TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Ngô Chí Cường - 52100778**

**Lê Trần Phú – 52100572**

**USING GRAPH PATTERN MINING**

**FOR ANALYZING OF**

**MAKING MAXIMUM PROFIT**

**IN SELL CULTURE**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN**

**KỸ THUẬT PHẦN MỀM**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Ngô Chí Cường - 52100778**

**Lê Trần Phú – 52100572**

**USING GRAPH PATTERN MINING**

**FOR ANALYZING OF**

**MAKING MAXIMUM PROFIT**

**IN SELL CULTURE**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN**

**KỸ THUẬT PHẦN MỀM**

Người hướng dẫn

**ThS Dzoãn Xuân Thanh**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Trước tiên, chúng em xin gửi tới thầy Dzoãn Xuân Thanh lời cảm ơn chân thành nhất. Thầy là người đã mang đến cho chúng em nhiều kiến thức, kinh nghiệm và trong lúc đồng hành để hoàn thành Dự Án Công Nghệ Thông Tin. Suốt quá trình đồng hành cùng chúng em, Thầy luôn đưa ra những kiến thức chuyên môn sâu rộng, đầy sự tận tâm và nhiệt huyết.

Chúng em cũng xin bày tỏ lòng biết ơn sâu sắc đến tất cả các giáo viên trong Khoa Công nghệ thông tin của Trường Đại học Tôn Đức Thắng. Nhờ những kiến thức và kinh nghiệm mà các thầy cô đã chia sẻ, chúng em đã có thể hoàn thiện bài báo cáo này với chất lượng tốt nhất.

Cuối cùng, chúng em xin gửi lời cảm ơn đến gia đình và các bạn trong nhóm thực hành của chúng em. Những người bạn này đã luôn ở bên chúng em trong suốt quá trình học tập và giúp đỡ chúng em, là nguồn động lực rất lớn giúp chúng em tiến bộ hơn trong môn học của mình.

Chúng em rất cảm kích và tự hào khi được học tập và trải nghiệm những điều tuyệt vời như vậy. Chúng em hy vọng rằng sẽ có thể tiếp tục nhận được sự giúp đỡ và hướng dẫn từ các thầy cô để phát triển bản thân và đóng góp vào xã hội. Một lần nữa, chúng em xin chân thành cảm ơn!

*TP. Hồ Chí Minh, ngày 19 tháng 7 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của Ths Dzoãn Xuân Thanh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 19 tháng 07 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**USING GRAPH PATTERN MINING FOR ANALYZING OF**

**MAKING MAXIMUM PROFIT IN SELL CULTURE**

**TÓM TẮT**

Đề tài của chúng em sẽ trình bày về lý thuyết của việc Graph Mining dựa trên việc phân tích dữ liệu tĩnh không thay đổi theo thời gian và dữ liệu động được thay đổi theo thời gian và áp dụng các thuật toán xử lý dữ liệu thành các đồ thị con từ đó phân tích như TKG,AER… Vừa củng cố các cơ sở lý thuyết đã nêu và đồng thời thể hiện tính ứng dụng của Graph Mining vào cuộc sống

**USING GRAPH PATTERN MINING FOR ANALYZING OF**

**MAKING MAXIMUM PROFIT IN SELL CULTURE**

**ABSTRACT**

(Time New Romans – 13)

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vii](#_Toc172495998)

[DANH MỤC BẢNG BIỂU viii](#_Toc172495999)

[DANH MỤC CÁC CHỮ VIẾT TẮT ix](#_Toc172496000)

[CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI 1](#_Toc172496001)

[1.1 Lý do chọn đề tài 1](#_Toc172496002)

[1.2 Mục tiêu thực hiện đề tài 1](#_Toc172496003)

[1.3 Đối tượng nghiên cứu 1](#_Toc172496004)

[1.4 Ý nghĩa đề tài 2](#_Toc172496005)

[CHƯƠNG 2. KHAI THÁC ĐỒ THỊ CON 3](#_Toc172496006)

[2.1 Mức độ quan trọng việc phát triển việc khai thác dữ liệu dưới dạng đồ thị và các ứng dụng của chúng. 3](#_Toc172496007)

[2.2 Khai thác đồ thị con 3](#_Toc172496008)

[2.2.1 Khái niệm 3](#_Toc172496009)

[2.2.2 Bài toán khai thác đồ thị con 3](#_Toc172496010)

[2.3 Thuật toán khai thác cơ bản đồ thị con (G-Span) 3](#_Toc172496011)

[2.3.1 Khái niệm đi kèm trong khai thác đồ thị con 3](#_Toc172496012)

[2.3.2 Thuật toán G-Span 3](#_Toc172496013)

[2.4 Thuật toán TKG 3](#_Toc172496014)

[2.4.1 Cơ sở lý thuyết thuật toán 3](#_Toc172496015)

[2.4.2 Ví dụ đi kèm 3](#_Toc172496016)

[2.4.3 Các cách tối ưu thuật toán đề xuất 4](#_Toc172496017)

[CHƯƠNG 3. KHAI THÁC MẪU ĐỒ THỊ THUỘC TÍNH ĐỘNG 7](#_Toc172496018)

[3.1 Đồ thị thuộc tính động 7](#_Toc172496019)

[3.1.1 Khái niệm đồ thị động 7](#_Toc172496020)

[3.1.2 Khái niệm đồ thị thuộc tính động 7](#_Toc172496021)

[3.1.3 Các quy tắc giới hạn khi khai thác đồ thị thuộc tính động 7](#_Toc172496022)

[3.2 Các khái niệm về khai thác đồ thị thuộc tính động 7](#_Toc172496023)

[3.2.1 Trend-Graph 7](#_Toc172496024)

[3.2.2 Quy luật khi viết về sự biến đổi của các node trong Graph 7](#_Toc172496025)

[3.2.3 Công thức tính độ tự tin của một mẫu cốt lõi so với dữ liệu đồ thị gốc 7](#_Toc172496026)

[3.2.4 Công thức tính độ dốc của một mẫu cốt lõi so với dữ liệu đồ thị gốc 7](#_Toc172496027)

[3.3 Thuật toán AER (Attribute Evolution Rules) 8](#_Toc172496028)

[3.3.1 Các bước thực hiện thuật toán AER 8](#_Toc172496029)

[3.3.2 .Cách tối ưu thuật toán AER 10](#_Toc172496030)

[CHƯƠNG 4. MÔ HÌNH ĐỀ XUẤT 12](#_Toc172496031)

[CHƯƠNG 5. PHÂN TÍCH THIẾT KẾ THỰC NGHIỆM 13](#_Toc172496032)

[5.1 Dữ liệu thực nghiệm 13](#_Toc172496033)

[5.2 Cài đặt thực nghiệm 13](#_Toc172496034)

[CHƯƠNG 6. HIỆN THỰC THỰC NGHIỆM 14](#_Toc172496035)

[CHƯƠNG 7. KẾT QUẢ THỰC NGHIỆM 15](#_Toc172496036)

[CHƯƠNG 8. KẾT LUẬN 16](#_Toc172496037)

[8.1 Kết luận 16](#_Toc172496038)

[8.2 Hướng phát triển 16](#_Toc172496039)

[TÀI LIỆU THAM KHẢO 17](#_Toc172496040)

# DANH MỤC HÌNH VẼ

[Hình 2.1: Scaled Dot-Product Attention 3](#_Toc142677545)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1: Thống kê kiểu thực thể trong tập VLSP 2016 4](#_Toc142677565)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| BERT | Bidirectional Encoder Representations from Transformers |
| GEC | Grammatical Error Correction |
| MLM | Masked Language Model |
| NLP | Natural Language Processing |
| NSP | Next Sentence Prediction |

# MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

## Lý do chọn đề tài

Hiện nay, AI đang là xu hướng của thế giới khi tất cả các ngành nghề nếu muốn được hiệu suất cao thì một trong những thứ tiên quyết nhất chính là việc áp dụng AI vào các trường hợp thực tế. Đặc biệt nhất trong môi trường kinh tế khi bài toán gia tăng hiệu suất kinh tế luôn là thứ tiên quyết được đặt lên hàng đầu do đó việc tận dụng AI vào kinh tế là một việc hết sức cần thiết đặc biệt trong thời đại hiện nay.

Với những niềm cảm hứng trên thì việc chọn 1 đề tài vừa áp dụng vào thực tế hiện nay nhất là trong lĩnh vực kinh doanh và mang tính học thuật về việc xử lý dữ liệu đã là động lực để chúng em hoàn thành đề tài về … nhằm hỗ trợ việc tạo ra các hệ thống hỗ trợ ra quyết định ở các cấp quản lý và với dự định xa hơn là phát triển hệ thống lên thành cấp chiến lược.

## Mục tiêu thực hiện đề tài

Với mục đích trên thì đề tài trên sẽ nhắm chủ yếu vào việc hỗ trợ người dùng ra quyết định dựa trên các dữ liệu có sẵn và ở đây chủ yếu là từ các giao dịch sản phẩm từ các cửa hàng vừa và nhỏ để từ đó có thể hỗ trợ tạo ra các gói ưu đãi nhằm từ 1 sản phẩm cho trước để tối ưu hóa doanh số bán hàng.

## Đối tượng nghiên cứu

Thuật toán nghiên cứu: Chủ yếu là từ các thuật toán của Graph Mining dựa trên việc phân tích dữ liệu tĩnh không thay đổi theo thời gian và dữ liệu động được thay đổi theo thời gian và áp dụng các thuật toán xử lý dữ liệu thành các đồ thị con từ đó phân tích như TKG,AER

Dữ liệu nghiên cứu: Bộ dữ liệu hiển thị giao dịch của các cửa hàng bán lẻ vừa và nhỏ

Người sử dụng: Quản lý của các cửa hàng bán lẻ vừa và nhỏ

Công cụ nghiên cứu: Ngôn ngữ Python kèm với các framework hỗ trợ tạo ra Web dựa trên nền backend Python. Kèm với các template hỗ trợ render dưới dạng server như Jinja2 và các ngôn ngữ lập trình Web phổ biến như HTML, CSS và Javascript đi cùng các thư viện phổ biến như Jquery.

## Ý nghĩa đề tài

Việc triển khai thành công hệ thống sẽ mang ý nghĩa rất to lớn cho các chủ cửa hàng vừa và nhỏ do các ứng dụng hiện nay chưa phổ biến quá nhiều đặc biệt là các ứng dụng xử lý dữ liệu và học máy cho các chủ cửa hàng vừa và nhỏ do nhiều nguyên nhân đi kèm về vấn đề chi phí phát sinh và hơn nữa những đóng góp trên sẽ góp phần hỗ trợ để hình thành đóng góp lớn hơn và có nhiều sức ảnh hưởng hơn trong công cuộc chuyển đổi số ở Việt Nam hiện nay.

# KHAI THÁC ĐỒ THỊ CON

## Mức độ quan trọng việc phát triển việc khai thác dữ liệu dưới dạng đồ thị và các ứng dụng của chúng.

Biểu đồ càng ngày quan trọng trong cuộc sống hầu hết được xuất hiện trong các ngành liên quan đến khoa học như:

* Hóa học: Các công thức hợp chất hữu cơ ….
* Sinh học: Cấu trúc Protein, mạng sinh học
* Công nghệ thông tin: Mạng xã hội …
* Điện: Các mạch điện…

Nhờ đó các thuật toán tìm kiếm đồ thị đã được phát triển trong lĩnh vực thông tin hóa học, thị giác máy tính… từ đó đẩy mạnh quá trình phân tích dữ liệu có cấu trúc đồ thị trong thời gian gần đây.

Trong đó phân tích đồ thị con được xem là một trong những hoạt động được đầu tư nghiều nhất nhiều nhất vì sự hữu ích trong tập hợp đồ thị, phân biệt các nhóm đồ thị khác nhau, phân loại và gom cụm đồ thị, xây dựng chỉ số đồ thị và hỗ trợ tìm kiếm đồ thị tương tự trong cơ sở dữ liệu đồ thị…

Qua đó các loại thuật toán liên quan đến đồ thị đã được ra đời và đưa ra được những ứng dụng tuyệt vời như sau

* Y học: Phân tích dữ liệu cấu trúc HIV bằng việc so sánh giữa các đồ thị con thường xuyên với nhau
* Hóa học: Phân loại đặc trưng của cấu trúc các chất hóa học
* Kinh doanh: Phân tích các dữ liệu liên kết giữa các khách hàng và mặt hàng với nhau nhằm tối ưu hóa việc kinh doanh dựa trên việc phát triển CRM

## Khai thác đồ thị con

### Khái niệm

Một đồ thị sẽ được xem là đồ thị con của đồ thị còn lại nếu cả tập hợp tập đỉnh và tập cạnh của đồ thị đó là con của tập hợp các tập đỉnh và tập cạnh của các đồ thị còn lại

Ví dụ: Giả sử G là đồ thị với tập đỉnh VG ={1,2,3,4} và tập cạnh EG = {(1,2), (2,3), (3,4)} và H là đồ thị với tập đỉnh VG ={1,2,3} và tập cạnh EG = {(1,2), (2,3)}. Thì lúc này H sẽ là đồ thị con của đồ thị G.

### Bài toán khai thác đồ thị con

Đầu vào:

* Các Database là dạng đồ thị chứa các đối tượng đại diện là các node trong đồ thị và các cạnh nối các node nó
* Các giá trị ngưỡng (min-sup)

Đầu ra:

* Các đồ thị con mà đảm bảo được số lượng đồ thị con  Giá trị ngưỡng (min-sup)

#### Vấn đề đặt ra khi chọn ngưỡng trả về

Nếu ngưỡng quá cao 🡺 Quá ít kết quả trả về 🡺 Hạn chế tính ứng dụng

Ngưỡng quá thấp 🡺 Quá nhiều kết quả trả về 🡺 Hiệu suất tính toán giảm đáng kể và thời gian tính toán tăng đột biến

Điều kiện để chọn ngưỡng

* Tài nguyên lưu trữ và thời gian tính toán
* Độ phức tạp của các tập mẫu
* Tính chất của tập mẫu

Vấn đề này còn lớn hơn khi trong thực tế việc chọn ngưỡng cần phải đảm bảo tính ứng dụng ở một số khía cạnh như không thể trả về quá nhiều đồ thị con làm ảnh hưởng tới việc phân tích của người dùng và đồng thời bài toán hiệu suất cũng là thứ cần phải cân nhắc

#### Phương pháp giải quyết một phần

Chúng ta sẽ chuyển bài toán về dạng xếp hạng mô hình có độ phổ biến cao nhất theo top-k (top-k frequent subgraph mining) để giảm số lượng đồ thị con cần trả về với các đầu vào và đầu ra như sau:

* Đầu vào: số nguyên k, đại diện cho số lượng mô hình con cần được sinh ra cùng với giá trị ngưỡng (min-sup) tối thiểu để mô hình con đó được xem là phổ biến
* Đầu ra: Tập hợp gồm k mô hình con phổ biến nhất trong dữ liệu

Mặc dù vấn đề về số đồ thị con trả về trong thực tế đã được giải quyết nhưng những vấn đề cốt lõi của việc chọn giá trị ngưỡng vẫn chưa giải quyết khi chưa thể giải quyết hoàn toàn vấn đề hiệu suất cho từng mô hình nhằm tối ưu hóa về mặt thời gian lẫn độ chính xác và vẫn còn đang là một ẩn số rất khó để tìm được lời giải.

## Thuật toán khai thác cơ bản đồ thị con (G-Span)

### Khái niệm đi kèm trong khai thác đồ thị con

#### Định luật Aptori:

Nếu một đồ thị con đã không phổ biến thì tất cả các đồ thị chứa nó sẽ không phổ biến vì khi một đồ thị con được nhận ra không đủ phổ biến thì cho dù có như thế nào thì tất cả các đồ thị chứa đồ thị con đó cũng không được xem là phổ biến.

#### Hệ số hỗ trợ của đồ thị con:

Hệ số hỗ trợ của đồ thị con được định nghĩa là tần số xuất hiện của đồ thị con đó dưới đồ thị gốc.

#### Biểu diễn các cạnh dưới dạng DFS code:

A table with numbers and lines

Description automatically generated

Mỗi cạnh có thể được biểu diễn dưới dạng DFS code bằng công thức chung như sau (số thứ tự cạnh 1, Số thứ tự cạnh 2, tên cạnh 1, trọng số nối giữa cạnh 1 và 2, tên cạnh 2)

#### Cách nhận biết trùng lặp trong đồ thị con:

* Ta biết rằng mỗi đồ thị con tương tự như 1 đồ thị thì đều có thể biểu diễn dưới dạng DFS code(Search DFS cho đồ thị và ra kết quả)
* Khi chúng ta thêm 1 cạnh vào đồ thị con thì nếu DFS code của chúng không là duy nhất thì chúng ta nhận ra đồ thị con đó bị trùng lặp và đều đó dẫn đến không được thêm đồ thị con đó vào hàng đợi Qk để tránh tính chất duy nhất của đồ thị
* Kỹ thuật này được áp dụng vào G-Span để nhận ra trùng lặp

### Thuật toán G-Span

Cho 1 tập dữ liệu gồm các tập hợp các cạnh {A,B,C,D} và các trọng số {a,b,c,d} và các đồ thị cần xét.

Bước 1: Tiến hành nối các đỉnh với nhau bắt đầu bắt đầu từ đỉnh A-A với trọng số là a. Tìm hệ số hỗ trợ của đồ thị con trên đồ thị cần xét và thêm vào hàng đợi Qk.

Bước 2: Tiếp tục với A-B với trọng số là a và xét tương tự theo đỉnh rồi trọng số tới số lượng đỉnh của đồ thị con.

Kết hợp với cách nhận biết trùng lặp để tránh xét các đồ thị con đã xét xong và định luật Aptori nhằm giảm số lượng bước cần thực hiện và cho ra kết quả cuối cùng.

## Thuật toán TKG

### Cơ sở lý thuyết thuật toán

Bước 1: Khởi tạo với minsup (ngưỡng độ phổ biến tối thiểu) là 1

Bước 2: Bắt đầu dùng DFS để rã đồ thị thành các đồ thị con

Bước 3: Tạo một hàng đợi Qk chứa các đồ thị con và tần suất xuất hiện của chúng theo thứ tự từ lớn đến bé

Bước 4: Khi k đồ thị con được tìm thấy thì bắt đầu nâng min-sup của việc tìm kiếm lên bằng với hệ số hỗ trợ bé nhất của các phần tử trong Qk: khi đã tìm đc k đồ thị con thường gặp nhất, minsup sẽ được tăng lên bằng độ phổ biến của đồ thị con ít phổ biến nhất trong Qk, điều này giúp giảm không gian tìm kiếm và loại bỏ các đồ thị con có độ phổ biến thấp hơn.

Bước 5: Sau đó mỗi khi các phần tử được thêm vào thì nâng cao giá trị min-sup lên: với mỗi đồ thị con có mã DFS duy nhất được thêm vào, giá trị minsup sẽ được tăng lên tương ứng bằng với của đồ thị con ít phổ biến nhất trong Qk, điều này giúp tối ưu hóa quá trình tìm kiếm bằng việc loại bỏ các đồ thị con không cần thiết. Khi thuật toán kết thúc, top-k các đồ thị con phổ biến nhất đã được tìm thấy.

### Ví dụ đi kèm

Bài toán: Database có 3 đồ thị cơ bản: G1 (A – B – C), G2 (A – B, B – C, C – D), G3 (A – B – C – D). Áp dụng TKG như sau:

Bước 1: Khởi tạo minsup = 1: Ban đầu, minsup được đặt là 1. Thuật toán bắt đầu tìm kiếm các cạnh đơn trong các đồ thị.

Bước 2: Tìm kiếm từ các cạnh đơn: Các cạnh đơn được xác định là: (A-B), (B-C), và (C-D). Các cạnh này xuất hiện trong tất cả các đồ thị.

Bước 3: Tạo queue Qk: queue Qk sẽ chứa các đồ thị con phổ biến nhất. Ban đầu, các đồ thị cạnh con có các cạnh đơn trên được thêm vào queue này.

Bước 4: Điều chỉnh minsup: Khi đã tìm thấy đủ k mẫu (giả sử k=3), minsup được điều chỉnh dựa trên độ phổ biến của tiểu đồ thị ít phổ biến nhất trong Qk. Giả sử tiểu đồ thị (C-D) ít phổ biến nhất, minsup sẽ được tăng lên tương ứng.

Bước 5: Mở rộng các đồ thị con: Thuật toán tiếp tục mở rộng các đồ thị con bằng cách thêm các cạnh. Ví dụ, từ (A-B) có thể mở rộng thành (A-B-C).

Bước 6: Kiểm tra kết quả DFS: Mỗi khi một đồ thị con mới được thêm vào , kết quả DFS của nó được kiểm tra để đảm bảo không bị trùng lặp. Nếu trùng lặp, tiểu đồ thị đó bị bỏ qua.

Cụ thể: Khi mở rộng (A-B) thành (A-B-C)

* Đồ thị con (A-B-C) có giá trị support > minsup. Do đó, nó được coi là đồ thị con trong top-k phổ biến và có được thêm vào Qk.
* Sau đó, minsup có thể được tăng lên dựa trên số lượng xuất hiện của các đồ thị con khác trong Qk.

#### 🡺 Vấn đề đặt ra: Việc thêm vào tính giá trị support của đồ thị làm cho bài toán tối ưu không gian tìm kiếm là một vấn đề rất nan giải

### Các cách tối ưu thuật toán đề xuất

#### Kĩ thuật Dynamic-Search(Tìm kiếm động):

Tổng quan: Dynamic search là một trong những kỹ thuật được sử dụng để tối ưu hóa thuật toán TKG (Top-K frequent subgraph mining) nhằm cải thiện hiệu suất bằng cách giảm không gian tìm kiếm. Phương pháp này tập trung vào việc tìm kiếm các đồ thị con có tần suất cao trước để nhanh chóng nâng ngưỡng tần suất tối thiểu (minsup), từ đó loại bỏ các đồ thị không tiềm năng

Quy trình:

* Khởi tạo
  + Giá trị minsup = 1
  + Khởi tạo priority queue Qk: lưu trữ các subgraph top-k phổ biến tại bất kỳ thời điểm nào. Các subgraph nào mà có giá trị support thấp sẽ có ưu tiên cao
  + Khởi tạo priority queue Qc: lưu trữ các subgraph mở rộng tiếp theo. Các subgraph nào mà có giá trị support cao sẽ có độ ưu tiên cao. Ban đầu queue này sẽ mang giá trị trống
* Các bước chạy thuật toán tối ưu
  + Trong khi Qc không rỗng:
* Lấy subgraph có độ ưu tiên cao nhất từ Qc
* Tìm tất cả các cạnh mở rộng của subgraph này và tính sup của chúng
* Đối với mỗi mở rộng của subgraph g, tạo subgraph g’ mới bằng cách thêm cạnh t vào DFS code của subgraph g ban đầu
* Nếu sup của g’ >= minsup và g’ là có mã DFS không bị trùng lặp, chèn g’ vào Qk
* Nếu thích thước của Qk >= k, nâng ngưỡng minsup và loại bỏ đồ thị con có sup thấp nhất khỏi Qk.
* Chèn g’ vào Qc như một ứng viên cho việc mở rộng trong tương lai

Kết thúc: trả về Qk chứa các đồ thị con phổ biến nhất tìm được.

#### Kĩ thuật loại bỏ đồ thị con không phổ biến

Bước 1: Cho đồ thị con g đã được lấy ra từ hàng đợi Qc, tìm kiếm và tính toán tần suất xuất hiện các đồ thị con và các đồ thị mở rộng của nó xuất hiện bên trong các đồ thị được cho sẵn.

Bước 2: Tìm tần suất cao nhất giữa các đồ thị đó với nhau gọi là hsup và rn là số lượng đồ thị còn lại không chứa bất kỳ đồ thị con g và mở rộng của chúng

Bước 3: Tính hsup + rn nếu tổng chúng bé hơn minsup thì chúng được xem là có đồ thị mở rộng không phổ biến và sẽ dừng bước tính toán đồ thị mở rộng ở các database còn lại.

#### Kĩ thuật tiền xử lý thuật toán

Tổng quan: Phương pháp trên quét cơ sở dữ liệu ban đầu nhằm tính toán sup của tất cả các đồ thị cạnh đơn trước khi thực hiện dynamic search, điều này giúp cập nhật các biến Qk, minsup và Qc, giảm thiểu thời gian xử lý cho các đồ thị cạnh đơn và tăng hiệu suất tổng thể

Quy trình:

* Scan: quét toàn bộ cơ sở dữ liệu để tính toán sup của tất cả các đồ thị cạnh đơn. Việc này nhằm xác định số lượng các lần xuất hiện của mỗi đồ thị cạnh đơn trong database
* Cập nhật Queue và minsup:
* Qk: thêm các đồ thị cạnh đơn có sup >= minsup vào Qk
* minsup: thiết lập minsup dựa trên giá trị hỗ trợ của các đồ thị cạnh đơn phổ biến nhất được tìm thấy
* Qc: thêm các đồ thị cạnh đơn có tiềm năng mở rộng vào Qc để tiếp tục mở rộng trong các bước tìm kiếm sau

Thực hiện dynamic search: sau khi cập nhật Qk, minsup, Qc, thực hiện dynamic search để khám phá các subgraph phổ biến hơn. Quy trình này bao gồm các bước mở rộng và kiểm tra các subgraph như đã mô tả trong phương pháp dynamic search

# KHAI THÁC MẪU ĐỒ THỊ THUỘC TÍNH ĐỘNG

## Đồ thị thuộc tính động

### Khái niệm đồ thị động

Một đồ thị mà bị thay đổi bởi yếu tố thời gian thì được gọi là một dynamic-graph

A diagram of a diagram

Description automatically generated

### Khái niệm đồ thị thuộc tính động

Dynamic Attributed-Graph là một dynamic graph mà trong đó các node có thể được xem là 1 thực thể chứa các thuộc tính khác nhau

A diagram of a diagram

Description automatically generated

### Các quy tắc giới hạn khi khai thác đồ thị thuộc tính động

Giới hạn về góc nhìn trong mô tả cho việc khai thác đồ thị thuộc tính động được nghiên cứu như sau

Giới hạn về pattern:

* Đồ thị tập trung vào các node có sự ảnh hưởng nhất định và họ hàng của chúng (significant trend sequences)
* Đồ thị mà các đỉnh có sự phát triển theo một quy luật như nhau và có thể không ảnh hưởng với nhau
* Đồ thị mà tập hợp các đỉnh không bị thay đổi theo thời gian

Giới hạn về quy luật:

* Chúng ta nên tập trung vào sự tiến hóa của tổng quan đồ thị khi khai thác thay vì để ý tới sự tiến hóa của các đỉnh bên trong chúng

## Các khái niệm về khai thác đồ thị thuộc tính động

### Trend-Graph

Đây là một đồ thị thể hiện sự biến thiên của các giá trị của các đồ thị gốc t1, t2… trong các khoảng thời gian ti1,ti2,… bằng các giá trị {+, -, =, ++, -- …}

A diagram of a mathematical equation

Description automatically generated

### Quy luật khi viết về sự biến đổi của các node trong Graph

Cách viết: (Subgraph thời gian t, Subgraph thời gian t+1, Giá trị đầu tiên, Giá trị tiếp theo)

### Công thức tính độ tự tin của một mẫu cốt lõi so với dữ liệu đồ thị gốc

Mục tiêu: Nhằm xác định xác suất xảy ra kết quả của node nguyên nhân



Trong đó:

V là đồ thị gốc cho trước

E là toàn bộ mẫu đơn cốt lõi

Sup là hệ số hỗ trợ so với đồ thị gốc

### Công thức tính độ dốc của một mẫu cốt lõi so với dữ liệu đồ thị gốc

Độ dốc thể hiện mức độ phụ thuộc các đỉnh Consequent so với các đỉnh Antecedent



Trong đó:

Conf là hệ số tự tin của mẫu cốt lõi đó so với đồ thị gốc

ExConf(antecedent(V,E)) là hệ số tự tin mong đợi của các mẫu đó so với đồ thị gốc

## Thuật toán AER (Attribute Evolution Rules)

Thuật toán AER (Attribute Evolution Rules) được đề xuất để khám phá một loạt đồ thị mẫu mới trong các đồ thị động có thuộc tính. Ý tưởng chung của thuật toán trên là từ việc tìm ra quy luật thay đổi các giá trị của Sub-Graph nhằm đưa ra về quy luật thay đổi của đồ thị chính

### Các bước thực hiện thuật toán AER

Bước 1: Biến đổi các giá trị liên tục thành các giá trị rời rạc

* Rời rạc hóa các giá trị liên tục: Chuyển các giá trị liên tục thành các nhóm rời rạc.

Ví dụ: Độ tuổi có thể được chuyển thành các nhóm tuổi: 0-12 (trẻ em), 13-19 (thiếu niên), 20-35 (thanh niên), 36-60 (trung niên), trên 60 (người cao tuổi).

* Gán nhãn cho các giá trị rời rạc: Mỗi nhóm giá trị rời rạc được gán một nhãn cụ thể.

Ví dụ: 0-12 được gán nhãn "trẻ em", 13-19 được gán nhãn "thiếu niên",…

Bước 2: Tìm kiếm các Consequent Node

* Xác định các Consequent Node: Tìm các đỉnh (nodes) trong đồ thị đại diện cho kết quả cuối cùng mà ta muốn phân tích.

Ví dụ: Trong một mạng xã hội, có thể chọn các đỉnh đại diện cho các sự kiện quan trọng như thay đổi trạng thái công việc, hôn nhân,…

* Tạo mẫu đơn: Mỗi mẫu đơn bao gồm một Consequent Node và các Antecedent Nodes (đỉnh nguyên nhân) liên kết với nó.

Ví dụ: Nếu Consequent Node là "thay đổi trạng thái công việc", thì Antecedent Nodes có thể là "thất nghiệp", "có việc mới",…

Bước 3: Mở rộng đồ thị

* **Mở rộng mẫu:** Bắt đầu mở rộng đồ thị theo kích thước k-1 (trong đó k là tổng số đỉnh Antecedent và Consequent) bằng cách thêm các đỉnh Antecedent vào đồ thị vào đồ thị.

Bước 4: Tính lại các hệ số

* **Tính tần suất (support):** Đếm số lần mẫu xuất hiện trong tập dữ liệu.

Ví dụ: Mẫu {(a,+)} → {(b,−)} xuất hiện trong 10% của các đồ thị xu hướng.

* **Tính độ tin cậy (confidence):** Tính tỷ lệ giữa số lần xuất hiện của mẫu và số lần xuất hiện trong đồ thị xu hướng.

Ví dụ: Độ tin cậy của {(a,+)} → {(b,−)} là 70%, nghĩa là trong 70% của các trường hợp khi aaa tăng, bbb giảm.

* **Tính độ dốc (lift):** Đo lường mức độ mạnh mẽ của mối quan hệ giữa các đỉnh trong đồ thị xu hướng.

Ví dụ: Độ dốc của {(a,+)} → {(b,−)} cho thấy mức độ phụ thuộc của b vào a.

**Bước 5: Lọc các mẫu không phổ biến:**

Sử dụng các hệ số phổ biến để lọc các mẫu không đáng kể :Dựa trên định luật Aptori, các mẫu không thể phát triển được nữa sẽ bị loại bỏ.

Bước 6: Tiếp tục mở rộng

Tiếp tục mở rộng ở bước 3 với điều kiện tập hợp các mẫu có kích thước k-1 sẽ không có phần tử là rỗng hay đồng nghĩa với việc là các tập hợp còn lại phải đảm bảo việc có thể mở rộng ở toàn bộ các tập hợp nếu không thì sẽ chấm dứt thuật toán và trả về kết quả

### .Cách tối ưu thuật toán AER

* Định nghĩa chỉ số hỗ trợ và độ dốc phổ biến
* Khi mở rộng mẫu thì tuân theo BFS với độ ưu tiên theo bảng chữ cái nhằm tạo ra tính đồng nhất
* Giới hạn kích thước cho các mẫu
* Loại bỏ các phần tĩnh hoặc dao động không qua nhiều trong dữ liệu gốc

# MÔ HÌNH ĐỀ XUẤT

…

# PHÂN TÍCH THIẾT KẾ THỰC NGHIỆM

## Dữ liệu thực nghiệm

Dữ liệu trong VLSP-2016 đã được chia sẵn thành hai tập huấn luyện và kiểm thử. Thống kê chi tiết của các kiểu thực thể và các tập dữ liệu được mô tả trong Bảng 4.1**.**

Bảng 4.1: Thống kê kiểu thực thể trong tập VLSP 2016

|  |  |  |
| --- | --- | --- |
| **Label** | **Train** | **Test** |
| **LOC** | 1210 | 1377 |
| **ORG** | 7478 | 274 |
| **PER** | 6230 | 1294 |
| **MISC** | 250 | 47 |
| **Total** | 15168 | 2992 |

## Cài đặt thực nghiệm

…

# HIỆN THỰC THỰC NGHIỆM

# KẾT QUẢ THỰC NGHIỆM

# KẾT LUẬN

## Kết luận

## Hướng phát triển

# TÀI LIỆU THAM KHẢO

Tiếng Việt

…

Tiếng Anh

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762