặp (x, y) mà TWU(x, y) nhỏ hơn ngưỡng lợi ích tối thiểu cho trước. Tuy nhiên, như đã phân tích thì TWU là ngưỡng cao hơn mức cần thiết.

Trong thuật toán FHM để giảm số lượng phép nối bằng phương pháp cắt tỉa ước lượng giá trị lợi ích xuất hiện cùng nhau (EUCP - Estimated Utility Co-occurrence Pruning) dựa trên cấu trúc ước lượng giá trị lợi ích xuất hiện cùng nhau (EUCS - Estimated Utility Co-Occurrence Structure). Một cách cụ thể là thuật toán FHM sử dụng EUCS để lưu trữ TWU của tất cả các cặp phần tử (a, b). Dựa vào tính chất đóng của TWU, tất cả các tập chứa cặp phần tử (a, b) có TWU(ab) nhỏ hơn ngưỡng lợi ích tối thiểu sẽ không phải là tập lợi ích cao để ngừng việc ghép nối các danh sách lợi ích.

Tuy nhiên, thuật toán FHM khai phá các tập lợi ích cao theo chiều sâu. Giả sử, các phần tử được sắp xếp theo thứ tự từ điển, {aX} là tất cả các tập có tiền tố là phần tử a, {bX} là tất cả các tập có tiền tố là phần tử b. Như vậy, các tập chứa {bX} sẽ không còn chứa phần tử a. Nhưng khi tính TWU({bX}) có thể vẫn gồm giá trị lợi ích của phần tử a. Điều này làm TWU({bX}) là cận trên của U({bX}) lớn hơn mức cần thiết và khi dùng TWU({bX}) để tỉa các tập ứng viên sẽ không hiệu quả.

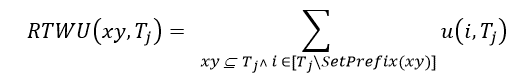
Để khắc phục những nhược điểm trên của thuật toán FHM, luận án đã đề xuất cấu trúc RTWU (Retail Transaction-Weighted Utility), xây dựng thuật toán tuần tự EAHUI-Miner sử dụng cấu trúc RTWU và thuật toán song song PEAHUI-Miner theo mô hình hạt mịn (fine-grain) từ thuật toán EAHUI-Miner.

**Định nghĩa 3.1.** [VI] Danh sách lợi ích mở rộng của một tập phần tử Px ký hiệu là exLstPx và được định nghĩa là một danh sách các phần tử, trong đó mỗi phần tử bao gồm bốn trường: *tid, iutil, itemutil và rutil*, trong đó:

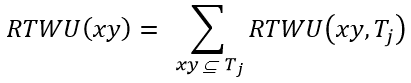
* *tid* là định danh của giao dịch chứa Px.
* *iutil* là lợi ích của tập phần tử P trong giao dịch tid chứa Px.
* *itemutil* là lợi ích của phần tử x trong giao dịch tid chứa Px.
* *rutil* là lợi ích còn lại của các phần tử còn lại trong giao dịch tid chứa Px, tính từ phần tử sau phần tử x.

Ngoài ra, danh sách lợi ích mở rộng của tập Px còn có các trường sau:

* *sumiutils* là tổng lợi ích của tập phần tử P trong các giao dịch tid chứa Px.
* *sumitemutils* là tổng lợi ích của phần tử x trong giao dịch tid chứa Px.
* *sumrutils* là tổng lợi ích còn lại của giao dịch có thứ tự *tid* chứa Px, bắt đầu tính từ phần tử kế tiếp sau phần tử x.

**Định nghĩa 3.2.** [VI]Giá trị lợi ích giao dịch còn lại của cặp phần tử xy trong giao dịch Tj chứa cặp phần tử xy là tổng lợi ích của các phần tử còn lại trong giao dịch có thứ tự Tj tính từ phần tử x. Kí hiệu là RTWU(xy, Tj), và

trong đó [Tj\ SetPrefix(xy)] – giao dịch Tj chứa cặp phần tử xy bỏ đi các phần tử đứng trước phần tử x.

**Định nghĩa 3.3.** [VI] Giá trị lợi ích giao dịch còn lại của cặp phần tử xy trong CSDL là tổng giá trị lợi ích giao dịch còn lại của cặp phần tử xy trong các giao dịch Tj chứa cặp phần tử xy trong CSDL. Kí hiệu là RTWU(xy) và

**Định nghĩa 3.4.** [VI] Cấu trúc RTWU được xác định bằng một tập các bộ ba: (x; y; c) I x I x R.

*Trong đó:*

* + I là tập các phần tử thuộc cơ sở dữ liệu;
  + x, y là 2 phần tử thuộc I (x đứng trước y theo một cách sắp xếp nào đó);
  + R là tập số thực và c = RTWU(xy).

**Định lý 3.1.** [VI] Cho hai tập Px, Py là mở rộng của tập P và hai danh sách lợi ích mở rộng của Px và Py lần lượt là exLstPx và exLstPy. Nếu min(exLstPx.sumiutls, exLstPy.sumiutls) + RTWU(xy) < minUtil thì Pxy và các các tập mở rộng của nó đều là các tập lợi ích thấp.

Dựa trên *Định lý 3.1*, luận án đề xuất cải tiến thuật toán FHM dựa trên cấu trúc RTWU, được trình bày ở phần tiếp.

## Thuật toán tuần tự EAHUI-Miner dựa trên điều kiện RTWU

Trong thuật toán EAHUI-Miner gồm 2 phần chính:

* + Xây dựng danh sách lợi ích mở rộng
  + Khai phá tập lợi ích cao EAHUI-Miner

Danh sách lợi ích mở rộng của tập chứa 1 phần tử được xây dựng theo *Định nghĩa 3.1* với tập P là rỗng (nghĩa là iutil=0) khi quét CSDL lần 1.

### Thuật toán song song PEAHUI-Miner

Thuật toán PEAHUI-Miner được xây dựng trên nền tảng OpenMP hỗ trợ lập trình song song trên môi trường bộ nhở chia sẻ. Thuật toán song song phân tải động theo mô hình hạt mịn (fine-grain) nhằm nâng cao khả năng cân bằng tải giữa các tiến trình.

### Kết quả thực nghiệm

* Số lượng ứng viên:Bảng 3.1 thể hiện số lượng tập ứng viên do hai thuật toán sinh ra. Kết quả cho thấy thuật toán FHM sinh ra nhiều tập ứng viện hơn so với thuật toán EAHUI-Miner.

Bảng 3.1. So sánh số lượng tập ứng viên.

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | minutil | FHM | EAHUI-Miner |
| 10I4D100K | 2500 | 153.016 | 125.647 |
| 10I4D100K | 2500 | 153.016 | 125.647 |
| Foodmart | 1000 | 259.876 | 258.921 |
| Mushroom | 100K | 1.588.018 | 1.587.927 |

* Thời gian thực hiện

Thời gian thực hiện của các thuật toán: EFIM, FHM và EAHUI-Miner được thể hình trên các *Hình 3.4, Hình 3.5, Hình 3.6 và Hình 3.7*. Kết quả này cho thấy, thuật toán EFIM thực hiện rất nhanh trên các cơ sở dữ liệu mà kích thước của tập phần tử I nhỏ, còn hai thuật toán FHM và EAHUI-Miner thực hiện nhanh hơn thuật toán EFIM trong các cơ sở dữ liệu mà kích thước tập phần tử I lớn.

|  |  |
| --- | --- |
| *Hình 3.4. Thời gian thực hiện trên Mushroom.* | *Hình 3.5. Thời gian thực hiện trên Foodmart* |

|  |  |
| --- | --- |
| *Hình 3.6. Thời gian thực hiện trên T10I4D100K* | *Hình 3.7. Thời gian thực hiện trên T10I4D200K* |

*Hình 3.8* và *Hình 3.9* so sánh thời gian thực hiện giữa thuật toán tuần tự EAHUI-Miner và thuật toán song song PEAHUI-Miner trên cơ sở dữ liệu T10I4D100K, T10I4D200K.

|  |  |
| --- | --- |
| Hình 3.8. Thời gian thực hiện trên T10I4D100K | Hình 3.9. Thời gian thực hiện trên T10I4D200K |

# Kết luận và kiến nghị

## *Kết quả chính của luận án:*

Với mục tiêu xây dựng mô hình, cấu trúc dữ liệu và thuật toán nhằm nâng cao hiệu quả thuật toán khai phá tập phổ biến có trọng số và tập lợi ích cao. Luận án đã đạt được các kết quả chính sau:

1. Mô hình lợi ích ứng viên có trọng số (CWU – Candidate Weighted Utility) [II] dựa trên phân tích cho thấy rằng mô hình TWU được nhiều thuật toán sử dụng để cắt tỉa ứng viên là không hiệu quả vì đánh giá ngưỡng cao hơn nhiều so với giá trị lợi ích thực tế. Từ mô hình CWU đề xuất hai thuật toán khai phá tập lợi ích cao là HP [II] sử dụng chỉ số hình chiếu, CTU-PRO+ [III] sử dụng cấu trúc cây cho số lượng ứng viên ít hơn và thời gian thực hiện nhanh hơn so với một số thuật toán.
2. Cấu trúc RTWU (Remaining Transaction-Weighted Utility) dựa trên giá trị lợi ích giao dịch còn lại kết hợp với danh sách lợi ích mở rộng của cặp phần tử cho cắt tỉa tập ứng viên. Phân tích thuật toán FHM [26] cho thấy để làm giảm chi phí kết nối (join) danh sách lợi ích dựa vào lưu trữ giá trị TWU của cặp phần tử. Tuy nhiên, mô hình TWU được đánh giá không hiệu quả cho cắt tỉa ứng viên. Do đó, luận án đề xuất cấu trúc RTWU làm giảm chi phí kết nối và tập ứng viên. Dựa trên cấu trúc RTWU, đề xuất thuật toán tuần tự EAHUI-Miner [VI] khai phá tập lợi ích cao và thuật toán song song PEAHUI-Miner [VI] khai phá tập lợi ích cao cho kết quả thực nghiệm có số lượng tập ứng viên ít hơn và thời gian thực hiện nhanh hơn khi cơ sở dữ liệu thưa và nhiều giao dịch.
3. Thuật toán song song PPB khai phá tập lợi ích cao kết hợp chỉ số hình chiếu, danh sách lợi ích và một phương pháp lưu trữ giá trị lợi ích của phần tử trên các giao dịch để tính nhanh giá trị *iutil* và *rutil* trong danh sách lợi ích.
4. Cấu trúc cây mẫu lợi ích nén (CUP) kết hợp với danh sách lợi ích [IV]. Mỗi nút trên cây CUP lưu trữ tập phần tử và danh sách lợi ích của nó. Các phần tử được sắp xếp giảm dần theo tần suất xuất hiện cho số nút trên cây là ít nhất. Để khai phá tập lợi ích cao trên cây CUP, luận án đề xuất thuật toán HUI-Growth [IV].
5. Thuật toán VMWFP [I] khai phá tập phổ biến lợi ích cao dựa trên cấu trúc diffset. Từ thuật toán VMWFP cho thấy rằng các nhóm, lớp các nhóm có thể xử lý độc lập nhau. Do đó, luận án đề xuất thuật toán song song PVMWFP [I] trên mô hình chia sẻ bộ nhớ.

## *Hướng phát triển*

Luận án tập trung vào bước quan trọng nhất trong khai phá luật kết hợp là khai phá tập phổ biến có trọng số và tập lợi ích cao. Cụ thể, đề xuất các mô hình, cấu trúc, thuật toán tuần tự và song song khai phá tập phổ biến có trọng số và tập lợi ích cao trên cơ sở dữ liệu giao dịch. Tuy nhiên, khối lượng dữ liệu ngày càng lớn và phức tạp, cần có những có những cấu trúc và thuật toán phù hợp. Do vậy, luận án sẽ tiếp tục các hướng nghiên cứu sau:

* Nghiên cứu các mô hình, cấu trúc và thuật toán hiệu quả khai tập phổ biến có trọng số và tập lợi ích cao.
* Đưa kỹ thuật khai phá dữ liệu mờ vào các thuật toán đã đề xuất.
* Cài đặt, thử nghiệm các thuật toán trên nền tảng lập trình Hadoop và mô hình Map-Reduce cho những bài toán dữ liệu lớn.