BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

NGUYỄN THANH TÀI

KHAI THÁC MẪU PHỔ BIẾN CỰC ĐẠI TRONG ĐỒ THỊ ĐƠN BẰNG PHƯƠNG PHÁP SO SÁNH GẦN ĐÚNG

LUẬN VĂN THẠC SĨ

Chuyên ngành: Công nghệ thông tin

Mã số ngành: 60480201

TP. HỒ CHÍ MINH, tháng 01 năm 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

NGUYỄN THANH TÀI

KHAI THÁC MẪU PHỔ BIẾN CỰC ĐẠI TRONG ĐỒ THỊ ĐƠN BẰNG PHƯƠNG PHÁP SO SÁNH GẦN ĐÚNG

LUẬN VĂN THẠC SĨ

Chuyên ngành: Công nghệ thông tin

Mã số ngành: 60480201

CÁN BỘ HƯỚNG DẪN KHOA HỌC: PGS. TS. LÊ HOÀI BẮC

TP. HỒ CHÍ MINH, tháng 01 năm 2016

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

Cán bộ hướng dẫn khoa học:
PGS. TS. LÊ HOÀI BẮC

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP. HCM ngày 30 tháng 01 năm 2016

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:

TT	Họ và tên	Chức danh Hội đồng	Cơ Quan Công Tác
1	PGS.TSKH. Nguyễn Xuân Huy	Chủ tịch	Viện Hàn Lâm KHCN Việt Nam
2	TS. Vũ Thanh Hiền	Phản biện 1	ĐH Kinh Tế Tài Chính
3	TS. Cao Tùng Anh	Phản biện 2	ĐH Công Nghệ TP.HCM
4	PGS.TS. Vũ Đức Lung	Ủy viên	ĐH Công Nghệ Thông Tin TP.HCM
5	TS. Hồ Đắc Nghĩa	Ủy viên, Thư ký	ĐH Công Nghệ TP.HCM

Xác nhận của Chủ tịch Hội đồng đánh giá Luận sau khi Luận văn đã được sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM
PHÒNG QLKH – ĐTSĐH

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

TP. HCM, ngày … tháng … năm 2016

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: Nguyễn Thanh Tài Giới tính: Nam

Ngày, tháng, năm sinh: 01 – 03 – 1990 Nơi sinh: Đức Phú – Mộ Đức – Quảng Ngãi

Chuyên ngành: Công Nghệ Thông Tin MSHV: 1441860020

I- Tên đề tài:

KHAI THÁC MẪU PHỔ BIẾN CỰC ĐẠI TRONG ĐỒ THỊ ĐƠN BẰNG PHƯƠNG PHÁP SO SÁNH GẦN ĐÚNG.

II- Nhiệm vụ và nội dung:

Nghiên cứu và triển khai các thuật toán khai thác MPBCĐ trong đồ thị đơn bằng phương pháp so sánh gần đúng.

Tìm hiểu và nghiên cứu thuật toán MaxAFG, cải tiến từ thuật toán MaxAFG để cải thiện về kết quả tìm được, đồng thời tối ưu về thời gian thực hiện và bộ nhớ sử dụng.

Đề xuất thuật toán ImaxAFG dựa trên thuật toán MaxAFG nhằm nâng cao tính hiệu quả của thuật toán, giúp người sử dụng khai thác được tối đa số MPBCĐ trên đồ thị đơn sử dụng phương pháp so sánh gần đúng.

III- Ngày giao nhiệm vụ: 20/8/2015

IV- Ngày hoàn thành nhiệm vụ: 20/2/2016

V- Cán bộ hướng dẫn:

Phó Giáo Sư. Tiến Sĩ. Lê Hoài Bắc

CÁN BỘ HƯỚNG DẪN **KHOA QUẢN LÝ CHUYÊN NGÀNH**

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

LỜI CÁM ƠN

Trước hết, cho tôi được gửi lời cảm ơn đến sự hướng dẫn và giúp đỡ tận tình của Thầy PGS.TS. Lê Hoài Bắc trong suốt thời gian nghiên cứu và thực hiện Luận văn.

Tôi cũng xin cảm ơn quý Thầy Cô đã nhiệt tình giảng dạy, truyền đạt cho chúng tôi những kiến thức bổ ích qua các môn học trong chương trình cao học.

Tôi cũng xin gởi lời cảm ơn đến gia đình, bạn bè và những người thân đã luôn quan tâm và giúp đỡ tôi trong suốt thời gian học tập và nghiên cứu hoàn thành Luận văn này.

Luận văn không thể tránh khỏi những sai sót, tôi rất mong nhận được ý kiến đóng góp của quý Thầy Cô và mọi người cho Luận văn được hoàn thiện hơn.

Tôi xin chân thành cảm ơn.

TP. Hồ Chí Minh, năm 2016

TÓM TẮT

Khai thác dữ liệu đồ thị đang nhận được sự quan tâm rất lớn vào những năm gần đây bởi vì tính phổ biến của dữ liệu đồ thị đang phát triển rất mạnh và được sử dụng rộng rãi trong các nghành khoa học công nghệ.

Bởi vì mẫu đồ thị phổ biến cực đại có ý nghĩa rất quan trọng trong các vấn đề nghiên cứu khoa học nên hiện nay có rất nhiều thuật toán khai thác MPBCĐ. Tuy nhiên kết hợp việc sử dụng phương pháp so sánh gần đúng trong khai thác MPB đang còn rất hạn chế. Luận Văn này sẽ đề xuất thuật toán ImaxAFG cải tiến để khai thác MPBCĐ trong đồ thị đơn sử dụng phương pháp so sánh gần đúng.

Để trình bày khả năng và tính hiệu quả của ImaxAFG, Luận Văn sẽ sử dụng bộ dữ liệu chuẩn SIS (Là một dạng dữ liệu hình khung có cấu trúc). Kết quả thực nghiệm cho thấy ImaxAFG tốt hơn về hiệu quả, giúp người sử dụng tối ưu hóa hơn về việc xác định MPBCĐ cũng như phân lớp cho một dữ liệu hình thể.

ABSTRACT

Graph Data Mining is receiving very great attention in recent years because of the popularity of graph data is developing strongly and widely and using in the science and technology sectors.

Because maximal frequent patterns are very important in matters of scientific research, so nowaday there are many maximal frequent pattern-mining algorithms. However, combining the use of inexact matching comparative method in mining maximal frequent patterns model is still very limited. This thesis will propose an improved algorithm ImaxAFG to minning maximal frequent patterns in a single graph using inexact matching.

To demonstrate the ability and effectiveness of ImaxAFG algorithm, this thesis will use the SIS standard data (structural images skeletons database). The experimental results will show, the ImaxAFG will be better than maxAFG in efficiency, easier for the users to optimize more about identifying maximal frequent patterns and layering extremes for a data form.

DANH MỤC CÁC TỪ VIẾT TẮT

Ký hiệu	**Diễn giải**
MPB	Mẫu phổ biến (Frequent pattern)
MPBCĐ	Mẫu phổ biến cực đại (Maximal Frequent Pattern)
NTĐ	Ngưỡng tương đồng Δ (dissimilarity threshold)
NTS	Ngưỡng tần số σ (Frequency threshold)
KTDL	Khai thác dữ liệu (Data Mining)
CSDL	Cơ sở dữ liệu (Database)

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH

MỤC LỤC

Chương 1: TỔNG QUAN

1.1. Giới thiệu

1.1.1. Giới thiệu khái quát về sự phát triển của khai thác dữ liệu đồ thị

KTDL đồ thị đã và đang nhận được sự quan tâm rất lớn từ những năm gần đây, có lẽ bởi vì tính phổ biến của dữ liệu đồ thị đang phát triển rất nhanh. Dữ liệu đồ thị được sử dụng trong rất nhiều phạm trù khác nhau như: hóa học, giải quyết vấn đề, phân tích tài liệu, phân tích mạng xã hội và nhiều lĩnh vực khác ….

Mẫu đồ thị phổ biến là một đồ thị con được tìm thấy trong một tập các đồ thị hoặc một đồ thị, và xuất hiện nhiều hơn NTS mà người dùng định nghĩa. Những mẫu này sẽ chứng minh tính hữu dụng trong công việc khai thác đồ thị và quá trình khai thác, đã trở thành một bài toán quan trọng trong lĩnh vực KTDL.

Trong hầu hết các trường hợp, quá trình khai thác diễn ra trong một tập các đồ thị và mục tiêu của chúng là tìm tất cả các tập con phổ biến với số lần xuất hiện đáp ứng NTS. Một đồ thị con được coi là một sự biểu diễn của một mẫu đồ thị nhất định hay không được quyết định bằng cách giải quyết các bài toán về đồ thị đẳng cấu, tất cả các biểu diễn đó đồng nhất với một mẫu mà chúng đại diện. Một số thuật toán tìm MPB trong hoàn cảnh chuẩn này đã được phát triển và rất thành công như: GraphSig[1], Gaston[2], gSpan[3] and gRed[4]. Tuy nhiên có một số bài toán mới cần được mô phỏng thông qua đồ thị, một tình huống mới xảy ra trong việc khai thác mẫu đồ thị.

Chi tiết vấn đề nghiên cứu là khai thác tất cả các mẫu phổ biến cực đại (MPBCD) trong một đồ thị đơn, sử dụng phương pháp so sánh gần đúng. Khai thác MPB từ một đồ thị đơn thì đơn giản hơn so với trường hợp khai thác MPB từ một tập các đồ thị. Hơn thế nữa, tập trung vào việc tìm kiếm các mẫu cực đại và việc sử dụng

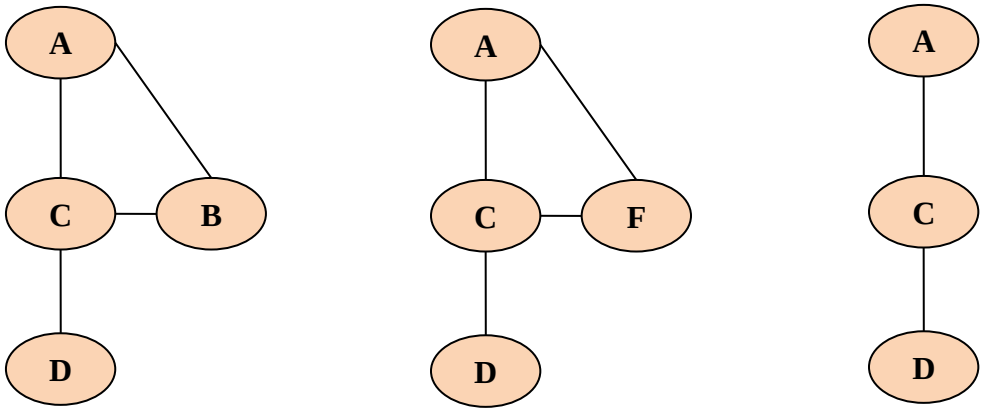
phương pháp so sánh gần đúng là hai thách thức cần được giải quyết, xuất hiện trong việc khai thác đồ thị hiện nay.

Đầu tiên phải kể đến sự bùng nổ số lượng mẫu đồ thị con tồn tại bên trong bài toán, thường thì kết quả của việc khai thác là một lượng lớn các mẫu, làm cho quá trình nghiên cứu và sử dụng chúng trở nên khó khăn. Bởi vì vậy nên trong những năm gần đây, một sự biến đổi đáng kể trong các thuật toán khai thác đã trở nên đáng chú ý. Sự thay đổi đó tập trung từ việc tìm kiếm tối đa mẫu đồ thị phổ biến đến việc tìm các tập con của chúng, sẽ dễ dàng hơn cho việc phân tích một tập nhỏ hơn.

Một phương pháp thông thường để lọc các mẫu dư thừa là tìm các mẫu đồ thị phổ biến cực đại, có nghĩa là các MPB không phải là đồ thị con của các MPB khác. Như được đề cập trong [6] và [7], tập các mẫu cực đại có thể là quan trọng hơn tập hoàn chỉnh của các mẫu. Hơn thế nữa, mẫu không cực đại có thể được xây dựng từ các mẫu cực đại. Như vậy thì, mặc dù thông tin xác định mẫu không cực đại sẽ không được lưu trữ, nhưng tất cả các đồ thị con phổ biến vẫn được tóm lượt trong những mẫu cực đại mà không thất lạc thông tin. Mặc dù một vài thuật toán đã được đề xuất để tìm các mẫu tiêu biểu [8], mẫu phân biệt [9], mẫu lớn nhất [10], trong đó mẫu cực đại đại diện một trong số những phương pháp hiệu quả để làm giảm bớt số lượng tập MPB.

Hướng nghiên cứu thứ hai sẽ hướng đến việc khảo sát một cách tỉ mỉ và nó có thể được phát hiện ra trong tình trạng hiện tại của khai thác đồ thị, dùng cho tính mềm dẻo trong các mẫu được tìm thấy trong dữ liệu. Thừa nhận phương pháp so sánh gần đúng có thể là một sự lựa chọn đúng trong một ngữ cảnh đặc biệt. Jia at al [11] đã đưa ra một trường hợp yêu cầu khai thác các mẫu hữu dụng từ một dữ liệu đồ thị tạp nhiễu. Trong trường hợp này, quan trọng là phải thừa nhận sự khác nhau giữa các nhãn của đỉnh hoặc cạnh, và chấp nhận một xác xuất nào đó cho việc gắn sai nhãn. Tương tự Chen at el [12] cũng quan tâm đến kho dữ liệu protein, trong đó MPB gần đúng là một điều đáng quan tâm trong sinh học. Không giống như những thuật toán trên, quan tâm việc sử dụng phương pháp so sánh gần đúng để thừa nhận sự khác cấu trúc trong các

đỉnh, theo cách đó thì hai đồ thị có một số đỉnh khác nhau có thể được xem xét là phù hợp. Một ví dụ ở hình H.01 bên dưới:



(a) G (b) P1 (c) P2

Hình 1: Sự biểu diễn của đồ thị sử dụng phương pháp so sánh gần đúng

P1 là một biểu diễn của G nếu sự khác nhãn được chấp nhận (Nhãn F thay thế nhãn B); đó là một phương pháp so sánh gần đúng được áp dụng cho thuật toán APGM[11].

P2 là một biểu diễn của G nếu sự khác cấu trúc được chấp nhận (những cạnh được gán nhãn với B thì bị khuyết); đó là một phương pháp so sánh gần đúng được áp dụng cho việc nghiên cứu thuật toán thuật toán(bên cạnh khác nhãn cũng được sử dụng).

Như vậy, chú ý đến việc giảm số lượng mẫu đã khai thác và ngược lại tạo thêm những mẫu không phù hợp hoàn toàn với các biểu diễn của đồ thị, nhưng quan trọng hơn trong một hoàn cảnh nhất định với một số thông tin hữu dụng có thể bị bỏ sót nếu sử dụng phương pháp so sánh chính xác.

Với những phương pháp và hạn chế nêu trên, luận văn sẽ tập trung tìm hiểu và đánh giá một thuật toán đã được đề xuất để khai thác tất cả các MPBCĐ trong đồ thị

đơn sử dụng phương pháp so sánh gần đúng. Từ đó đề xuất phương pháp cải tiến hiệu quả thuật toán, qua đó góp phần đáng kể trong việc khai thác các MPBCĐ.

1.1.2. Mục tiêu của đề tài

Tập trung tìm hiểu, đánh giá và đề xuất cải tiến hiệu quả thuật toán khai thác MPBCĐ trong đồ thị đơn sử dụng phương pháp so sánh gần đúng.

1.1.3. Nội dung nghiên cứu

Tìm hiểu các phương pháp khai thác đồ thị bao gồm khai thác đồ thị đơn và tập hợp các đồ thị.

Tìm hiểu các thuật toán về khai thác MPBCĐ trong đồ thị đơn và tập các đồ thị.

Tìm hiểu phương pháp so sánh gần đúng giữa hai đồ thị và các thuật toán liên quan.

Định hướng cải tiến và kiểm chứng thuật toán về khai thác MPBCĐ trong đồ thị đơn sử dụng phương pháp so sánh gần đúng.

1.2. Tổng quan về mẫu phổ biến cực đại

Vào năm 2004 Huan et al. [6] chú ý đến vấn đề khai thác các đồ thị con cực đại như một cách để làm hiệu quả hơn quá trình khai thác toàn bộ dữ liệu, làm giảm số lượng bộ nhớ cần và số lượng mẫu đã khai thác. Thuật toán SPIN khai thác cây phổ biến trong một tập các đồ thị, sau đó mở rộng cây phổ biến thành đồ thị tuần hoàn phổ biến, cuối cùng sẽ xây dựng nên đồ thị con phổ biến cực đại, sử dụng một vài kỹ thuật cắt tỉa để khai thác đồ thị con cực đại hiệu quả hơn.

Sau đó vào năm 2006 Thomas et al. [14] đề xuất thuật toán Margin để khai thác mẫu cực đại trong tập đồ thị. Cho mỗi một đồ thị trong tập dữ liệu nhập, thuật toán sử dụng một lưới đồ thị để miêu tả không gian tìm kiếm và định nghĩa các ứng viên là đồ thị phổ biến cực đại, chúng là các đồ thị con phổ biến mà không tồn tại bất kỳ đồ thị con phổ biến nào. Để tìm mẫu cực đại, đầu tiên tác giả tìm một đồ thị con liên thông

phổ biến và mở rộng nó cho đến cực đại, biểu diễn đồ thị cực đại bằng một điểm trên lưới. Sau đó họ nghiên cứu lưới để nhận dạng các ứng viên cực đại khác. Cuối cùng là bước hậu xử lý, tác giả sẽ kết hợp các ứng viên và chọn các MPBCĐ.

Vào năm 2012, Chen at el [15] đề xuất một phương pháp tìm mẫu cực đại trong tập các đồ thị bằng phương pháp khai thác từ trên xuống dưới. Đầu tiên, tác giả gán lại nhãn cho các đỉnh của đồ thị, sau đó những cạnh đối xứng với chúng thì định nghĩa bằng nhãn của chúng. Bước tiếp theo họ sẽ xây dựng ra một cấu trúc dạng cây (cây có cấu trúc) cho những đồ thị lớn trong tập hợp; mỗi cấp trong cây cấu trúc bao gồm những đồ thị con tìm được bằng việc xóa những cạnh không phổ biến từ mỗi đồ thị trong cấp trước. Dựa vào tính chất chống đơn điệu và sử dụng tính đối xứng trong nhãn của mỗi cạnh, thuật toán xóa bỏ các cạnh cho tới khi tìm thấy đồ thị phổ biến, là cực đại nếu các đồ thị cha của nó không phổ biến. Sau đó thuật toán sẽ tiếp tục bằng việc thêm những đồ thị còn lại trong tập hợp vào cây cấu trúc, sắp xếp giảm dần thứ tự theo kích thước, tìm cấp độ tương ứng cho mỗi đồ thị trong cây cấu trúc và sử dụng phương pháp đồng hình để so sánh các đồ thị con cùng một cấp độ trong cây cấu trúc.

1.3. Khai thác đồ thị trong đồ thị đơn

Vào năm 2004, Kuramochi và Karypis [19] [20] đưa ra một số thuật toán Hsigram, Vsigram và GREW để khai thác MPB trong một đồ thị đơn, ý tưởng của ba thuật toán này tương tự nhau, nhưng Hsigram sử dụng phương pháp tìm kiếm theo chiều rộng trước, ngược lại với Hsigram, Vsigram thì sử dụng phương pháp tìm kiếm theo chiều sau trước, còn với GREW là một sự cải tiến đáng kể của Vsigram: thuật toán cải thiện hiệu xuất của Vsigram bằng việc tập trung vào những đồ thị đăc trưng. Để tính toán độ hổ trợ của một mẫu đồ thị, thuật toán xây dựng đồ thị bao phủ cho tất cả các biểu diễn của mẫu đồ thị (một đồ thị với một đỉnh cho mỗi sự biểu diễn và một cạnh kết hợp cho mỗi cặp biểu diễn, như vậy là có sự bao phủ lên nhau) và định nghĩa độ hổ trợ của mẫu là số lượng phần tử trong một tập độc lập cực đại (MIS) của đồ thị

bao phủ. Nhưng việc tìm một tập MIS của đồ thị bao phủ là một vấn đề hết sức khó khăn, vì thế nên thuật toán sẽ phải đối phó với một thủ tục rờm rà mỗi lần nó tính toán độ hổ trợ của mẫu.

Vào năm 2008, Bringmann và Nijssen[17] đề xuất ra một cách tính độ hỗ trợ có ít sự tính toán hơn. Giả thuyết có một đồ thị G và một mẫu đồ thị P, thuật toán định nghĩa cách tính độ hỗ trợ như sau:

$\sigma(P,Q) = \min|\{\varphi(v)\ v \epsilon V : \varphi$ là một phép ánh xạ giữa P và một trong những biểu diễn của nó trong G}

Nghĩa là: Với mỗi đỉnh v của mẫu P sẽ có một số lượng đỉnh trong G mà v là ánh xạ của nó, độ hỗ trợ của P được xác định dựa trên số lượng đỉnh ánh xạ tối thiểu.

1.4. Khai thác đồ thị sử dụng phương pháp so sánh gần đúng

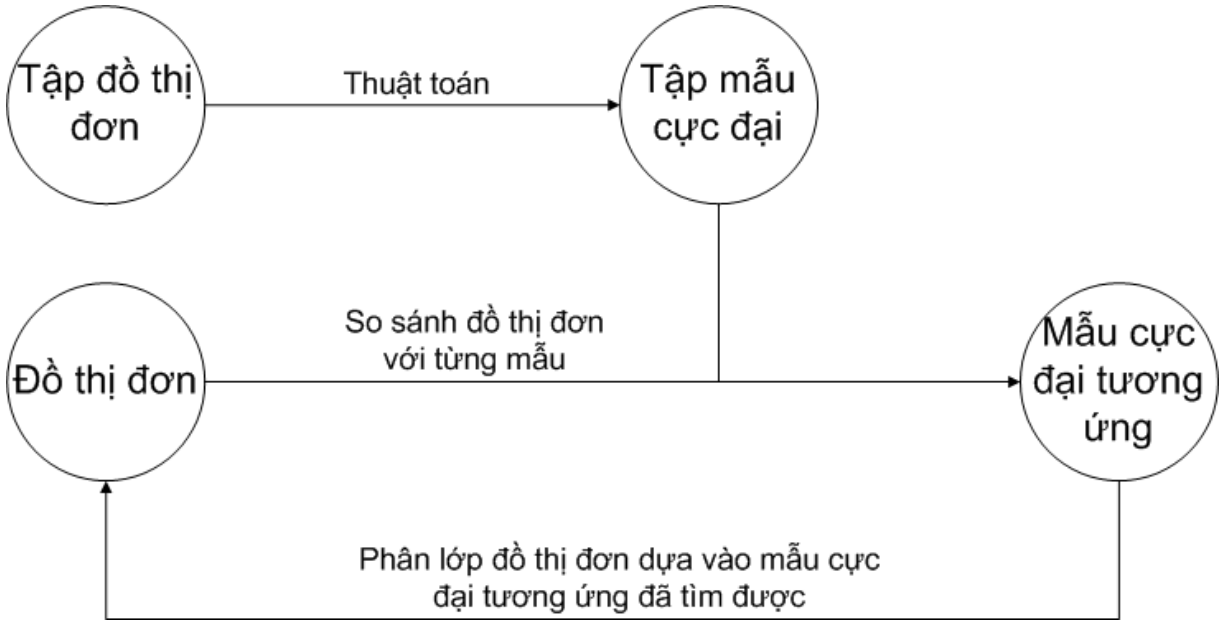
Có rất ít sự nghiên cứu về tìm kiếm MPB sử dụng phương pháp so sánh gần đúng, mặc dù không có sự nghiên cứu nào thừa nhận sự khác nhau về cấu trúc trong các đỉnh giữa MPB và các biểu diễn của nó. Vào năm 2011, Jia et al [11] đưa ra một thuật toán APGM để khai thác các MPB từ một tập các đồ thị trong một hoàn cảnh nhất định, với một lượng dữ liệu khổng lồ và đôi khi tồn tại những sai sót nhãn của các đỉnh và cạnh. Để giải quyết trường hợp này, tác giả đã đưa ra cách sử dụng ma trận thay thế, mỗi đầu vào ij của ma trận sẽ cho biết xác xuất nhãn i được gán nhãn sai bởi j. Sau đó thuật toán sẽ định nghĩa ra 2 đồ thị được gọi là đẳng cấu nếu như độ tương đồng của chúng thấp hơn ngưỡng cho phép. Mặc dù tác giả cho biết rằng thuật toán của họ có thể làm việc với sự thay đổi nhãn cho cả đỉnh lẫn cạnh nhưng chúng chỉ giải quyết trường hợp thay đổi đỉnh. Sau đó vào năm 2012, Acosta at al. đã đưa ra một thuật toán VEAM là một cải tiến của thuật toán APGM, để giải quyết cho cả hai trường hợp thay đổi đỉnh và cạnh. Cả hai thuật toán APGM và VEAM đều có yêu cầu những đồ thị kết hợp với nhau phải cùng cấu trúc liên kết.

Vào năm 2007, Chen at al [12] giới thiệu thuật toán gApprox tìm các MPB mà có thể không giống với các biểu diễn của nó về nhãn hoặc cấu trúc cạnh. Những mẫu được khai thác từ một đồ thị đơn sử dụng độ hỗ trợ được để xuất ở [16] và tính toán sự giống nhau của các đồ thị bằng việc phối hợp với khoảng cách điều chỉnh[13]. Vấn đề này tương đồng với hướng nghiên cứu nhưng tác giả không khai thác MPBCĐ.

1.5. Kiến trúc, hạ tầng của một hệ thống khai thác dữ liệu đồ thị

Dữ liệu đồ thị sử dụng để kiểm chứng thuật toán là một dạng dữ liệu hình ảnh bộ xương có cấu trúc, gồm 36 đồ thị biểu diễn bộ khung của những hình bóng thực tế. Dữ liệu này được chia thành 9 lớp: con voi, cái nĩa, quả tim, con ngựa, ngôi sao lớn, ngôi sao, con rùa và con cá voi; mỗi lớp có 4 đồ thị. Trong dữ liệu đồ thị này, mỗi đỉnh của đồ thị được gắn nhãn là một bộ phận của thân thể, trong khi đó nhãn của mỗi cạnh đồ thị là khoảng cách giữa 2 đỉnh mà chúng liên kết.

Từ dữ liệu đồ thị mẫu trên, một tập các MPBCĐ sẽ được khai thác bằng cách sử dụng thuật toán khai thác MPBCĐ trong đồ thị đơn sử dụng phương pháp so sánh gần đúng. Tìm ra MPBCĐ của một đồ thị mới từ tập MPBCĐ trên bằng cách so sánh đồ thị với từng mẫu trong tập mẫu đã tìm được và chọn ra mẫu tương thích nhất. Từ đó dễ dàng phân lớp được một đồ thị mới, lớp dự đoán của đồ thị mới là lớp của MPB tương thích nhất.



Hình 2: Tổng quan về hệ thống khai thác mẫu phổ biến cực đại

Chương 2: KHAI THÁC MẪU PHỔ BIẾN CỰC ĐẠI TRONG ĐỒ THỊ ĐƠN BẰNG PHƯƠNG PHÁP SO SÁNH GẦN ĐÚNG.

2.1. Tổng quan

Chương này sẽ giới thiệu một thuật toán dung để khai thác MPBCĐ trong đồ thị đơn sử dụng phương pháp so sánh gần đúng. Đầu tiên chương này sẽ giới thiệu một vài khái niệm cơ bản về đồ thị sẽ được sử dụng trong các phần sau. Sau đó sẽ miêu tả một hàm f(sim) gần đúng để so sánh các đồ thị có cấu trúc khác nhau. Tiếp theo sẽ mô tả chiến lược tìm kiếm mà thuật toán sử dụng để tìm kiếm những đồ thị biểu diễn khác nhau của mẫu, có thể khác nhãn hoặc khác cấu trúc. Từ đó sẽ phát họa thuật toán, và giải thích cách sử dụng hàm gần đúng và chiến lược tìm kiếm. Cuối cùng luận văn sẽ tìm hiểu kỹ thuật toán bằng một ví dụ nhỏ.

2.2. Khái niệm cơ bản và các ký hiệu

Trong phần này thuật toán sẽ sử dụng một vài khái niệm cơ bản về đồ thị. Từ lâu các nghiên cứu về đồ thị cũng đã làm quen với đồ thị có gán nhãn, đó là một đồ thị gồm 4 thành phần cơ bản G=(V,E, £, Ψ) trong đó:

V: là tập hợp các đỉnh của đồ thị

E: là tập hợp các cạnh của đồ thị $E \subset \{ (u,v) \mid u,v \in V , u \neq v \}$

£ : là một tập hữu hạn các nhãn gán cho đỉnh và cạnh của đồ thị

Ψ: là một chức năng để gán các nhãn trong £ cho cách đỉnh và cạnh của đồ thị

Ký hiệu V(G), E(G), và Ψ_G để tham chiếu đến một tập các đỉnh, một tập các cạnh và một hàm gán nhãn cho đồ thị G.

Một đồ thị H được gọi là đồ thị con của đồ thị G, được biểu diễn bằng $H \subset G$, nếu $V(H) \subset V(G)$, $E(H) \subset E(G)$, và $\forall v \in V(H)$ suy ra $\Psi_H(v) = \Psi_G(v)$.

Giả sử $V' \subset V(G)$ là một tập con các đỉnh của đồ thị G; đồ thị con của G được sinh ra từ V' nếu V(G') = V' và cho tất cả các đỉnh u, v ∈ V' thì có thể kết luận rằng (u,v) ∈ E(G') khi và chỉ khi (u,v) ∈ E(G).

Cuối cùng thuật toán sử dụng ký hiệu <> để biểu diễn sự kết nối vào nhau giữa một đồ thị con và một đỉnh mới. Ví dụ: Giả sử có một độ thị con của G là H và một đỉnh v ∈ V(G), biểu diễn H<>v là độ thị con của G được tạo ra bởi V(H) ∪ {v}.

2.3. Thuật toán ImaxAFG (cải tiến từ thuật toán MaxAFG)

2.3.1. Bài toán so sánh độ tương đồng

Thuật toán đòi hỏi một hàm so sánh đồ thị dùng để thừa nhận sự đồng dạnh giữa một số đồ thị khác cấu trúc, hay khác đỉnh hoặc cạnh.

Theo như Xiao [21], đơn vị đo lường khoảng cách giữa các đồ thị đã được đề xuất trong một số tài liệu có thể được phân loại dựa trên một số đặt tính như: giá thành, cấu trúc, chức năng. Đơn vị đo lường được phân loại gần đây dựa trên đồ thị tượng trưng thông qua các Vector, và không phù hợp với nghiên cứu này. Từ hai sự phân loại khác, sự đo lường dựa trên giá thành là một sự kết hợp tốt với thuật toán đang nghiên cứu. Bên cạnh đó, một vài sự đo lường dựa trên cấu trúc cũng tương đương với dựa trên giá thành.

Trong hướng nghiên cứu này, thuật toánsẽ đưa ra hàm khoảng cách chỉnh sửa của đồ thị như một chức năng tính toán sự tương đồ giữa các đồ thị: f_{sim}.

Giả sử đồ thị G_1 và G_2 là 2 đồ thị mà người dùng muốn so sánh; nếu không yêu cầu quan hệ tương đồng một – một giữa các đỉnh của đồ thị $V(G_1)$ và $V(G_2)$ thì sẽ có một tập hợp, R_{V1}, là các đỉnh của $V(G_1)$ mà không tương xứng với bất kỳ đỉnh nào của $V(G_2)$, và tương tự như vậy sẽ có một tập hợp, R_{V2}, là các đỉnh của $V(G_2)$ mà không tương xứng với bất kỳ đỉnh nào của $V(G_1)$. Giả sử rằng có một sự thiết lập quan hệ nhị phân một – một $m \in V(G_1) \times V(G_2)$ nghĩa là có một sự tương xứng giữa các tập con của $V(G_1)$ và các tập con của $V(G_2)$, định nghĩa độ giống nhau giữa hai tập đỉnh của hai đồ thị G_1 và G_2 như sau:

$$v_{edit} = \sum_{v \in V(G_1) \setminus R_{V1}} d_v\big(v, m(v)\big) + |R_{V1}| + |R_{V2}|$$

Trong đó d_v tượng trưng cho chi phí thay thế $\Psi_{G1}(v)$ bởi $\Psi_{G2}(m(v))$

Cùng chung một phương pháp phân tích như vậy, độ giống nhau giữa các cạnh của đồ thị được định nghĩa như sau:

$$e_{edit} = \sum_{(u,v)\in E(G_1)\setminus R_{E1}} d_e\big((u,v),(m(u),m(v))\big) + |R_{E1}| + |R_{E2}|$$

Trong đó R_{E1} và R_{E2} là tập hợp các cạnh không tương xứng giữa hai đồ thị.

Cuối cùng, độ tương đồng giữa hai đồ thị được tính toán như sau:

$$\mathrm{f}_{sim}(G_1, G_2) = kv_{edit} + (1-k)e_{edit}$$

Trong đó: k là một đơn vị đo trọng lượng giữa các cạnh và đỉnh mà hệ thống có thể yêu cầu.

Chú ý rằng hàm tương đồng này rất là quan trọng, bởi vì nó có thể được sử dụng để tính toán trong suốt quá trình KTDL đồ thị, làm tăng thêm các mẫu đồ thị mà không tăng thêm độ phức tạp của thuật toán.

2.3.2. Phương pháp so sánh gần đúng

Một trong những đặt thù chính của thuật toán là sẽ đề xuất ra khả năng tìm kiếm các mẫu mà không tương thích hoàn toàn với các biểu diễn của nó. Như vậy thuật toán sẽ sử dụng NTĐ Δ và hàm so sánh f_{sim} như đã diễn tả trong phần trước để xác định khi nào một đồ thị con là đủ giống với mẫu đồ thị đang phân tích.

Không gian tìm kiếm được thăm dò dựa trên một phương pháp tiếp cận mô hình tăng trưởng. Nếu có mẫu P và các sự biểu diễn của nó, thêm vào một đỉnh mới để tạo thành một mẫu P' mới, P' = P <> v và các sự biểu diễn của đồ thị P' sẽ được tìm ra bằng cách phân tích và phát triển các sự biểu diễn của P. Chiến lược này sẽ sử dụng để thừa nhận sự khác nhau của mỗi loại sẽ diễn tả sau đây.

a) Sự khác nhãn trong đồ thị.

Cung cấp một bảng dùng để định nghĩa rõ các mối tương đồng giữa các nhãn. Phương pháp này sẽ chấp nhận một số biểu diễn để tăng các mẫu thông qua một đỉnh mới mà nhãn của mẫu có thể sẽ không như chúng định nghĩa, sự thay thế nhãn được

chấp nhận thông qua thông tin tương đồng đó. Và có thể sẽ tìm ra các đồ thị biểu diễn mà nhãn của nó có thể sẽ không chính xác tuyệt đối.

b) Sự khác cấu trúc giữa các đỉnh trong đồ thị.

Chấp nhận sự khác nhau về cấu trúc ở đỉnh giữa các đồ thị, có nghĩa là một đồ thị biểu diễn của mẫu có thể có ít đỉnh hơn hoặc nhiều hơn đỉnh hơn so với mẫu của nó. Trong trường hợp đầu tiên, nếu một sự biểu diễn của P không thể phát triển để trở thành một sự biểu diễn của P', lưu chúng lại trong một tập hợp các sự biểu diễn của P' và đánh dấu chúng như một ứng viên cho đồ thị có đỉnh khuyết. Trong trường hợp thứ hai, những sự biểu diễn có thể có nhiều đỉnh hơn so với mẫu của chúng được tìm thấy bằng cách thay thế yêu cầu của một cạnh giữa các mẫu của P và một đỉnh đối xứng v

c) Sự khác cấu trúc giữa các cạnh trong đồ thị.

Cuối cùng, để tìm ra các đồ thị biểu diễn với các cấu trúc khác nhau về cạnh, chấp nhận thêm một đỉnh mới mà nó liên kết với sự biểu diễn của đồ thị P bằng một cách nào đó, nó giống như cái cách mà thuật toán gApprox làm.

Trong các trường hợp định nghĩa trước, dù có sự khác nhau về cấu trúc hoặc nhãn,nhưng phải luôn luôn ghi nhớ và theo dõi sự khác nhau đó giữa các mẫu và giữa các sự biểu diễn của nó. Trong trường hợp này, có thể dễ dàng sử dụng một hàm f_{sim} không tồn tại chi phí thêm vào để định nghĩa ra các sự biểu diễn phù hợp với NTĐ Δ và tính toán độ hỗ trợ của mẫu đồ thị.

2.3.3. Thuật toán ImaxAFG

Để tìm ra các MPBCĐ trong đồ thị, thuật toán đề xuất ImaxAFG là sự kết hợp giữa sách lượt tìm kiến đã giới thiệu ở các phần trước, hàm so sánh tính đồng nhất f_{sim} và sơ đồ tìm kiếm theo chiều sâu. Khi tìm ra một mẫu P, khai thác từ mẫu P một danh sách các sự biểu diễn của mẫu; sau đó khi phát triển mẫu P thành P', các sự biểu diễn của mẫu P' sẽ được khai thác bằng cách phân tích các biểu diễn của mẫu P. Mỗi lần

mở rộng một sự biểu diễn, các mẫu và biểu diễn sẽ được lưu trữ và theo dõi giá trị chi phí chỉnh sửa liên quan. Bằng cách mở rộng đó thuật toán sẽ dễ dàng phân tích độ tương đồng giữa P' và bất kỳ các sự biểu diễn thông qua hàm f_{sim}. Cuối cùng, để nhận biết được các mẫu cực đại, thuật toán chỉ cần lưu lại những mẫu đồ thị mà không thể mở rộng đến một mẫu mới mà phù hợp với đồ tương đồng.

Một sự thay thế nhãn tương đồng được định nghĩa thông qua một từ điển D và hỗ trợ việc tính toán cho mỗi mẫu thông qua một hàm được công bố bởi Bringmann và Nijssen [22]. ImaxAFG được phát họa sau đây, thuật toán dựa vào việc gọi nhiều lần các hàm như: Explore, Traverse, Expand, ExpandOccurrence ….

Main function:

Input:

G: Đồ thị được phân tích

σ: Ngưỡng tần số phổ biến

Δ: Ngưỡng tương đồng

D: Từ điển tương đồng giữa các nhãn

Output:

P: tập hợp các MPBCĐ của đồ thị G

$for\ v \in G\ do$

$explore(v) = False;$

$P = \emptyset;$

$for\ v \in G\ do$

$P_v = Expore(G, \sigma, \Delta, D, v);$

$P = mergeMaximals(P \cup P_v);$

Function Expore(G, σ, Δ, v, D):

Input:

G: Đồ thị được phân tích

σ: Ngưỡng tần số phổ biến

Δ: Ngưỡng tương đồng

v: Một đỉnh trong V_G dùng để phát triển MPB

D: Từ điển dùng để định nghĩa các nhãn tương đồng

Output:

P_v: tập hợp các MPB xuất phát từ đỉnh v

$for\ u \in G\ do$

$marked(u) = False;$

$P_v = \emptyset;$

M_v là một danh sách các đỉnh có nhãn giống hoặc tương đồng

(được tham khảo từ bộ từ điển nhãn đã cung cấp)với nhãn của đỉnh v;

C_v là một danh sách các edit cost giữa mỗi đỉnh trong M_v và mẫu đồ thị;

$if\ |M_v| \geq \sigma\ then$

$P_{Traverse} = Traverse(G, \{v\}, M_v, C_v, \sigma, \Delta, D);$

$if\ P_{Traverse} \neq \emptyset\ then$

$P_v = P_v \cup P_{Traverse};$

$else$

$P_v = P_v \cup \{v\};$

$expored(v) = true;$

<u>Function Traverse(G, P, M_P, C_P, σ, Δ, D):</u>

Input:

G: Đồ thị được phân tích

P: MPB ứng tuyển (sử dụng để duyệt tìm mẫu)

M_P: Danh sách các biểu diễn của P

C_P: Danh sách các edit cost giữa P và các biểu diễn của P

σ: Ngưỡng tần số phổ biến

Δ: Ngưỡng tương đồng

D: Từ điển dùng để định nghĩa các nhãn tương đồng

Output:

P_{exp}: tập hợp các MPB đạt được từ việc duyệt P (phát triển mẫu P)

V_{exp} *tập hợp các đỉnh liên kết đến P mà chưa khảo sát* $(explored)$*và đánh dấu*$(marked)$;

$if\ V_{exp}\ \neq\ \emptyset\ then$

$for\ u\ \in\ V_{exp}\ do$

$marked(u) = True;$

$V_{exp} = Expand(G, P, M_P, C_P, V_{exp}, \sigma, \Delta, D)$;

$for\ u\ \in\ V_{exp}\ do$

$marked(u) = False;$

<u>Function Expand(G, P, M_P, C_P, V_{exp}, σ, Δ, D):</u>

Input:

G: Đồ thị được phân tích

P: MPB ứng tuyển (sử dụng để duyệt tìm mẫu)

M_P: Danh sách các biểu diễn của P

C_P: Danh sách các edit cost giữa P và các biểu diễn của P

V_{exp}: Danh sách các đỉnh chưa khảo sát mà liên kết đến mẫu P

σ: Ngưỡng tần số phổ biến

Δ: Ngưỡng tương đồng

D: Từ điển dùng để định nghĩa các nhãn tương đồng

Output:

P_P: tập hợp các MPB đạt được từ việc mở rộng P

$P_P, P_E, P_T\ =\ \emptyset$;

$for\ v_{exp}\ \in\ V_{exp}\ do$

$P' = P <> v_{exp}$;

$M'_P, C'_P = ExpandOccurrence(G, P, M_P, C_P, v_{exp}, \Delta, D)$;

$sup_{P'} =$ độ hỗ trợ của P';

$if\ sup_{P'} \geq \sigma\ then$

$V'_{exp} = V_{exp} \backslash \{v_{exp}\}$;

$P_E = Expand(G, P', M'_P, C'_P, V'_{exp}, \sigma, \Delta, D)$;

$P_T = Traverse(G, P', M'_v, C'_v, \sigma, \Delta, D)$;

$if\ P_E \cup P_T = \emptyset\ then$

$P_P = P_P \cup P'$;

$else$

$P_P = P_P \cup P_E \cup P_T$;

Function ExpandOccurrence(G, P, M_P, C_P, newVertex, Δ, D):

Input:

G: Đồ thị được phân tích

P: MPB ứng tuyển (sử dụng để duyệt tìm mẫu)

M_P: Danh sách các biểu diễn của P

C_P: Danh sách các edit cost giữa P và các biểu diễn của P

newVertex: Đỉnh liên kết với mẫu P

Δ: Ngưỡng tương đồng

D: Từ điển dùng để định nghĩa các nhãn tương đồng

Output:

M'$_P$: Danh sách các đồ thị biểu diễn của $P' = P \cup \{newVertex\}$

C'$_P$: Edit cost liên quan đến mỗi đồ thị biểu diễn trong M'$_P$

$M'_P = \emptyset;\ C'_P = \emptyset;$

$for\ O\ (các\ đồ\ thị\ biểu\ diễn) \in M_P\ do$

//** Phân tích các đỉnh tương ứng với đỉnh $newVertex$
và liên kết với đồ thị biểu diễn O bằng một hay nhiều cạnh

v_{Neigh}: danh sách các đỉnh của đồ thị G ngoài trừ các đỉnh của đồ thị O
và liên kết với đồ thị O bằng một hay nhiều cạnh;
$for\ vertexV\ \in v_{Neigh}\ do$
$if\ label(vertexV) == or \approx\approx\ label(newVertex)\ then$
Thêm đồ thị $newOccurence(O\ \cup \{vertexV\})$ vào danh sách M'_P
Tính toán edit cost giữa $newOccurence$ và P'
Thêm đồ thị biểu diễn $newOccurence$ vào C'_P
$else$
Thêm O vào danh sách M'_P
Đánh dấu vắng mặt của đỉnh $newVertex$ như $'-'$
Tính toán edit cost giữa đồ thị biểu diễn O và P'
Thêm đồ thị biểu diễn O vào C'_P
//** Phân tích các đỉnh liên kết với O bằng đường đi ngẫu nhiên.

l: Đánh giá chiều dài đường đi cực đại có thể tồn tại giữa đồ thị biểu diễn O
và $newVertex$ mà không vượt qua ngưỡng tương đồng Δ
$for\ i = 2: l\ do$
$vertPath =$
Đỉnh thuộc đồ thị G ngoài trừ các đỉnh trong O và v_{Neigh}
và liên kết với O bằng một đường đi có chiều dài $= i$
$for\ vertexC\ \in vertPath\ do$
$if\ label(vertexC) == or \approx\approx label(newVertex)\ then$

$Thêm$ đồ $thị$ $newOccurence(O \cup \{vertexV\})$
$vào\ danh\ sách\ M'_P$
$Tính\ toán\ edit\ cost\ giữa\ newOccurence\ và\ P'$
$Thêm$ đồ $thị\ biểu\ diễn\ newOccurence\ vào\ C'_P$

Duyệt tất cả các đỉnh của đồ thị G, thuật toán cải tiến ImaxAFG chỉ thực hiện cho một đỉnh v duy nhất và khảo sát mở rộng MPB từ đỉnh v thông qua việc gọi hàm Explore. Sau khi đỉnh v khảo sát thành công, nó sẽ được đánh dấu là đã khảo sát và kết hợp các mẫu tìm được với các mẫu đã khảo sát trước đó (được phát triển từ khác đỉnh khác trước đó), chú ý chỉ giữ lại duy nhất các mẫu cực đại.

Việc khảo sát các mẫu cực đại được hoàn thành bởi việc phát triển đỉnh v thông qua việc gọi đệ quy đến hàm Traverse và Expand. Trong mỗi tình huống, hàm Expand gán nhãn unexplored (chưa khảo sát) cho đỉnh v_{exp} liên kết đến mẫu đã chỉ định. Và bằng việc gọi hàm Expand, thực hiện một công việc tìm kiếm theo chiều sâu để tìm các mẫu bắt nguồn từ P<> v_{exp}. Hàm Expand chỉ lưu trữ những MPB khi chúng không thể phát triển thêm được nữa.

Một bước quan trọng trong hàm Expand là khi một tập hợp các đồ thị biểu diễn của một mẩu mới P’ được hình thành thông qua việc gọi hàm ExpandOccurence. Trong hàm này mỗi một đồ thị biểu diễn của P được phân tích để nhận ra rằng đồ thị biểu diễn đó có thể mở rộng ra nữa dù khác nhãn, hoặc cạnh, hoặc đỉnh bị thiếu, hoặc các đỉnh dư thừa. Nếu một biểu diễn có thể được mở rộng, edit cost về đỉnh và cạnh của mẫu đó sẽ được tính toán và giữ lại cùng với đồ thị biểu diễn đó. Theo phương pháp này, có thể biết khi nào thì một đồ thị biểu diễn vượt qua được NTĐ đã cho trước.

2.3.4. Độ phức tạp của thuật toán ImaxAFG

Một vài chú ý nhỏ liên quan đến độ phức tạp của thuật toán. Rất khó khăn khi đưa ra các thảo luận liên quan đến tính hiệu quả của thuật toán, từ trước đến nay chưa

có tài liệu báo cáo về sự khác cấu trúc đỉnh của đồ thị, vì vậy khó có thể đưa ra một sự so sánh cân bằng.

Nếu chỉ xem xét một vài trường hợp khó cụ thể, một đồ thị với n đỉnh, liên kết hoàn toàn, những nhãn được định nghĩa, có thể tìm ra được độ phức tập của thuật toán là $O(n^2n!)$.

Giới hạn trên có được bởi sự tính toán như bên dưới:

Thuật toán ImaxAFG thực hiên thông qua tập đỉnh của đồ thị V(G), phân tích trong mỗi trường hợp thông qua hàm Explore, các MPB có thể tăng lên từ đỉnh của đồ thị. Với đỉnh đầu tiên v_1 hàm Explore sẽ gọi hàm Traverse, sau khi tìm ra tất cả các đỉnh liên kết với v_1 (Trong trường hợp này là $G \setminus \{v_1\}$) thì gọi lại hàm Expand. Sau cùng hàm Expand sẽ đệ quy lại chính nó trong một vòng lặp for, tăng lên đến n-1 cấp đệ quy (độ dài của tập $V_{exp} = G \setminus \{v_1\}$). Trong trường hợp này, những hàm gọi sau hàm Traverse là vô nghĩa vì tất cả các đỉnh trong đồ thị G đã được đánh dấu. Như vậy sẽ có "n-1" lần gọi đệ quy với hàm Expand và mỗi một lần thuật toán gọi một hàm ExpandOccurrence. Trong trường hợp phân tích này, đối với một mẫu P và một tập biểu diễn M_P, hàm ExpandOccurrence sẽ thực hiên $|M_P|.(n-|V(P)|)$ lần so sánh. Vì thế, xem xét đến những lần so sánh đó và số lần hàm Expand gọi lại chính nó, cuối cùng của việc khảo sát mẫu đồ thị từ đỉnh V_1 $(n-1)^2.(n-1)!$ lần so sánh. Cũng giống như vậy đối với đỉnh v_2 ngoài trừ lần đầu $V_{exp}=G \setminus \{v_1, v_2\}$ và kết quả là (n-2). (n-1).(n-1)! lần so sánh. Dựa vào sự lý luận trên, tổng số lần so sánh đối với thuật toán như sau:

$$T = (n-1)(n-1)(n-1)! + (n-2)(n-1)(n-1)! + \ldots + (n-1)(n-1)!$$
$$= (n-1)(n-1)![(n-1) + (n-2) + \ldots + 1]$$
$$= (n-1)(n-1)!\frac{(n-1)n}{2} = \frac{1}{2}(n-1)^2n!$$

Vậy kết luận: $T = \frac{1}{2}(n-1)^2n!$

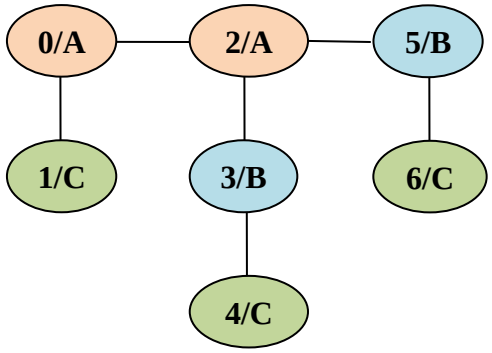
2.4. Bài toán tìm mẫu phổ biến cực đại trong đồ thị đơn sử dụng phương pháp so sánh gần đúng

Đồ thị dùng để phân tích thuật toán:

Ký hiệu "x/y": x là đỉnh, y là nhãn của đỉnh x.

Các đỉnh: 0, 1, 2, 3, 4, 5, 6

Các cạnh của đồ thị: [(0,1),(0,2),(2,3),(3,4),(2,5),(5,6)]



Hình 3: Đồ thị đơn có gắn nhãn

Duyệt từng đỉnh của đồ thị để tìm MPBCĐ với ngưỡng phổ biến δ= 3 và Δ = 2

I. Khảo sát từ đỉnh: [0] có nhãn là A

Tất cả các đỉnh có cùng nhãn với đỉnh [0]: Mp = {[2]}

Số lượng phần tử của tập Mp: len(Mp) = 1

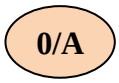
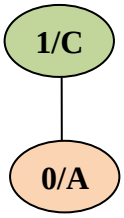
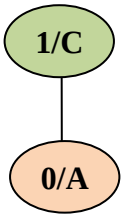
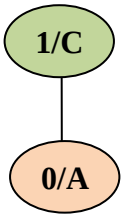
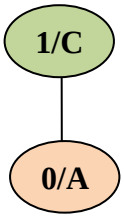
Vì len(Mp) < δ-1 => Dừng việc khảo sát.

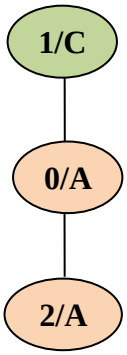
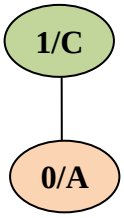
II. Khảo sát từ đỉnh: [1] có nhãn là C

Tất cả các đỉnh có cùng nhãn với đỉnh [1]: Mp = {[4], [6]}

Số lượng phần tử của Mp: len(Mp) = 2

Vì len(Mp) = δ-1 => Mở rộng đồ thị từ đỉnh [1] để tìm mẫu cực đại.

Bước	Mẫu cực đại	Mẫu mở rộng P	Các đỉnh nối với mẫu P	Biểu diễn của mẫu P. Các mẫu biểu diễn có độ khác biệt không quá $\Delta = 2$
Tạm lưu mẫu P là mẫu cực đại.				
Tìm các đỉnh nối với mẫu P mà chưa duyệt qua.				{[1], [4], [6]} Support = 3 > δ-1
Mở rộng mẫu P từ những đỉnh nối để tìm tất cả các mẫu có độ support > δ.				[[[1, '-'], [1, 1]], [[4, -1, 2], [1, 3]], [[4, '-'], [1, 1]], [[6, -1, 2], [1, 3]], [[6, '-'], [1, 1]]] Support = 3 = δ
Tạm lưu các mẫu mở rộng ở bước 2 là mẫu cực đại.				[[[1, '-'], [1, 1]], [[4, -1, 2], [1, 3]], [[4, '-'], [1, 1]], [[6, -1, 2], [1, 3]], [[6, '-'], [1, 1]]] Support = 3 = δ
Tìm các đỉnh nối với mẫu P mà chưa duyệt qua.				[[[1, '-'], [1, 1]], [[4, -1, 2], [1, 3]], [[4, '-'], [1, 1]], [[6, -1, 2], [1, 3]], [[6, '-'], [1, 1]]] Support = 3 = δ

Mở rộng mẫu P từ những đỉnh nối để tìm tất cả các mẫu có độ support >= δ.				[[[1, 0, '-'], [1, 1]], [[1, '-', 0], [1, 3]], [[1, '-', '-'], [2, 2]], [[4, '-', '-'], [2, 2]], [[6, '-', '-'], [2, 2]]] Support = 2 < δ => Dừng mở rộng mẫu
Vì độ Support < δ nên dừng việc khảo sát. Kết hợp các mẫu cực đại đã tìm được.				

Bảng 1: Mở rộng đỉnh 1/C tìm mẫu phổ biến

Gán nhãn đã duyệt cho đỉnh [1].

III. Khảo sát từ đỉnh: [2] có nhãn là A

Tất cả các đỉnh có cùng nhãn với đỉnh [0]: Mp = {[0]}

Số lượng phần tử của tập Mp: len(Mp) = 1

Vì len(Mp) < δ-1 => Dừng việc khảo sát.

IV. Khảo sát từ đỉnh: [3] có nhãn là B

Tất cả các đỉnh có cùng nhãn với đỉnh [3]: Mp = {[5]}

Số lượng phần tử của tập Mp: len(Mp) = 1

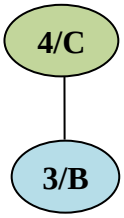
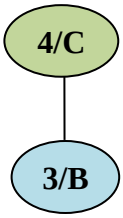
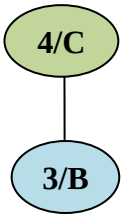
Vì len(Mp) < δ-1 => Dừng việc khảo sát.

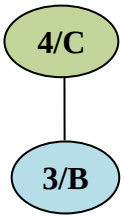
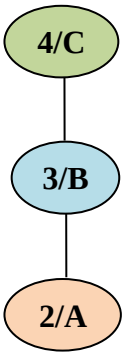
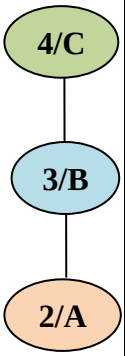
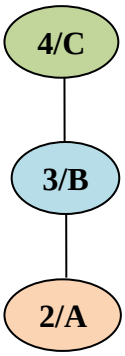
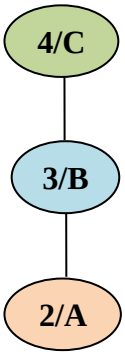
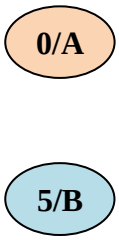
V. Khảo sát từ đỉnh: [4] có nhãn là C

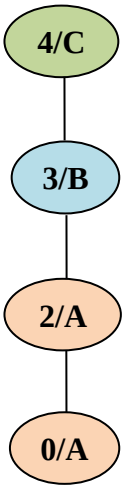
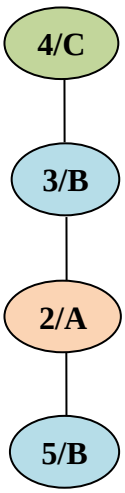
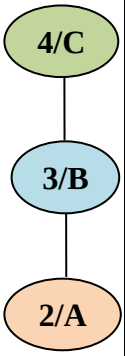
Tất cả các đỉnh có cùng nhãn với đỉnh [4]: Mp = {[1], [6]}

Số lượng phần tử của tập Mp: len(Mp) = 2

Vì len(Mp) = δ-1 => Mở rộng đồ thị từ đỉnh [4] để tìm mẫu cực đại.

Bước	Mẫu cực đại	Mẫu mở rộng P	Các đỉnh nối với mẫu P mà chưa được duyệt	Biểu diễn của mẫu P
Tạm lưu mẫu P là mẫu cực đại.				
Tìm các đỉnh nối với mẫu P mà chưa duyệt qua.				{[1], [4], [6]} Support = 3 = δ
Mở rộng mẫu P từ những đỉnh nối để tìm tất cả các mẫu có độ support > δ.				[[[4, '-'], [1, 1]], [[1, '-'], [1, 1]], [[6, 5], [0, 0]], [[6, '-'], [1, 1]]] Support = 3 = δ
Tạm lưu các mẫu mở rộng ở bước trên là mẫu cực đại.				

Tìm các đỉnh nối với mẫu P mà chưa duyệt qua.				
Mở rộng mẫu P từ những đỉnh nối để tìm tất cả các mẫu có độ support >= δ.				[[[4, 3, '-'], [1, 1]], [[4, '-', '-'], [2, 2]], [[1, '-', 0], [1, 3]], [[1, '-', '-'], [2, 2]], [[6, 5, 2], [0, 0]], [[6, 5, -1, 0], [1, 3]], [[6, 5, '-'], [1, 1]], [[6, '-', '-'], [2, 2]]] Support = 3 = δ
Tạm lưu các mẫu mở rộng ở bước trên là mẫu cực đại.				
Tìm các đỉnh nối với mẫu P mà chưa duyệt qua.				

Mở rộng mẫu P từ những đỉnh nối để tìm tất cả các mẫu có độ support >= δ.				[[[4, 3, 2, '-'], [1, 1]], [[4, 3, '-', 2], [1, 3]], [[4, 3, '-', '-'], [2, 2]], [[6, 5, 2, 0], [0, 0]], [[6, 5, 2, '-'], [1, 1]], [[6, 5, '-', 2], [1, 3]], [[6, 5, '-', '-'], [2, 2]]] Support = 2 < δ : Dừng việc mở rộng mẫu
				[[[4, 3, 2, '-'], [1, 1]], [[4, 3, '-', '-'], [2, 2]], [[6, 5, 2, 3], [0, 0]], [[6, 5, 2, '-'], [1, 1]], [[6, 5, '-', '-'], [2, 2]]] Support = 2 < δ : Dừng việc mở rộng mẫu
Vì độ support của tất cả các mẫu mở rộng đều nhỏ hơn δ nên dừng việc mở rộng. Hợp các mẫu cực đại tìm được.				

Bảng 2: Mở rộng đỉnh 4/C tìm mẫu phổ biến

Gán nhãn đã duyệt cho đỉnh [4].

VI. Khảo sát từ đỉnh: [5] có nhãn là B

Tất cả các đỉnh có cùng nhãn với đỉnh [5]: Mp = {[3]}

Số lượng phần tử của tập Mp: len(Mp) = 1

Vì len(Mp) < δ-1 => Dừng việc khảo sát.

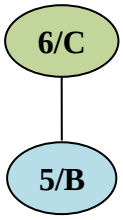
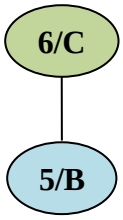
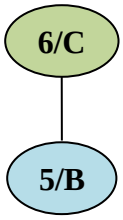
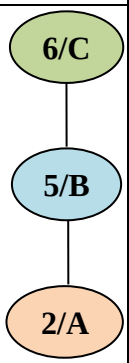
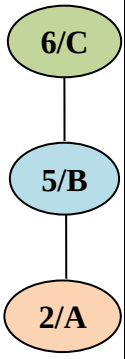
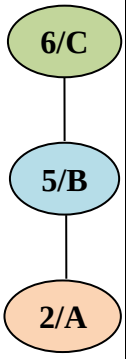
VII. Khảo sát từ đỉnh: [6] có nhãn là C

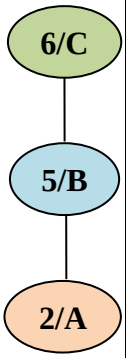
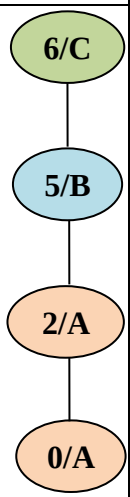
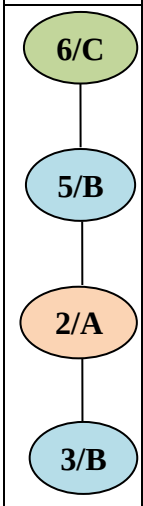
Tất cả các đỉnh có cùng nhãn với đỉnh [6]: Mp = {[1], [4]}

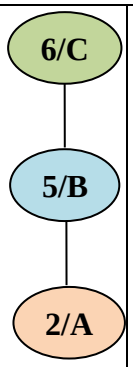
Số lượng phần tử của tập Mp: len(Mp) = 2

Vì len(Mp) = δ-1 => Mở rộng đồ thị từ đỉnh [6] để tìm mẫu cực đại.

Bước	Mẫu cực đại	Mẫu mở rộng P	Các đỉnh nối với mẫu P mà chưa được duyệt	Biểu diễn của mẫu P
Tạm lưu mẫu P là mẫu cực đại.	**6/C**	**6/C**		
Tìm các đỉnh nối với mẫu P mà chưa duyệt qua.		**6/C**	**5/B**	{[1], [4], [6]} Support = 3 = δ
Mở rộng mẫu P từ những đỉnh nối để tìm tất cả các mẫu có độ support > δ.		**6/C** **5/B**		[[[6, '-'], [1, 1]], [[1, '-'], [1, 1]], [[4, 3], [0, 0]], [[4, '-'], [1, 1]]] Support = 3 = δ

Tạm lưu các mẫu mở rộng ở bước trên là mẫu cực đại.				
Tìm các đỉnh nối với mẫu P mà chưa duyệt qua.				
Mở rộng mẫu P từ những đỉnh nối để tìm tất cả các mẫu có độ support >= δ.				[[[6, 5, '-'], [1, 1]], [[6, '-', '-'], [2, 2]], [[1, '-', 0], [1, 3]], [[1, '-', '-'], [2, 2]], [[4, 3, 2], [0, 0]], [[4, 3, -1, 0], [1, 3]], [[4, 3, '-'], [1, 1]], [[4, '-', '-'], [2, 2]]] Support = 3 = δ
Tạm lưu các mẫu mở rộng ở bước trên là mẫu cực đại.				

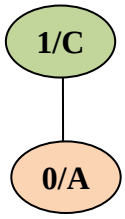
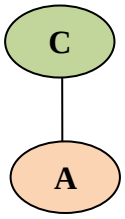
Tìm các đỉnh nối với mẫu P mà chưa duyệt qua.				
Mở rộng mẫu P từ những đỉnh nối để tìm tất cả các mẫu có độ support >= δ.				[[[6, 5, 2, '-'], [1, 1]], [[6, 5, '-', 2], [1, 3]], [[6, 5, '-', '-'], [2, 2]], [[4, 3, 2, 0], [0, 0]], [[4, 3, 2, '-'], [1, 1]], [[4, 3, '-', 2], [1, 3]], [[4, 3, '-', '-'], [2, 2]]] Support = 2 < δ : Dừng việc mở rộng mẫu
				[[[6, 5, 2, '-'], [1, 1]], [[6, 5, '-', '-'], [2, 2]], [[4, 3, 2, 5], [0, 0]], [[4, 3, 2, '-'], [1, 1]], [[4, 3, '-', '-'], [2, 2]]] Support = 2 < δ : Dừng việc mở rộng mẫu

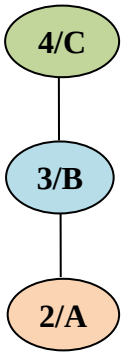
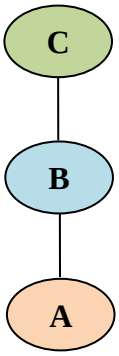
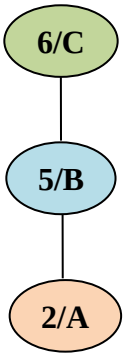
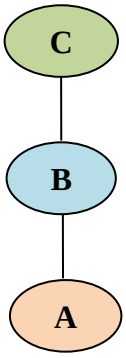
Vì độ support của tất cả các mẫu mở rộng đều nhỏ hơn δ nên dừng việc mở rộng. Hợp các mẫu cực đại tìm được.				

Bảng 3: Mở rộng đỉnh 6/C tìm mẫu phổ biến

Gán nhãn đã duyệt cho đỉnh [4].

Tóm tắt kết quả khảo sát 6 đỉnh của đồ thị:

Đỉnh	Đồ thị phổ biến cực đại	MPBCĐ	
[0]	Không tìm thấy		
[1]		 Mẫu #1	[[[1, '-'], [1, 1]], [[4, -1, 2], [1, 3]], [[4, '-'], [1, 1]], [[6, -1, 2], [1, 3]], [[6, '-'], [1, 1]]] Support = 3 = δ
[2]	Không tìm thấy		
[3]	Không tìm thấy		

[4]		 Mẫu số #2	[[[4, 3, '-'], [1, 1]], [[4, '-', '-'], [2, 2]], [[1, '-', 0], [1, 3]], [[1, '-', '-'], [2, 2]], [[6, 5, 2], [0, 0]], [[6, 5, -1, 0], [1, 3]], [[6, 5, '-'], [1, 1]], [[6, '-', '-'], [2, 2]]] Support = 3 = δ
[5]	Không tìm thấy		
[6]		 Mẫu số #3	[[[6, 5, '-'], [1, 1]], [[6, '-', '-'], [2, 2]], [[1, '-', 0], [1, 3]], [[1, '-', '-'], [2, 2]], [[4, 3, 2], [0, 0]], [[4, 3, -1, 0], [1, 3]], [[4, 3, '-'], [1, 1]], [[4, '-', '-'], [2, 2]]] Support = 3 = δ

Bảng 4: Tóm tắt quá trình mở rộng tìm mẫu phổ biến

Cuối cùng duyệt qua tất cả các MPBCĐ, chỉ chọn những mẫu tối ưu và lưu vào tập P:

P = []

Mẫu #1: Chọn mẫu #1 vào tập P vì mẫu này không phải là đồ thị con của bất kỳ mẫu nào trong P

P= [Mẫu #1]

Mẫu #2: Chọn mẫu #2 vào tập P vì mẫu này không phải là đồ thị con của bất kỳ mẫu nào trong P

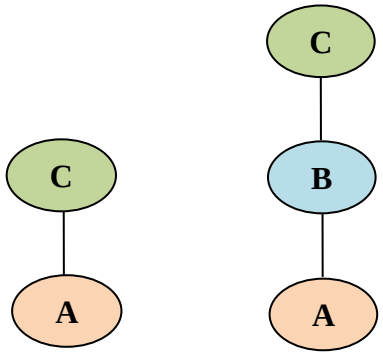
P= [Mẫu #1, Mẫu #2]

Mẫu #3: Không chọn mẫu #3 vào tập P vì mẫu này là đồ thị con của mẫu #2 trong P

P= [Mẫu #1, Mẫu #2]

Kết luận:

Có 2 MPBCĐ khi khảo sát đồ thị trên là:



Hình 4: Mẫu phổ biến cực đại của đồ thị ví dụ

Chương 3: KẾT QUẢ THỰC NGHIỆM VÀ HƯỚNG PHÁT TRIỂN.

3.1. Giới thiệu

Dữ liệu đồ thị đầu tiên sử dụng để kiểm tra tính đúng của thuật toán là SIS (Là một dạng dữ liệu hình khung có cấu trúc). Dữ liệu SIS bao gồm 36 đồ thị, mỗi đồ thị là một hình khung tượng trưng cho một cái bóng của một hình ảnh thực. Bộ khung của mỗi hình được tính toán ở hai trạng thái: trạng thái đầu tiên là trạng thái bán tự động, trạng thái thứ hai là trạng thái điều chỉnh bằng tay. CSDL được chia thành 9 lớp: con voi, cái nĩa, trái tim, con ngựa, con người, ngôi sao lớn, ngôi sao trung bình, con rùa và cá voi; mỗi lớp có 4 đồ thị. Trong CSDL đồ thị này, các đỉnh được gắn nhãn là bộ phận của vật thể, trong khi đó cạnh được gắn nhãn là khoảng cách giữa các đỉnh mà chúng liên kết với nhau. Tập hợp này bao gồm 13 nhãn khác nhau được gán cho đỉnh và 211 nhãn gán cho cạnh.

3.2. Kết quả thực nghiệm thuật toán ImaxAFG dựa vào kỹ thuật kiểm tra ”k-fold cross validation”

Luận văn này sử dụng phương pháp “4-fold cross validation” (4-fold cross) để tạo ra tập học và tập kiểm tra bởi vì dữ liệu đang sử dụng được chia ra gồm 4 lớp khác nhau.

Phương pháp “4-fold cross validation” là một kỹ thuật thông thường để đánh giá hiệu quả của việc phân lớp.

Trong mỗi trường hợp tìm kiếm MPB của tập học, sử dụng NTS $\sigma = 2$ và NTĐ $\Delta = 6$; giá trị của ngưỡng tầng số σ là một NTS nhỏ nhất mà người dùng có thể sử dụng để tìm những mẫu khác nhau từ đồ thị đưa vào, trong khi đó NTĐ Δ thừa nhận sự khác nhau về cấu trúc đỉnh lên đến 3 đỉnh, một nữa số lượng đỉnh trung bình trong một đồ thị của tập dữ liệu trên.

1. **Sắp xếp thứ tự ngẫu nhiên cho tập dữ liệu học:**

Ký hiệu chi tiết của đồ thị:

"**t # n c**" – t: viết tắt của "tree" là ký hiệu bắt đầu 1 đồ thị, n: là số thứ tự bắt đầu từ 0, c: là lớp của đồ thị.

"v i j" – v: viết tắt của "vertex" là đỉnh của đồ thị, i: tên của đỉnh, j: là nhãn của đỉnh.

"e x y d" – e: viết tắt của "edge" là cạnh của đồ thị, x,y: là hai đỉnh của cạnh, d: là nhãn của cạnh và cũng là khoảng cách giữa hai đỉnh.

Lớp	**Đồ Thị**	**Chi tiết**		**Lớp**	**Đồ Thị**	**Chi tiết**		**Lớp**	**Đồ Thị**	**Chi tiết**
#0	1	t # 0 0		#3	13	t # 12 3		#6	25	t # 24 6
		v 0 0				v 0 1		`		v 0 7
		v 1 1				v 1 10				v 1 7
		v 2 2				v 2 4				v 2 7
		v 3 3				v 3 4				v 3 7
		v 4 4				v 4 4				v 4 7
		v 5 4				v 5 4				v 5 6
		v 6 5				v 6 3				e 0 5 146
		v 7 6				v 7 3				e 1 5 195
		e 0 6 64				v 8 5				e 2 5 144
		e 1 6 47				v 9 3				e 3 5 196
		e 2 7 17				v 10 6				e 4 5 178
		e 3 7 87				v 11 6		#6	26	t # 25 6
		e 4 7 122				e 0 8 10				v 0 7
		e 5 6 116				e 1 8 28				v 1 7
		e 6 7 41				e 2 10 197				v 2 7
#0	2	t # 1 0				e 3 10 176				v 3 7
		v 0 2				e 4 11 154				v 4 7

		v 1 3				e 5 11 167				v 5 6
		v 2 4				e 6 9 96				e 0 5 156
		v 3 4				e 7 9 1				e 1 5 142
		v 4 0				e 8 10 115				e 2 5 171
		v 5 1				e 9 11 70				e 3 5 150
		v 6 6				e 10 11 120				e 4 5 198
		v 7 5		#3	14	t # 13 3		#6	27	t # 26 6
		e 0 6 13				v 0 1				v 0 7
		e 1 6 62				v 1 10				v 1 7
		e 2 6 73				v 2 4				v 2 7
		e 3 7 110				v 3 4				v 3 7
		e 4 7 74				v 4 4				v 4 7
		e 5 7 42				v 5 4				v 5 6
		e 6 7 39				v 6 3				e 0 5 158
#0	3	t # 2 0				v 7 3				e 1 5 181
		v 0 3				v 8 5				e 2 5 157
		v 1 4				v 9 6				e 3 5 141
		v 2 4				v 10 3				e 4 5 160
		v 3 0				v 11 6		#6	28	t # 27 6
		v 4 1				e 0 8 4				v 0 7
		v 5 2				e 1 8 26				v 1 7
		v 6 6				e 2 9 128				v 2 7
		v 7 5				e 3 9 118				v 3 7
		e 0 6 45				e 4 11 86				v 4 7
		e 1 6 84				e 5 11 86				v 5 6
		e 2 7 136				e 6 10 48				e 0 5 153

		e 3 7 166				e 7 10 2				e 1 5 164
		e 4 7 71				e 8 9 58				e 2 5 93
		e 5 6 36				e 9 11 101				e 3 5 184
		e 6 7 66				e 10 11 37				e 4 5 179
#0	4	t # 3 0		#3	15	t # 14 3		#7	29	t # 28 7
		v 0 2				v 0 2				v 0 5
		v 1 3				v 1 3				v 1 13
		v 2 4				v 2 3				v 2 13
		v 3 4				v 3 4				v 3 3
		v 4 0				v 4 4				v 4 13
		v 5 1				v 5 4				v 5 13
		v 6 5				v 6 4				v 6 6
		v 7 6				v 7 10				e 0 6 188
		e 0 7 163				v 8 1				e 1 6 192
		e 1 7 204				v 9 3				e 2 6 103
		e 2 7 136				v 10 6				e 3 6 69
		e 3 6 210				v 11 6				e 4 6 102
		e 4 6 162				v 12 5				e 5 6 165
		e 5 6 36				e 0 10 32		#7	30	t # 29 7
		e 6 7 66				e 1 9 5				v 0 3
#1	5	t # 4 1				e 2 9 107				v 1 13
		v 0 7				e 3 11 173				v 2 13
		v 1 7				e 4 11 186				v 3 5
		v 2 8				e 5 10 189				v 4 13
		v 3 8				e 6 10 174				v 5 13
		v 4 8				e 7 12 67				v 6 6

		v 5 8				e 8 12 3				e 0 6 69
		v 6 7				e 9 11 53				e 1 6 102
		v 7 6				e 10 11 190				e 2 6 165
		v 8 6				e 10 12 121				e 3 6 188
		e 0 7 121		#3	16	t # 15 3				e 4 6 192
		e 1 7 106				v 0 3				e 5 6 103
		e 2 8 18				v 1 3		#7	31	t # 30 7
		e 3 8 43				v 2 4				v 0 5
		e 4 8 14				v 3 4				v 1 13
		e 5 8 35				v 4 4				v 2 13
		e 6 7 112				v 5 4				v 3 3
		e 7 8 175				v 6 10				v 4 13
#1	6	t # 5 1				v 7 1				v 5 13
		v 0 8				v 8 5				v 6 6
		v 1 7				v 9 3				e 0 6 188
		v 2 7				v 10 6				e 1 6 192
		v 3 7				v 11 6				e 2 6 103
		v 4 8				e 0 9 0				e 3 6 69
		v 5 8				e 1 9 91				e 4 6 102
		v 6 8				e 2 11 183				e 5 6 131
		v 7 6				e 3 11 182		#7	32	t # 31 7
		v 8 6				e 4 10 202				v 0 5
		e 0 7 7				e 5 10 207				v 1 13
		e 1 8 82				e 6 8 48				v 2 13
		e 2 8 108				e 7 8 16				v 3 3
		e 3 8 75				e 8 10 125				v 4 13

		e 4 7 8				e 9 11 56				v 5 13
		e 5 7 9				e 10 11 159				v 6 6
		e 6 7 6		#4	17	t # 16 4				e 0 6 191
		e 7 8 135				v 0 5				e 1 6 143
#1	7	t # 6 1				v 1 11				e 2 6 105
		v 0 7				v 2 12				e 3 6 65
		v 1 7				v 3 12				e 4 6 97
		v 2 8				v 4 11				e 5 6 130
		v 3 8				v 5 6		#8	33	t # 32 8
		v 4 8				v 6 6				v 0 13
		v 5 8				e 0 5 29				v 1 3
		v 6 7				e 1 5 80				v 2 3
		v 7 6				e 2 6 79				v 3 3
		v 8 6				e 3 6 81				v 4 13
		e 0 7 149				e 4 5 76				v 5 5
		e 1 7 100				e 5 6 29				v 6 6
		e 2 8 31		#4	18	t # 17 4				v 7 3
		e 3 8 15				v 0 5				e 0 6 205
		e 4 8 22				v 1 11				e 1 7 78
		e 5 8 20				v 2 12				e 2 7 77
		e 6 7 126				v 3 12				e 3 7 104
		e 7 8 180				v 4 11				e 4 6 185
#1	8	t # 7 1				v 5 6				e 5 6 200
		v 0 7				v 6 6				e 6 7 33
		v 1 7				e 0 5 30		#8	34	t # 33 8
		v 2 7				e 1 5 123				v 0 13

		v 3 8				e 2 6 109				v 1 3
		v 4 8				e 3 6 115				v 2 3
		v 5 8				e 4 5 88				v 3 3
		v 6 8				e 5 6 23				v 4 13
		v 7 6		#4	19	t # 18 4				v 5 5
		v 8 6				v 0 11				v 6 3
		e 0 7 114				v 1 5				v 7 6
		e 1 7 124				v 2 11				e 0 7 140
		e 2 7 129				v 3 12				e 1 6 38
		e 3 8 34				v 4 12				e 2 6 24
		e 4 8 12				v 5 6				e 3 6 47
		e 5 8 27				v 6 6				e 4 7 199
		e 6 8 25				e 0 5 99				e 5 7 194
		e 7 8 155				e 1 5 60				e 6 7 72
#2	9	t # 8 2				e 2 5 85		#8	35	t # 34 8
		v 0 9				e 3 6 95				v 0 13
		v 1 9				e 4 6 113				v 1 5
		v 2 9				e 5 6 63				v 2 13
		v 3 6		#4	20	t # 19 4				v 3 3
		e 0 3 134				v 0 12				v 4 3
		e 1 3 59				v 1 11				v 5 3
		e 2 3 152				v 2 5				v 6 6
#2	10	t # 9 2				v 3 11				v 7 3
		v 0 9				v 4 12				e 0 6 177
		v 1 9				v 5 6				e 1 6 169
		v 2 9				v 6 6				e 2 6 187

		v 3 6				e 0 5 111				e 3 7 68
		e 0 3 55				e 1 6 83				e 4 7 49
		e 1 3 161				e 2 6 21				e 5 7 57
		e 2 3 193				e 3 6 119				e 6 7 58
#2	11	t # 10 2				e 4 5 117		#8	36	t # 35 8
		v 0 9				e 5 6 19				v 0 5
		v 1 9		#5	21	t # 20 5				v 1 13
		v 2 9				v 0 7				v 2 3
		v 3 6				v 1 7				v 3 3
		e 0 3 52				v 2 7				v 4 3
		e 1 3 147				v 3 7				v 5 13
		e 2 3 127				v 4 7				v 6 6
#2	12	t # 11 2				v 5 6				v 7 3
		v 0 9				e 0 5 211				e 0 6 138
		v 1 9				e 1 5 92				e 1 6 148
		v 2 9				e 2 5 98				e 2 7 54
		v 3 6				e 3 5 137				e 3 7 50
		e 0 3 133				e 4 5 94				e 4 7 51
		e 1 3 40		#5	22	t # 21 5				e 5 6 203
		e 2 3 89				v 0 7				e 6 7 11
						v 1 7				
						v 2 7				
						v 3 7				
						v 4 7				
						v 5 6				
						e 0 5 170				

						e 1 5 132				
						e 2 5 208				
						e 3 5 168				
						e 4 5 206				
				#5	23	t # 22 5				
						v 0 7				
						v 1 7				
						v 2 7				
						v 3 7				
						v 4 7				
						v 5 6				
						e 0 5 201				
						e 1 5 145				
						e 2 5 209				
						e 3 5 139				
						e 4 5 172				
				#5	24	t # 23 5				
						v 0 7				
						v 1 7				
						v 2 7				
						v 3 7				
						v 4 7				
						v 5 6				
						e 0 5 61				
						e 1 5 90				
						e 2 5 151				

						e 3 5 44				
						e 4 5 46				

Bảng 5: Cơ sở dữ liệu đồ thị SIS

2. Chia tập dữ liệu học ra thành 4 nhóm khác nhau:

Nhóm 1:

Nhóm	**Lớp**	**Đồ Thị**	**Chi tiết**
1	#0	1	t # 0 0
1	#1	5	t # 4 1
1	#2	9	t # 8 2
1	#3	13	t # 12 3
1	#4	17	t # 16 4
1	#5	21	t # 20 5
1	#6	25	t # 24 6
1	#7	29	t # 28 7
1	#8	33	t # 32 8

Nhóm 2:

Nhóm	**Lớp**	**Đồ Thị**	**Chi tiết**
2	#0	2	t # 1 0
2	#1	6	t # 5 1
2	#2	10	t # 9 2
2	#3	14	t # 13 3
2	#4	18	t # 17 4
2	#5	22	t # 21 5

2	#6	26	t # 25 6
2	#7	30	t # 29 7
2	#8	34	t # 33 8

Nhóm 3:

Nhóm	**Lớp**	**Đồ Thị**	**Chi tiết**
3	#0	3	t # 2 0
3	#1	7	t # 6 1
3	#2	11	t # 10 2
3	#3	15	t # 14 3
3	#4	19	t # 18 4
3	#5	23	t # 22 5
3	#6	27	t # 26 6
3	#7	31	t # 30 7
3	#8	35	t # 34 8

Nhóm 4:

Nhóm	**Lớp**	**Đồ Thị**	**Chi tiết**
4	#0	4	t # 3 0
4	#1	8	t # 7 1
4	#2	12	t # 11 2
4	#3	16	t # 15 3
4	#4	20	t # 19 4
4	#5	24	t # 23 5
4	#6	28	t # 27 6
4	#7	32	t # 31 7

4	#8	36	t # 35 8

Bảng 6: Cơ sở dữ liệu đồ thị SIS đã phân nhóm ngẫu nhiên

3. **Duyệt từng nhóm để tính độ hiệu quả.**

   a. Duyệt nhóm 1:

- Tập dữ liệu huấn luyện gồm tất cả các đồ thị trong nhóm 2, 3, 4
- Tập dữ liệu kiểm tra gồm tất cả các đồ thị trong nhóm 1.
- Huấn luyện sự phân lớp sử dụng tất cả đồ thị trong tập dữ liệu huấn luyện.

Sử dụng thuật toán để tìm tất cả các MPBCĐ của các đồ thị trong tập dữ liệu huấn luyện:

Đồ Thị	**Đồ thị phổ biến cực đại**
2	Pattern: ['3', '7', '5', '4', '6', '1', '0', '2']/4-5-1-0-6-3-2-4
3	Pattern: ['1', '6', '0', '5', '7', '3', '2', '4']/4-6-3-2-5-0-4-1
4	Pattern: ['3', '6', '5', '4', '7', '1', '0', '2']/4-5-1-0-6-3-2-4
6	Pattern: ['1', '8', '3', '2', '7', '0', '6', '5', '4']/7-6-7-7-6-8-8-8-8
7	Pattern: ['1', '7', '0', '6', '8', '3', '2', '5', '4']/7-6-7-7-6-8-8-8-8
8	Pattern: ['1', '7', '0', '2', '8', '3', '5', '4', '6']/7-6-7-7-6-8-8-8-8
10	Pattern: ['1', '3', '0', '2']/9-6-9-9
11	Pattern: ['1', '3', '0', '2']/9-6-9-9
12	Pattern: ['1', '3', '0', '2']/9-6-9-9
14	Pattern: ['11', '9', '10', '5', '4', '8', '3', '2', '7', '6', '1', '0']/6-6-3-4-4-5-4-4-3-3-10-1
15	Pattern: ['11', '9', '10', '3', '4', '1', '2', '0', '12', '5', '6', '8', '7']/6-3-6-4-4-3-3-2-5-4-4-1-10
16	Pattern: ['11', '9', '10', '3', '2', '1', '0', '8', '5', '4', '7', '6']/6-3-6-4-4-3-3-5-4-4-

	1-10
18	Pattern: ['1', '5', '0', '4', '6', '3', '2']/11-6-5-11-6-12-12
19	Pattern: ['0', '5', '1', '2', '6', '3', '4']/11-6-5-11-6-12-12
20	Pattern: ['1', '6', '3', '2', '5', '0', '4']/11-6-11-5-6-12-12
22	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
23	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
24	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
26	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
27	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
28	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
30	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-3-5-13-13-13
31	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-5-3-13-13-13
32	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-5-3-13-13-13
34	Pattern: ['1', '6', '3', '2', '7', '0', '5', '4']/3-3-3-3-6-13-5-13
35	Pattern: ['0', '6', '1', '2', '7', '3', '5', '4']/13-6-5-13-3-3-3-3
36	Pattern: ['1', '6', '0', '5', '7', '3', '2', '4']/13-6-5-13-3-3-3-3

Bảng 7: Mẫu phổ biến của tập huấn luyện nhóm 2, 3, 4

Loại trừ các MPBCĐ mà xuất hiện trong nhiều hơn 1 lớp:

Nhóm	**Lớp**	**Đồ Thị**	**Mẫu Phổ Biến cực đại**	**Loại**
2	#0	2	4-5-1-0-6-3-2-4	
3	#0	3	4-6-3-2-5-0-4-1	
4	#0	4	4-5-1-0-6-3-2-4	
2	#1	6	7-6-7-7-6-8-8-8-8	
3	#1	7	7-6-7-7-6-8-8-8-8	

4	#1	8	7-6-7-7-6-8-8-8-8	
2	#2	10	9-6-9-9	
3	#2	11	9-6-9-9	
4	#2	12	9-6-9-9	
2	#3	14	6-6-3-4-4-5-4-4-3-3-10-1	
3	#3	15	6-3-6-4-4-3-3-2-5-4-4-1-10	
4	#3	16	6-3-6-4-4-3-3-5-4-4-1-10	
2	#4	18	11-6-5-11-6-12-12	
3	#4	19	11-6-5-11-6-12-12	
4	#4	20	11-6-11-5-6-12-12	
2	#5	22	7-6-7-7-7-7	Loại
3	#5	23	7-6-7-7-7-7	Loại
4	#5	24	7-6-7-7-7-7	Loại
2	#6	26	7-6-7-7-7-7	Loại
3	#6	27	7-6-7-7-7-7	Loại
4	#6	28	7-6-7-7-7-7	Loại
2	#7	30	13-6-3-5-13-13-13	
3	#7	31	13-6-5-3-13-13-13	
4	#7	32	13-6-5-3-13-13-13	
2	#8	34	3-3-3-3-6-13-5-13	
3	#8	35	13-6-5-13-3-3-3-3	
4	#8	36	13-6-5-13-3-3-3-3	

Bảng 8: Loại trừ các mẫu xuất hiện nhiều hơn một lần trong nhóm 2, 3, 4

Rút gọn, sắp xếp MPBCĐ tìm được để tạo ra một tập tối ưu các MPBCĐ:

Mẫu	Đồ thị	Đồ thị phổ biến cực đại	Mẫu phổ biến cực đại
1	2	['3', '7', '5', '4', '6', '1', '0', '2']	4-5-1-0-6-3-2-4
2	6	['1', '8', '3', '2', '7', '0', '6', '5', '4']	7-6-7-7-6-8-8-8-8
3	10	['1', '3', '0', '2']	9-6-9-9
4	14	['11', '9', '10', '5', '4', '8', '3', '2', '7', '6', '1', '0']	6-6-3-4-4-5-4-4-3-3-10-1
5	15	['11', '9', '10', '3', '4', '1', '2', '0', '12', '5', '6', '8', '7']	6-3-6-4-4-3-3-2-5-4-4-1-10
6	18	['1', '5', '0', '4', '6', '3', '2']	11-6-5-11-6-12-12
7	30	['1', '6', '0', '3', '2', '5', '4']	13-6-5-3-13-13-13
8	34	['1', '6', '3', '2', '7', '0', '5', '4']	13-6-5-13-3-3-3-3

Bảng 9: Mẫu phổ biến cực đại tối ưu của nhóm 2, 3, 4

Tính toán độ khác nhau của từng đồ thị trong dữ liệu kiểm tra (nhóm 1) với từng MPBCĐ tìm được trong tập huấn luyện. Sau đó dự đoán MPBCĐ tương ứng cho từng đồ thị trong dữ liệu kiểm tra bằng cách so sánh độ khác nhau với NTĐ ($\Delta = 6$) (độ khác nhau phải nhỏ hơn hoặc bằng NTĐ Δ).

Nhóm 1	Mẫu 1	Mẫu 2	Mẫu 3	Mẫu 4	Mẫu 5	Mẫu 6	Mẫu 7	Mẫu 8	Mẫu dự đoán	Lớp dự đoán
1	0	15	10	9	8	11	9	10	1	#0
5	15	0	11	17	18	12	14	15	2	#1
9	10	11	0	14	15	9	9	10	3	#2
13	9	17	14	0	1	13	13	10	4	#3
17	11	12	9	13	14	0	10	11	6	#4

21	12	7	8	16	17	11	11	12	**NA**	**NA**
25	12	7	8	16	17	11	11	12	**NA**	**NA**
29	9	14	9	13	14	10	0	5	7	#7
33	10	15	10	10	11	11	5	0	8	#8

Bảng 10: Độ tương đồng các đồ thị nhóm 1 và các mẫu của nhóm 2, 3, 4

- Tính toán độ sai lệch n_1, là số lượng đồ thị trong nhóm 1 bị phân loại Sai.

Đồ Thị Nhóm 1	Lớp	Lớp Dự Đoán	Kết Quả
1	#0	#0	Đúng
5	#1	#1	Đúng
9	#2	#2	Đúng
13	#3	#3	Đúng
17	#4	#4	Đúng
21	#5	**NA**	**Sai**
25	#6	**NA**	**Sai**
29	#7	#7	Đúng
33	#8	#8	Đúng

Bảng 11: Kết quả phân lớp các đồ thị nhóm 1

Kết luận: Vậy độ sai lệch trong nhóm 1 là $\mathbf{n_1=2}$

b. Duyệt nhóm 2:

- Tập dữ liệu huấn luyện gồm tất tất cả các đồ thị trong nhóm 1, 3, 4
- Tập dữ liệu kiểm tra gồm tất cả các đồ thị trong nhóm 2.
- Huấn luyện sự phân lớp sử dụng tất cả đồ thị trong tập dữ liệu huấn luyện.

Sẽ sử dụng thuật toán để tìm tất cả các MPBCĐ của các đồ thị trong tập dữ liệu huấn luyện:

Đồ Thị	**Đồ thị phổ biến cực đại**
1	Pattern: ['5', '6', '1', '0', '7', '3', '2', '4']/4-5-1-0-6-3-2-4
3	Pattern: ['1', '6', '0', '5', '7', '3', '2', '4']/4-6-3-2-5-0-4-1
4	Pattern: ['3', '6', '5', '4', '7', '1', '0', '2']/4-5-1-0-6-3-2-4
5	Pattern: ['1', '7', '0', '6', '8', '3', '2', '5', '4']/7-6-7-7-6-8-8-8-8
7	Pattern: ['1', '7', '0', '6', '8', '3', '2', '5', '4']/7-6-7-7-6-8-8-8-8
8	Pattern: ['1', '7', '0', '2', '8', '3', '5', '4', '6']/7-6-7-7-6-8-8-8-8
9	Pattern: ['1', '3', '0', '2']/9-6-9-9
11	Pattern: ['1', '3', '0', '2']/9-6-9-9
12	Pattern: ['1', '3', '0', '2']/9-6-9-9
13	Pattern: ['11', '9', '10', '5', '4', '7', '6', '8', '3', '2', '1', '0']/6-3-6-4-4-3-3-5-4-4-10-1
15	Pattern: ['11', '9', '10', '3', '4', '1', '2', '0', '12', '5', '6', '8', '7']/6-3-6-4-4-3-3-2-5-4-4-1-10
16	Pattern: ['11', '9', '10', '3', '2', '1', '0', '8', '5', '4', '7', '6']/6-3-6-4-4-3-3-5-4-4-1-10
17	Pattern: ['1', '5', '0', '4', '6', '3', '2']/11-6-5-11-6-12-12
19	Pattern: ['0', '5', '1', '2', '6', '3', '4']/11-6-5-11-6-12-12
20	Pattern: ['1', '6', '3', '2', '5', '0', '4']/11-6-11-5-6-12-12
21	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
23	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
24	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
25	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
27	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7

28	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
29	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-5-3-13-13-13
31	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-5-3-13-13-13
32	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-5-3-13-13-13
33	Pattern: ['1', '7', '3', '2', '6', '0', '5', '4']/3-3-3-3-6-13-5-13
35	Pattern: ['0', '6', '1', '2', '7', '3', '5', '4']/13-6-5-13-3-3-3-3
36	Pattern: ['1', '6', '0', '5', '7', '3', '2', '4']/13-6-5-13-3-3-3-3

Bảng 12: Mẫu phổ biến của tập huấn luyện nhóm 1, 3, 4

Loại trừ các MPBCĐ mà xuất hiện trong nhiều hơn 1 lớp:

Nhóm	**Lớp**	**Đồ Thị**	**Mẫu Phổ Biến cực đại**	**Loại**
1	#0	1	4-5-1-0-6-3-2-4	
3	#0	3	4-6-3-2-5-0-4-1	
4	#0	4	4-5-1-0-6-3-2-4	
1	#1	5	7-6-7-7-6-8-8-8-8	
3	#1	7	7-6-7-7-6-8-8-8-8	
4	#1	8	7-6-7-7-6-8-8-8-8	
1	#2	9	9-6-9-9	
3	#2	11	9-6-9-9	
4	#2	12	9-6-9-9	
1	#3	13	6-3-6-4-4-3-3-5-4-4-10-1	
3	#3	15	6-3-6-4-4-3-3-2-5-4-4-1-10	
4	#3	16	6-3-6-4-4-3-3-5-4-4-1-10	
1	#4	17	11-6-5-11-6-12-12	
3	#4	19	11-6-5-11-6-12-12	

4	#4	20	11-6-11-5-6-12-12	
1	#5	21	7-6-7-7-7-7	**Loại**
3	#5	23	7-6-7-7-7-7	**Loại**
4	#5	24	7-6-7-7-7-7	**Loại**
1	#6	25	7-6-7-7-7-7	**Loại**
3	#6	27	7-6-7-7-7-7	**Loại**
4	#6	28	7-6-7-7-7-7	**Loại**
1	#7	29	13-6-5-3-13-13-13	
3	#7	31	13-6-5-3-13-13-13	
4	#7	32	13-6-5-3-13-13-13	
1	#8	33	3-3-3-3-6-13-5-13	
3	#8	35	13-6-5-13-3-3-3-3	
4	#8	36	13-6-5-13-3-3-3-3	

Bảng 13: Loại trừ các mẫu xuất hiện nhiều hơn một lần trong nhóm 1, 3, 4

Rút gọn, sắp xếp MPBCĐ tìm được để tạo ra một tập tối ưu các MPBCĐ:

Mẫu	**Đồ thị**	**Đồ Thị Phổ Biến Cực Đại**	**Mẫu Phổ Biến Cực Đại**
1	1	['5', '6', '1', '0', '7', '3', '2', '4']	4-5-1-0-6-3-2-4
2	5	['1', '7', '0', '6', '8', '3', '2', '5', '4']	7-6-7-7-6-8-8-8-8
3	9	['1', '3', '0', '2']	9-6-9-9
4	13	['11', '9', '10', '5', '4', '7', '6', '8', '3', '2', '1', '0']	6-3-6-4-4-3-3-5-4-4-10-1
5	15	['11', '9', '10', '3', '4', '1', '2', '0', '12', '5', '6', '8', '7']	6-3-6-4-4-3-3-2-5-4-4-1-10
6	17	['1', '5', '0', '4', '6', '3', '2']	11-6-5-11-6-12-12
7	29	['1', '6', '0', '3', '2', '5', '4']	13-6-5-3-13-13-13
8	33	['1', '7', '3', '2', '6', '0', '5', '4']	3-3-3-3-6-13-5-13

Bảng 14: Mẫu phổ biến cực đại tối ưu của nhóm 1, 3, 4

Tính toán độ khác nhau của từng đồ thị trong dữ liệu kiểm tra (nhóm 2) với từng MPBCĐ tìm được trong tập huấn luyện. Sau đó dự đoán MPBCĐ tương ứng cho từng đồ thị trong dữ liệu kiểm tra bằng cách so sánh độ khác nhau với NTĐ ($\Delta = 6$) (độ khác nhau phải nhỏ hơn hoặc bằng NTĐ Δ).

Nhóm 2	Mẫu 1	Mẫu 2	Mẫu 3	Mẫu 4	Mẫu 5	Mẫu 6	Mẫu 7	Mẫu 8	Mẫu dự đoán	Lớp dự đoán
2	0	15	10	9	8	11	9	10	1	#0
6	15	0	11	17	18	12	14	15	2	#1
10	10	11	0	14	15	9	9	10	3	#2
14	9	17	14	0	1	13	13	0	4	#3
18	11	12	9	13	14	0	10	11	6	#4
22	12	7	8	16	17	11	11	12	**NA**	**NA**
26	12	7	8	16	17	11	11	12	**NA**	**NA**
30	9	14	9	13	14	10	0	5	7	#7
34	10	15	10	10	11	11	5	0	8	#8

Bảng 15: Độ tương đồng các đồ thị nhóm 2 và các mẫu của nhóm 1, 3, 4

- Tính toán độ sai lệch n_2, là số lượng đồ thị trong nhóm 1 bị phân loại Sai.

Đồ Thị Nhóm 2	Lớp	Lớp Dự Đoán	Kết Quả
2	#0	#0	Đúng
6	#1	#1	Đúng

10	#2	#2	Đúng
14	#3	#3	Đúng
18	#4	#4	Đúng
22	#5	**NA**	**Sai**
26	#6	**NA**	**Sai**
30	#7	#7	Đúng
34	#8	#8	Đúng

Bảng 16: Kết quả phân lớp các đồ thị nhóm 2

Kết luận: Vậy độ sai lệch trong nhóm 2 là $\mathbf{n_2=2}$

c. Duyệt nhóm 3:

- Tập dữ liệu huấn luyện gồm tất tất cả các đồ thị trong nhóm 1, 2, 4
- Tập dữ liệu kiểm tra gồm tất cả các đồ thị trong nhóm 3.
- Huấn luyện sự phân lớp sử dụng tất cả đồ thị trong tập dữ liệu huấn luyện.

Sẽ sử dụng thuật toán để tìm tất cả các MPBCĐ của các đồ thị trong tập dữ liệu huấn luyện:

Đồ Thị	**Đồ thị phổ biến cực đại**
1	Pattern: ['5', '6', '1', '0', '7', '3', '2', '4']/4-5-1-0-6-3-2-4
2	Pattern: ['3', '7', '5', '4', '6', '1', '0', '2']/4-5-1-0-6-3-2-4
4	Pattern: ['3', '6', '5', '4', '7', '1', '0', '2']/4-5-1-0-6-3-2-4
5	Pattern: ['1', '7', '0', '6', '8', '3', '2', '5', '4']/7-6-7-7-6-8-8-8-8
6	Pattern: ['1', '8', '3', '2', '7', '0', '6', '5', '4']/7-6-7-7-6-8-8-8-8
8	Pattern: ['1', '7', '0', '2', '8', '3', '5', '4', '6']/7-6-7-7-6-8-8-8-8
9	Pattern: ['1', '3', '0', '2']/9-6-9-9
10	Pattern: ['1', '3', '0', '2']/9-6-9-9

12	Pattern: ['1', '3', '0', '2']/9-6-9-9
13	Pattern: ['11', '9', '10', '5', '4', '7', '6', '8', '3', '2', '1', '0']/6-3-6-4-4-3-3-5-4-4-10-1
14	Pattern: ['11', '9', '10', '5', '4', '8', '3', '2', '7', '6', '1', '0']/6-6-3-4-4-5-4-4-3-3-10-1
16	Pattern: ['11', '9', '10', '3', '2', '1', '0', '8', '5', '4', '7', '6']/6-3-6-4-4-3-3-5-4-4-1-10
17	Pattern: ['1', '5', '0', '4', '6', '3', '2']/11-6-5-11-6-12-12
18	Pattern: ['1', '5', '0', '4', '6', '3', '2']/11-6-5-11-6-12-12
20	Pattern: ['1', '6', '3', '2', '5', '0', '4']/11-6-11-5-6-12-12
21	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
22	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
24	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
25	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
26	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
28	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
29	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-5-3-13-13-13
30	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-3-5-13-13-13
32	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-5-3-13-13-13
33	Pattern: ['1', '7', '3', '2', '6', '0', '5', '4']/3-3-3-3-6-13-5-13
34	Pattern: ['1', '6', '3', '2', '7', '0', '5', '4']/3-3-3-3-6-13-5-13
36	Pattern: ['1', '6', '0', '5', '7', '3', '2', '4']/13-6-5-13-3-3-3-3

Bảng 17: Mẫu phổ biến của tập huấn luyện nhóm 1, 2, 4

Loại trừ các MPBCĐ mà xuất hiện trong nhiều hơn 1 lớp:

Nhóm	Lớp	Đồ Thị	Mẫu Phổ Biến cực đại	Loại
1	#0	1	4-5-1-0-6-3-2-4	
2	#0	2	4-5-1-0-6-3-2-4	
4	#0	4	4-5-1-0-6-3-2-4	
1	#1	5	7-6-7-7-6-8-8-8-8	
2	#1	6	7-6-7-7-6-8-8-8-8	
4	#1	8	7-6-7-7-6-8-8-8-8	
1	#2	9	9-6-9-9	
2	#2	10	9-6-9-9	
4	#2	12	9-6-9-9	
1	#3	13	6-3-6-4-4-3-3-5-4-4-10-1	
2	#3	14	6-6-3-4-4-5-4-4-3-3-10-1	
4	#3	16	6-3-6-4-4-3-3-5-4-4-1-10	
1	#4	17	11-6-5-11-6-12-12	
2	#4	18	11-6-5-11-6-12-12	
4	#4	20	11-6-11-5-6-12-12	
1	#5	21	7-6-7-7-7-7	Loại
2	#5	22	7-6-7-7-7-7	Loại
4	#5	24	7-6-7-7-7-7	Loại
1	#6	25	7-6-7-7-7-7	Loại
2	#6	26	7-6-7-7-7-7	Loại
4	#6	28	7-6-7-7-7-7	Loại
1	#7	29	13-6-5-3-13-13-13	
2	#7	30	13-6-3-5-13-13-13	
4	#7	32	13-6-5-3-13-13-13	
1	#8	33	3-3-3-3-6-13-5-13	

2	#8	34	3-3-3-3-6-13-5-13	
4	#8	36	13-6-5-13-3-3-3-3	

Bảng 18: Loại trừ các mẫu xuất hiện nhiều hơn một lần trong nhóm 1, 2, 4

Rút gọn, sắp xếp MPBCĐ tìm được để tạo ra một tập tối ưu các MPBCĐ:

Mẫu	**Đồ Thị**	**Đồ thị phổ biến cực đại**	**Mẫu Phổ Biến cực đại**
1	1	['5', '6', '1', '0', '7', '3', '2', '4']	4-5-1-0-6-3-2-4
2	5	['1', '7', '0', '6', '8', '3', '2', '5', '4']	7-6-7-7-6-8-8-8-8
3	9	['1', '3', '0', '2']	9-6-9-9
4	13	['11', '9', '10', '5', '4', '7', '6', '8', '3', '2', '1', '0']	6-3-6-4-4-3-3-5-4-4-10-1
5	17	['1', '5', '0', '4', '6', '3', '2']	11-6-5-11-6-12-12
6	29	['1', '6', '0', '3', '2', '5', '4']	13-6-5-3-13-13-13
7	33	['1', '7', '3', '2', '6', '0', '5', '4']	3-3-3-3-6-13-5-13

Bảng 19: Mẫu phổ biến cực đại tối ưu của nhóm 1, 2, 4

Tính toán độ khác nhau của từng đồ thị trong dữ liệu kiểm tra (nhóm 3) với từng MPBCĐ tìm được trong tập huấn luyện. Sau đó dự đoán MPBCĐ tương ứng cho từng đồ thị trong dữ liệu kiểm tra bằng cách so sánh độ khác nhau với NTĐ ($\Delta = 6$) (độ khác nhau phải nhỏ hơn hoặc bằng NTĐ Δ).

Nhóm 3	Mẫu 1	Mẫu 2	Mẫu 3	Mẫu 4	Mẫu 5	Mẫu 6	Mẫu 7	Mẫu dự đoán	Lớp dự đoán

3	0	15	10	9	11	9	10	1	#0
7	15	0	11	17	12	14	15	2	#1
11	10	11	0	14	9	9	10	3	#2
15	8	18	15	1	14	14	11	4	#3
19	11	12	9	13	0	10	11	5	#4
23	12	7	8	16	11	11	12	**NA**	**NA**
27	12	7	8	16	11	11	12	**NA**	**NA**
31	9	14	9	13	10	0	5	6	#7
35	10	15	10	10	11	5	0	7	#8

Bảng 20: Độ tương đồng các đồ thị nhóm 2 và các mẫu của nhóm 1, 2, 4

- Tính toán độ sai lệch n_3, là số lượng đồ thị trong nhóm 1 bị phân loại Sai.

Đồ Thị Nhóm 3	Lớp	Lớp Dự Đoán	Kết Quả
3	#0	#0	Đúng
7	#1	#1	Đúng
11	#2	#2	Đúng
15	#3	#3	Đúng
19	#4	#4	Đúng
23	#5	**NA**	**Sai**
27	#6	**NA**	**Sai**
31	#7	#7	Đúng
35	#8	#8	Đúng

Bảng 21: Kết quả phân lớp các đồ thị nhóm 3

Kết luận: Vậy độ sai lệch trong nhóm 3 là $\mathbf{n_3=2}$

d. Duyệt nhóm 4:

- Tập dữ liệu huấn luyện gồm tất tất cả các đồ thị trong nhóm 1, 2, 3
- Tập dữ liệu kiểm tra gồm tất cả các đồ thị trong nhóm 4.
- Huấn luyện sự phân lớp sử dụng tất cả đồ thị trong tập dữ liệu huấn luyện.

Sẽ sử dụng thuật toán để tìm tất cả các MPBCĐ của các đồ thị trong tập dữ liệu huấn luyện:

Đồ Thị	**Đồ thị phổ biến cực đại**
1	Pattern: ['5', '6', '1', '0', '7', '3', '2', '4']/4-5-1-0-6-3-2-4
2	Pattern: ['3', '7', '5', '4', '6', '1', '0', '2']/4-5-1-0-6-3-2-4
3	Pattern: ['1', '6', '0', '5', '7', '3', '2', '4']/4-6-3-2-5-0-4-1
5	Pattern: ['1', '7', '0', '6', '8', '3', '2', '5', '4']/7-6-7-7-6-8-8-8-8
6	Pattern: ['1', '8', '3', '2', '7', '0', '6', '5', '4']/7-6-7-7-6-8-8-8-8
7	Pattern: ['1', '7', '0', '6', '8', '3', '2', '5', '4']/7-6-7-7-6-8-8-8-8
9	Pattern: ['1', '3', '0', '2']/9-6-9-9
10	Pattern: ['1', '3', '0', '2']/9-6-9-9
11	Pattern: ['1', '3', '0', '2']/9-6-9-9
13	Pattern: ['11', '9', '10', '5', '4', '7', '6', '8', '3', '2', '1', '0']/6-3-6-4-4-3-3-5-4-4-10-1
14	Pattern: ['11', '9', '10', '5', '4', '8', '3', '2', '7', '6', '1', '0']/6-6-3-4-4-5-4-4-3-3-10-1
15	Pattern: ['11', '9', '10', '3', '4', '1', '2', '0', '12', '5', '6', '8', '7']/6-3-6-4-4-3-3-2-5-4-4-1-10
17	Pattern: ['1', '5', '0', '4', '6', '3', '2']/11-6-5-11-6-12-12
18	Pattern: ['1', '5', '0', '4', '6', '3', '2']/11-6-5-11-6-12-12
19	Pattern: ['0', '5', '1', '2', '6', '3', '4']/11-6-5-11-6-12-12

21	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
22	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
23	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
25	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
26	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
27	Pattern: ['1', '5', '0', '3', '2', '4']/7-6-7-7-7-7
29	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-5-3-13-13-13
30	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-3-5-13-13-13
31	Pattern: ['1', '6', '0', '3', '2', '5', '4']/13-6-5-3-13-13-13
33	Pattern: ['1', '7', '3', '2', '6', '0', '5', '4']/3-3-3-3-6-13-5-13
34	Pattern: ['1', '6', '3', '2', '7', '0', '5', '4']/3-3-3-3-6-13-5-13
35	Pattern: ['0', '6', '1', '2', '7', '3', '5', '4']/13-6-5-13-3-3-3-3

Bảng 22: Mẫu phổ biến của tập huấn luyện nhóm 1, 2, 3

Loại trừ các MPBCĐ mà xuất hiện trong nhiều hơn 1 lớp:

Nhóm	**Lớp**	**Đồ Thị**	**Mẫu Phổ Biến cực đại**	**Loai**
1	#0	1	4-5-1-0-6-3-2-4	
2	#0	2	4-5-1-0-6-3-2-4	
3	#0	3	4-6-3-2-5-0-4-1	
1	#1	5	7-6-7-7-6-8-8-8-8	
2	#1	6	7-6-7-7-6-8-8-8-8	
3	#1	7	7-6-7-7-6-8-8-8-8	
1	#2	9	9-6-9-9	
2	#2	10	9-6-9-9	
3	#2	11	9-6-9-9	

1	#3	13	6-3-6-4-4-3-3-5-4-4-10-1	
2	#3	14	6-6-3-4-4-5-4-4-3-3-10-1	
3	#3	15	6-3-6-4-4-3-3-2-5-4-4-1-10	
1	#4	17	11-6-5-11-6-12-12	
2	#4	18	11-6-5-11-6-12-12	
3	#4	19	11-6-5-11-6-12-12	
1	#5	21	7-6-7-7-7-7	Loại
2	#5	22	7-6-7-7-7-7	Loại
3	#5	23	7-6-7-7-7-7	Loại
1	#6	25	7-6-7-7-7-7	Loại
2	#6	26	7-6-7-7-7-7	Loại
3	#6	27	7-6-7-7-7-7	Loại
1	#7	29	13-6-5-3-13-13-13	
2	#7	30	13-6-3-5-13-13-13	
3	#7	31	13-6-5-3-13-13-13	
1	#8	33	3-3-3-3-6-13-5-13	
2	#8	34	3-3-3-3-6-13-5-13	
3	#8	35	13-6-5-13-3-3-3-3	

Bảng 23: Loại trừ các mẫu xuất hiện nhiều hơn một lần trong nhóm 1, 2, 3

Rút gọn, sắp xếp MPBCĐ tìm được để tạo ra một tập tối ưu các MPBCĐ:

Mẫu	**Đồ Thị**	**Đồ thị phổ biến cực đại**	**Mẫu Phổ Biến cực đại**
1	1	['5', '6', '1', '0', '7', '3', '2', '4']	4-5-1-0-6-3-2-4
2	5	['1', '7', '0', '6', '8', '3', '2', '5', '4']	7-6-7-7-6-8-8-8-8

3	9	['1', '3', '0', '2']	9-6-9-9
4	13	['11', '9', '10', '5', '4', '7', '6', '8', '3', '2', '1', '0']	6-3-6-4-4-3-3-5-4-4-10-1
5	15	['11', '9', '10', '3', '4', '1', '2', '0', '12', '5', '6', '8', '7']	6-3-6-4-4-3-3-2-5-4-4-1-10
6	17	['1', '5', '0', '4', '6', '3', '2']	11-6-5-11-6-12-12
7	29	['1', '6', '0', '3', '2', '5', '4']	13-6-5-3-13-13-13
8	33	['1', '7', '3', '2', '6', '0', '5', '4']	3-3-3-3-6-13-5-13

Bảng 24: Mẫu phổ biến cực đại tối ưu của nhóm 1, 2, 3

Tính toán độ khác nhau của từng đồ thị trong dữ liệu kiểm tra (nhóm 4) với từng MPBCĐ tìm được trong tập huấn luyện. Sau đó dự đoán MPBCĐ tương ứng cho từng đồ thị trong dữ liệu kiểm tra bằng cách so sánh độ khác nhau với NTĐ ($\Delta = 6$) (độ khác nhau phải nhỏ hơn hoặc bằng NTĐ Δ).

Nhóm 4	Mẫu 1	Mẫu 2	Mẫu 3	Mẫu 4	Mẫu 5	Mẫu 6	Mẫu 7	Mẫu 8	Mẫu dự đoán	Lớp dự đoán
4	0	15	10	9	8	11	9	10	1	#0
8	15	0	11	17	18	12	14	15	2	#1
12	10	11	0	14	15	9	9	10	3	#2
16	9	17	14	0	1	13	13	0	4	#3
20	11	12	9	13	14	0	10	11	6	#4
24	12	7	8	16	17	11	11	12	**NA**	**NA**
28	12	7	8	16	17	11	11	12	**NA**	**NA**
32	9	14	9	13	14	10	0	5	7	#7

36	10	15	10	10	11	11	5	0	8	#8

Bảng 25: Độ tương đồng các đồ thị nhóm 2 và các mẫu của nhóm 1, 2, 3

- Tính toán độ sai lệch n_4, là số lượng đồ thị trong nhóm 4 bị phân loại Sai.

Đồ Thị Nhóm 3	Lớp	Lớp Dự Đoán	Kết Quả
4	#0	#0	Đúng
8	#1	#1	Đúng
12	#2	#2	Đúng
16	#3	#3	Đúng
20	#4	#4	Đúng
24	#5	**NA**	**Sai**
28	#6	**NA**	**Sai**
32	#7	#7	Đúng
36	#8	#8	Đúng

Bảng 26: Kết quả phân lớp các đồ thị nhóm 4

Kết luận: Vậy độ sai lệch trong nhóm 4 là $\mathbf{n_4=2}$

4. Đánh giá tính hiệu quả của thuật toán:

Xác xuất sai lệch (E) của sự phân lớp trên là:

E = $\frac{\sum_{i=1}^{4} ni}{m}$ Trong đó: m – Số lượng đồ thị sử dụng m=36

E = $\frac{n1+n2+n3+n4}{36}$ = $\frac{2+2+2+2}{36}$ = $\frac{8}{36}$ = 0.2222

E = 22,22%

Vậy tính hiệu quả của thuật toán sẽ là:

P = 1 – E = 0.7778

P = 77,78%

3.3. So sánh kết quả ImaxAFG và MaxAFG

Dựa vào quy trình ”k-ford cross validation” như đã kiểm chứng ở trên (với k = 4 tương đương với dữ liệu đồ thị SIS; là một dạng dữ liệu hình khung có cấu trúc, chia ra làm 4 lớp, sử dụng NTS σ = 2 và NTĐ Δ = 6), độ chính xác phân loại trung bình của đồ thị lên đến 77,78% khi sử dụng thuật toán ImaxAFG.

Cùng với quy trình kiểm chứng ” k-ford cross validation” với bộ dữ liệu SIS và các chỉ số k = 4; NTS σ = 2; NTĐ Δ = 6, chỉ đạt được độ chính xác phân loại trung bình của đồ thị là 69,44% khi sử dụng thuật toán MaxAFG [18], và 66,6% khi sử dụng thuật toán không sử dụng phương pháp so sánh gần đúng gAppox.

Như vậy đối với hai thuật toán có sử dụng phương pháp so sánh gần đúng và không sử dụng phương pháp so sánh gần đúng, thì độ chính xác phân lớp của thuật toán đang nghiên cứu ImaxAFG cũng tốt hơn.

3.4. Kết luận và hướng phát triển

Trong bài Luận Văn này trình bày thuật toán ImaxAFG, một thuật toán khai thác MPBCĐ trong dồ thị đơn sử dụng phương pháp so sánh gần đúng. Bằng việc thừa nhận sự khác nhau về cấu trúc như các đỉnh cũng như các cạnh của đồ thị, giữa mẫu đồ thị phổ biến và các sự biểu diễn của nó, có thể tìm ra được các MPB còn sót bởi các thuật toán không sử dụng phương pháp so sánh gần đúng. Trong một khía cạnh khác, tập trung vào việc khai thác mẫu đồ thị cực đại giúp giảm số lượng mẫu đáng kể, đó là một vấn đề rất quan trọng bởi vì việc sử dụng phương pháp so sánh gần đúng thì số lượng MPB tìm được có thể tăng lên gấp 100 lần so với thuật toán không sử dụng phương pháp so sánh gần đúng.

Kết quả thí nghiệm cho thấy rằng, những MPBCĐ được tìm thấy bởi thuật toán ImaxAFG rất hữu dụng trong nhiều công việc như thực hiện việc phân lớp đồ thị; bởi

vậy nên có thể kết luận rằng mẫu đồ thị phổ biến cực đại được khai thác bằng phương pháp so sánh gần đúng có khả năng là những thông tin hữu dụng mà có thể bị bỏ sót khi sử dụng phương pháp so sánh chính xác tuyệt đối.

Một sự hạn chế của thuật toán là số lượng thời gian mà thuật toán yêu cầu, nhưng quan trọng hơn là việc gọi đệ quy lại các hàm làm tăng độ phức tạp của thuật toán. Trong tổng quan của vấn đề, đó là một trong những thách thức chung của khai thác đồ thị và hiện tại nó cũng là một hướng nghiên cứu quan trọng trong tương lai đối với công việc khai thác đồ thị. Nghiên cứu về vấn đề cải thiện hiệu quả của việc khai thác các MPB với các dữ liệu đầu vào lớn hơn, sẽ là một bước quan trọng trong hướng nghiên cứu phát triển tri thức hữu dụng thông qua MPB gần đúng.

Một phạm vi nghiên cứu khác có thể phát triển trong tương lai là tìm ra một cách để làm giảm bớt số lượng mẫu đồ thị tìm được trong khi vẫn giữ lại được những thông tin đạt được bằng việc sử dụng phương pháp so sánh gần đúng; sử dụng các hàm khác nhau để tính toán độ tương đồng của đồ thị; áp dụng một thuật toán đã công bố cho một vài trường hợp cụ thể giống như đồ thị động.

TÀI LIỆU THAM KHẢO

[1] S. Ranu, A. Singh, Graphsig: a scalable approach to mining significant subgraphs in large graph databases, in: IEEE 25th International Conference on Data Engineering, 2009, pp. 844–855.

[2] S. Nijssen, J.N. Kok, A quickstart in frequent structure mining can make a difference, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’04, ACM, 2004, pp. 647–652.

[3] X. Yan, J. Han, gspan: graph-based substructure pattern mining, in: Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM’02, 2002.

[4] A. Gago-Alonso, J. Medina-Pagola, J. Carrasco-Ochoa, J. Martínez-Trinidad, Mining frequent connected subgraphs reducing the number of candidates, in: W. Daelemans, B. Goethals, K. Morik (Eds.), Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science, vol. 5211, Springer, Berlin/Heidelberg, 2008, pp. 365–376.

[5] H. Cheng, X. Yan, J. Han, Mining graph patterns, in: C. Aggarwal, H. Wang (Eds.), Managing and Mining Graph Data, Advances in Database Systems, vol. 40, Springer, 2010, pp. 365–392.

[6] J. Huan, W. Wang, J. Prins, J. Yang, Spin: mining maximal frequent subgraphs from graph databases, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’04, ACM, 2004, pp. 581–586.

[7] J. Han, H. Cheng, D. Xin, X. Yan, Frequent pattern mining: current status and future directions, Data Min. Knowl. Discov. 15 (2007) 55–86.

[8] M. Al-Hasan, V. Chaoji, S. Salem, J. Besson, M.J. Zaki, Origami: mining representative orthogonal graph patterns, in: ICDM, IEEE Computer Society, 2007, pp.

153–162.

[9] W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. Yu, O. Verscheure, Direct mining of discriminative and essential frequent patterns via model-based search tree, in: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 230–238.

[10] F. Zhu, Q. Qu, D. Lo, X. Yan, J. Han, P.S. Yu, Mining top-k large structural patterns in a massive network, PVLDB 4 (2011) 807–818.

[11] Y. Jia, J. Zhang, J. Huan, An efficient graph-mining method for complicated and noisy data with real-world applications, Knowl. Inf. Syst. 28 (2011) 423–447.

[12] C. Chen, X. Yan, F. Zhu, J. Han, gApprox: mining frequent approximate patterns from a massive network, in: ICDM, IEEE Computer Society, 2007, pp. 445–450.

[13] A. Sanfeliu, K.S. Fu, A distance measure between attributed relational graphs for pattern recognition, IEEE Trans. Syst. Man Cybern. 13 (1983) 353–363.

[14] L.T. Thomas, S.R. Valluri, K. Karlapalem, Margin: maximal frequent subgraph mining, ACM Trans. Knowl. Discov. Data 4 (2010) 10:1–10:42.

[15] X. Chen, C. Zhang, F. Liu, J. Guo, Algorithm research of top-down mining maximal frequent subgraph based on tree structure, in: P. Snac, M. Ott, A.Seneviratne (Eds.), Wireless Communications and Applications, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 72, Springer, Berlin Heidelberg, 2012, pp. 401–411.

[16] M. Kuramochi, G. Karypis, Finding frequent patterns in a large sparse graph, Data Min. Knowl. Discov. 11 (2005) 243–271.

[17] B. Bringmann, S. Nijssen, What is frequent in a single graph?, in: T. Washio, E.Suzuki, K. Ting, A. Inokuchi (Eds.), Advances in Knowledge Discovery and Data

Mining, Lecture Notes in Computer Science, vol. 5012, Springer, Berlin/Heidelberg, 2008, pp. 858–863.

[18] M. Flores-Garrido, J.A. Carrasco-Ochoa, J.F. Martínez-Trinidad, Mining maximal frequent patterns in a single graph using inexact matching, Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonantzintla, Puebla, Mexico

[19] M. Kuramochi, G. Karypis, Grew – a scalable frequent subgraph discovery algorithm, in: Proceedings of the Fourth IEEE International Conference on Data Mining, 2004, pp. 439 – 442.

[20] M. Kuramochi, G. Karypis, Finding frequent patterns in a large sparse graph, Data Min. Knowl. Discov. 11 (2005) 243–271.

[21] Y. Xiao, H. Dong, W. Wu, M. Xiong, W. Wang, B. Shi, Structure-based graph distance measures of high degree of precision, Pattern Recognit. 41 (2008) 3547–3561.

[22] B. Bringmann, S. Nijssen, What is frequent in a single graph?, in: T. Washio, E. Suzuki, K. Ting, A. Inokuchi (Eds.), Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, vol. 5012, Springer, Berlin/ Heidelberg, 2008, pp. 858–863.