Trường Đại Học Bách Khoa Tp. Hồ Chí Minh
Khoa Khoa Học & Kỹ Thuật Máy Tính

CƠ SỞ TOÁN CHO KHOA HỌC MÁY TÍNH (055263)

Bài tập lớn

PHÁT HIỆN CẤU TRÚC CỘNG ĐỒNG TRONG ĐỒ THỊ

Giảng viên hướng dẫn:

Nguyen An Khuong, *CSE-HCMUT*
Nguyen Tien Thinh, *CSE-HCMUT*

Nhóm 1

Trần Thị Phượng - 2270234
Nguyễn Nguyên Ngọc - 2270212
Đặng Thành Lập - 2270247
Huỳnh Ý Vy - 2270129
Đinh Thanh Phong - 2270243

Phiên bản 1.0

Mục lục

Tìm Hiểu Một Số Thuật Toán Phát Hiện Cộng Đồng Trên Mạng Xã Hội Facebook

Tran Thi Phuong, Nguyen Nguyen Ngoc, Dang Thanh Lap, Dinh Thanh Phong, Huynh Y Vy
Faculty of Computer Science & Engineering,
Ho Chi Minh city University of Technology, Vietnam
268 Lý Thường Kiệt, Hồ Chí Minh, Viet Nam
dtlap.sdh221@hcmut.edu.vn

Ngày 22 tháng 8 năm 2024

Tóm tắt nội dung

Ngày nay, phát hiện cộng đồng trên một mạng xã hội đang là hướng nghiên cứu quan trọng trong lĩnh vực khoa học máy tính. Mạng xã hội thường được biểu diễn dưới dạng cấu trúc đồ thị. Chính vì vậy, phát hiện cộng đồng trên mạng xã hội chủ yếu gắn liền với bài toán phân cụm trên đồ thị. Để giải quyết bài toán, đã có rất nhiều thuật toán được quan tâm nghiên cứu. Trong bài báo này, nhóm tác giả sẽ trình bày các thuật toán cơ bản trong phát hiện cộng đồng trong mạng xã hội Facebook để từ đó đưa ra các quyết định như giới thiệu bạn bè hay quảng cáo hiệu quả.

Keywords: Community structure identification problem; Discover Social Circles in Ego Networks.

1 Giới thiệu

Các mạng xã hội trực tuyến cho phép người dùng theo dõi các bài đăng được tạo bởi hàng trăm bạn bè và người quen của họ. Bạn bè của người dùng tạo ra khối lượng thông tin khổng lồ và để đối phó với tình trạng 'quá tải thông tin', họ cần tổ chức các trangi cá nhân của mình. Một trong những cơ chế chính để người dùng các trang mạng xã hội tổ chức trang của họ và nội dung do họ tạo ra là phân loại bạn bè của họ thành những nhóm riêng. Trên thực tế, tất cả các mạng xã hội lớn đều cung cấp chức năng như vậy, chẳng hạn như 'vòng kết nối' trên Google và 'danh sách' trên Facebook và Twitter. Sau khi người dùng tạo vòng kết nối của mình, chúng có thể được sử dụng để lọc nội dung (ví dụ: để lọc các cập nhật trạng thái được đăng bởi những người quen ở xa), để bảo mật (ví dụ: để ẩn thông tin cá nhân với đồng nghiệp) và để chia sẻ các nhóm người dùng mà những người khác có thể muốn theo dõi

Hiện tại, người dùng trên Facebook, Google và Twitter xác định vòng kết nối của họ theo cách thủ công hoặc theo cách tự nhiên bằng cách xác định những người bạn có chung thuộc tính. Không có cách tiếp cận nào đặc biệt thỏa đáng: cách thứ nhất tốn thời gian và không cập nhật tự động khi người dùng thêm nhiều bạn bè hơn, trong khi cách thứ hai không nắm bắt được các khía cạnh riêng lẻ của cộng đồng người dùng và có thể hoạt động kém khi thông tin hồ sơ bị thiếu hoặc bị giữ lại.

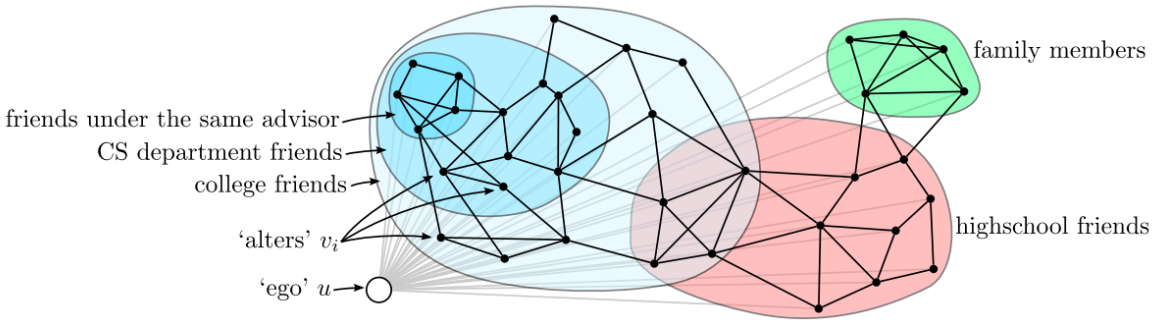
Trong phần báo cáo này, chúng tôi tìm hiểu một số thuật toán cơ bản để phát hiện cộng đồng trên tập dữ liệu mạng xã hội Facebook. Cụ thể, với một người dùng duy nhất có mạng xã hội cá nhân của người đó, mục tiêu của chúng tôi là xác định vòng kết nỗi của cô ấy, mỗi vòng kết nối là một tập hợp con bạn bè của cô ấy. Vòng kết nối dành riêng cho người dùng vì mỗi người dùng tổ chức mạng lưới bạn bè cá nhân của mình một cách độc lập với tất cả những người dùng khác mà cô ấy không kết nối. Điều này có nghĩa là chúng ta có thể hình thành bài toán phát hiện vòng kết nối như một bài toán phân cụm trên mạng lưới cá nhân của cô ấy, mạng lưới bạn bè giữa những người bạn của cô ấy.

Trong Hình 1, chúng ta có một người dùng duy nhất là u và chúng ta tạo một mạng lưới giữa những người bạn của cô ấy v_i . Chúng tôi coi người dùng u là nút cá nhân của người dùng và các nút v_i là những người khác

có liên quan. Sau đó, nhiệm vụ là xác định các vòng kết nối mà mỗi v_i thuộc về. Nói cách khác, mục tiêu là tìm các cộng đồng/cụm lồng nhau cũng như chồng chéo trong mạng cá nhân của người dùng. Dữ liệu về một bạn bè của ngưởng dùng có thể nằm trong nhiều nhóm khác nhau và các nhóm nhỏ có thể trùng lặp trong các nhóm khác nhau. Các thành viên trong vòng kết nối có chung thuộc tính. Để mô hình có thể có kết quả tốt , ta hi vọng các node là các bạn bè của người dùng nghiên cứu không chỉ thuộc về riêng một nhóm và cần thuộc nhiều nhóm khác nhau , cũng như việc các vòng kết nối càng chồng lên nhau hoặc giao nhau càng nhiều càng có lợi cho việc huấn luyện mô hình. Vì thế cần chỉ rõ các thuộc tính để phân loại nhóm người dùng cũng như kích thước cụm / nhóm của mỗi nhóm.

Mạng xã hội thường được mô hình hóa như là đồ thị và được gọi là đồ thị mạng xã hội. Các thực thể là các đỉnh (nút) và cạnh (cung), giữa hai đỉnh là mối liên kết giữa hai thực thể trên mạng.Một số đỉnh có liên kết chặt chẽ với nhau tạo thành từng cụm, và giữa các cụm đó được nối với nhau chỉ bằng một vài cạnh khác.Cộng đồng là một nhóm các thực thể có những tính chất tương tự nhau, liên kết chặt chẽ với nhau hơn và cùng đóng một vai trò nhất định trong mạng xã hội.Trong phạm vi bài tập lớn này, chúng tôi giới thiệu những khái niệm cơ bản về mô hình đồ thị mạng xã hội, các phương pháp tiếp cận nhằm phát hiện cấu trúc cộng đồng để từ đó đưa ra các quyết định như giới thiệu bạn bè hay quảng cáo hiệuq quả .

Phần còn lại của bài tập lớn được tổ chức như sau: Phần 2 giới thiệu những khái niệm cơ sở liên quan và cần thiết của lý thuyết đồ thị trong phát hiện cấu trúc cộng đồng. Phần 3, trình bày các thuật toán phát hiện cộng đồng được sử dụng bao gồm thuật toán Louvian , thuật toán truyền nhãn gốc (Label Propagation), thuật toán cộng đồng linh hoạt (Fluid Communities) và thuật toán Girvan-Newman. Phần 4 Biểu diễn sự thực nghệm các thuất toán đã đề ra và đánh giá giá trị thực thi trên tệp dữ liệu mẫu. Phần 5 đánh giá và kết luận lại phần nghiên cứu.



Hình 1: Network với các nhóm được gán nhãn

1.1 Dataset description

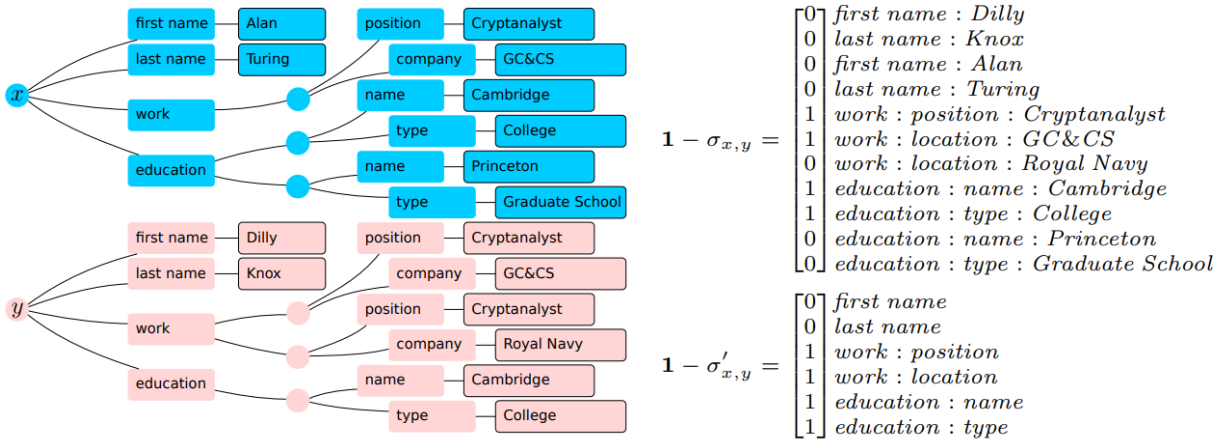
Dữ liệu được sử dụng trong bài nghiên cứu được tải xuống từ trang nghiên cứu của đại học Stanford trong đó đồ thị của dữ liệu được hình thành với mỗi đỉnh của đổ thị đại diện cho một người dùng đươc liệt kê trong dữ liệu và cạnh của đồ thị được hình thành khi hai người dùng đó có kết bạn trên Facebook. Dữ liệu Faceboook được thu thập dựa trên sự phân nhãn thủ công của 10 người dùng bao gồm 193 nhóm người dùng (vòng kết nối) và có tất cả 4039 bạn bè có liên quan của họ. Trung bình, mỗi người dùng cung cấp data với 19 nhóm và có kích thước 22 người bạn thuộc các nhóm như : bạn tiểu học, bạn đại học, người thân, bạn cùng nhóm chơi thể thao,...

Thông tin dữ liệu của từng node trong mạng của từng người dùng được cấu trúc theo dạng cây ở Hình 2. Ta giả sử có thông tin của hai người dùng x (màu xanh) và y (màu hồng) ở bên trái trong hình được xây dựng thành các véc-tơ đặc trưng có cấu trúc như bên phải hình. Ở tận cùng trên, bên phải hình ta dùng các chi số nhị phân thể hiện sự khác biệt giữa các nút lá trong cây. Những nút lá giống có thuộc tính giống nhau ở cả 2 người dùng được đánh số 1 và các thuộc tính không giống nhau cả x và y được đánh số 0. Dễ thấy x và y có chung work → position → Cryptanalyst ; work→location→GSCS; education → name → Cambridge ; education → type → College.

Ở dưới cùng của hình là tập hợp các thuộc tính có chung ở các nút lá trong sơ đồ đầu tiên với các thuộc

tính chung là nơi chốn làm việc và học tập mà loại bỏ đi cấp chỉ rõ tên các cơ quan nơi chốn.

Thông tin của từng node trong mạng được cấu trúc theo cấu trúc cây, với mỗi cấp của cây thì thông tin càng được cụ thể hóa.



Hình 2: Cấu trúc thuộc tính các node trong mạng

2 Một số vấn đề cơ sở

2.1 Đồ thị mạng xã hội

Là cấu trúc G=(V, E), trong đó V là tập các đỉnh và E là tập các cạnh, trong đó tập cạnh E thể hiện mối quan hệ xã hội giữa các thành viên với nhau.Đồ thị G có thể là vô hướng hoặc có hướng tùy theo mối quan hệ giữa các thành viên. Cấu trúc cộng đồng C được định nghĩa là tập con các đỉnh của V sao cho với mỗi đỉnh $v \in C$ có nhiều cạnh kết nối v với những đỉnh khác trong C và ít cạnh kết nối v với những đỉnh khác thuộc V.Ta ký hiệu δ_{st} là số đường đi ngắn nhất đi từ s tới t; $\delta_{st}(v)$ là số đường đi ngắn nhất đi từ s tới t và có đi qua v;
$\tau_{st}(v) = \frac{\delta_{st}(v)}{st}$ là độ phụ thuộc của cặp đỉnh s, t vào đỉnh v. Không làm mất tính tổng quát, giả thiết G là đồ thị vô hướng và áp dụng một số tính chất về đường đi trong đồ thị G, khi đó các cạnh và các đường đi luôn có tính đối xứng nghĩa là: $(u, v) \in E$ thì $(v, u) \in E$,$\delta_{st} = \delta_{ts}, \delta_{st}(v) = \delta_{vt}, \tau_{st}(v) = \tau_{ts}(v)$.

2.2 Độ đo trung gian đỉnh

Là độ đo của một đối tượng v trong mạng G được xác định bằng tổng số các đối tượng khác trong mạng có thể trao đổi và liên kết với nhau thông qua đối tượng đó trên G . Độ đo trung gian được xác định bằng công thức:

$$C_B(v) = \sum_{s \neq t \neq v} \tau_{st}(v) = \sum_{s \neq t \neq v} \frac{\delta_(st)}{\delta_{st}} \tag{1}$$

Trong đó, nếu $s = t$ thì $\delta_{s,t} = 1$, nếu $v = s$ hoặc $v = t$ thì $\delta_{s,t} = 0$, $\delta_{s,t} = \sum_{u \in N(t)\delta_{s,u}}$ trong đó $N(t) = u : (u, t) \in E; d(s, t) = d(s, u) + 1$;
$d(s, t)$ là khoảng cách giữa hai đỉnh s, t. Trong công thức 1, do tính chất đối xứng của các đường đi, ta chỉ cần tính một chiều từ s đến t. Giá trị $C_B(v)$ có ý nghĩa là hệ số tiềm năng để điều khiển sự liên kết giữa các đỉnh trên đồ thị. Nói cách khác, độ trung gian của đỉnh v là tổng các xác suất những đường đi ngắn nhất đi qua v trên tất cả các cặp đỉnh có thể của đồ thị G.

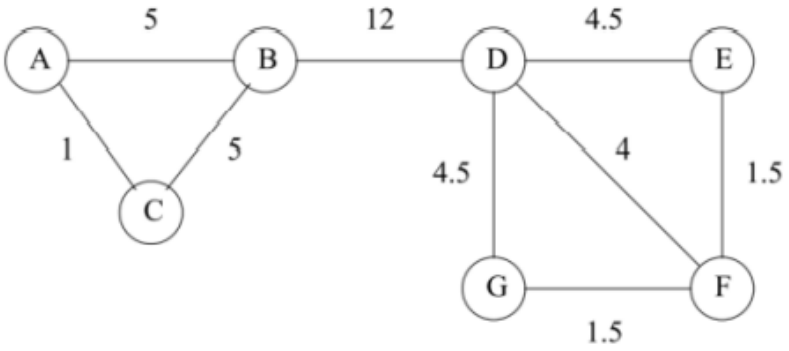
2.3 Độ trung gian của cạnh

$C_B(e)$ là độ đo trung gian của cạnh e = (a, b) trên đồ thị mạng G được định nghĩa bằng tỷ số các cặp đỉnh x và y mà cạnh e nằm trên đường đi ngắn nhất giữa x và y, với mọi $x, y \in V$. Độ đo trung gian được xác định bằng công thức:

$$C_B(e) = \sum_{s \neq t} \frac{\delta_{st}(e)}{\delta_{st}} \tag{2}$$

Do tính đối xứng nên độ đo trung gian của cạnh e theo công thức 3 sẽ bằng hai lần tổng của $\tau_{st}(e)$ tính theo một chiều với $s \in V$. Để tính độ đo trung gian của các đỉnh và các cạnh trên G, ta thường sử dụng phương pháp duyệt theo chiều rộng BFS bắt đầu từ đỉnh x bất kỳ, kết quả sẽ đưa ra một đồ thị định hướng phi chu trình DAGx (Directed Acyclic Graph) trong đó mỗi cạnh của DAGx sẽ là một phần của một đường đi ngắn nhất xuất phát từ x.

Chúng ta có thể phân cụm dựa vào độ trung gian của các cạnh bằng cách sắp xếp các cạnh của đồ thị theo thứ tự tăng dần của độ trung gian và tạo ra một đồ thị tương ứng. Mỗi bước, những thành phần liên thông của đồ thị sẽ tạo thành các cụm. Như vậy, độ trung gian lớn hơn, thì nhận được nhiều cạnh hơn và các cụm sẽ lớn hơn. Phương pháp này được thể hiện qua quá trình loại bỏ dần các cạnh có độ trung gian lớn hơn một ngưỡng nào đó. Bắt đầu từ đồ thị có tất cả các cạnh cho trước, loại bỏ đi những cạnh có độ trung gian lớn nhất, tiếp tục cho đến khi đồ thị được chia thành các thành phần liên thông theo yêu cầu. Trong hình 3 biểu diễn đô trung bình cạnh của một đồ thị.



Hình 3: Độ đo trung gian của đồ thị

2.4 Tiêu chuẩn đánh giá cộng đồng

Đại lượng Modularity và tối ưu hóa để đánh giá chất lượng cộng đồng:

$$Q = \frac{1}{2m} \sum_{ij} (a_{ij} - p_{ij} \delta(Ci, Cj) \tag{3}$$

Trong đó, $A = (a_{ij})_{nxn}$ là ma trận kề, $2m = \sum_{ij}, p_{ij}$l à số cạnh dự kiến có trong C, hàm $\delta(Ci, Cj) = 1$ nếu i, j thuộc cùng một cộng đồng và $\delta(Ci, Cj) = 0$ nếu ngược lại. Dựa trên xác suất kết nối giữa đỉnh i và đỉnh j ta có:

$$Q = \frac{1}{2m} \sum_{ij} (a_{ij} - \frac{k_i k_j}{2m}) \delta(Ci, Cj) \tag{4}$$

với ki, kj là bậc của đỉnh i và đỉnh j. Nếu kí hiệu n_c là số cộng đồng, l_c là số cạnh nối các đỉnh, d_c là tổng số bậc của các đỉnh của cộng đồng C. Khi đó ta có:

$$Q = \sum_{c=1}^{n_c} (\frac{l_c}{m} - (\frac{d-c}{2m})^2) \tag{5}$$

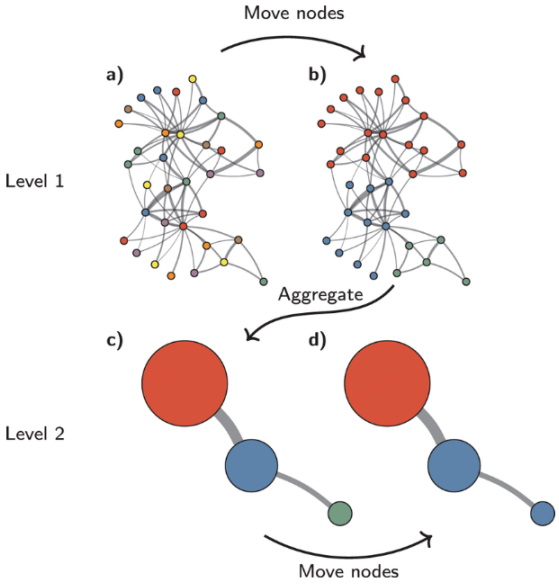
Một cộng đồng C có đại lượng Q đạt giá trị dương và càng lớn thì cộng đồng được xác định càng rõ, tức là việc phân cộng đồng là tốt.

3 Thuật toán

3.1 Louvain algorithm

Thuật toán Louvain do Blondel et al. đề xuất năm 2008 dùng để phát hiện cộng đồng cho những mạng kích thước lớn bằng cách tối ưu đại lượng Modularity (đã trình bày ở phần trên) cho mỗi cộng đồng.Thuật toán Louvain phân cụm theo cấp bậc, hợp nhất đệ quy các cộng đồng thành một nút duy nhất và thực hiện phân cụm theo mô đun trên các biểu đồ cô đọng.

Thuật toán gồm có 2 giai đoạn. Giai đoạn đầu tiên là quá trình tối ưu hóa Modularity (di chuyển cục bộ các nút) và giai đoạn thứ hai là quá trình tổng hợp cộng đồng (tổng hợp mạng). Được thể hiện như Hình 4



Hình 4: Hai giai đoạn của thuật toán Louvain.

Modularity của cộng đồng C được tính bằng công thức:

$$Q_c = \frac{\sum in}{2m} - (\frac{\sum tot}{2m})^2 \tag{6}$$

Trong đó :

$\sum in$ là tổng trọng số cạnh giữa các nút trong cộng đồng c (mỗi cạnh được xem xét hai lần);

$\sum tot$ là tổng của tất cả các trọng số cạnh cho các nút trong cộng đồng (bao gồm cả các cạnh liên kết với các cộng đồng khác).

Để tối đa hóa tính Modularity một cách hiệu quả, phương pháp Louvain có hai giai đoạn được lặp đi lặp lại nhiều lần.

Đầu tiên, mỗi nút trong mạng được gán cho cộng đồng của chính nó. Sau đó, đối với mỗi nút i, tính độ thay đổi của Modularity để loại bỏ i khỏi cộng đồng của chính nó và di chuyển nó vào cộng đồng của mỗi hàng xóm j của i. Giá trị này dễ dàng được tính bằng hai bước:

Bước 1: xóa i khỏi cộng đồng ban đầu của nó.

Bước 2: chèn i vào cộng đồng của j.

Phương trình cho bước 2 là:

$$\Delta Q = [\frac{\sum in + 2k_{i,in}}{2m} - (\frac{\sum tot + k_i}{2m})^2] - [\frac{\sum in}{2m} - (\frac{\sum tot}{2m})^2 - (\frac{k_i}{2m})^2] \tag{7}$$

Trong đó :

$\sum in$ là tổng của tất cả các trọng số của các liên kết bên trong cộng đồng mà i đang chuyển đến.

$\sum tot$ là tổng của tất cả các trọng số của các liên kết đến các nút trong cộng đồng i đang di chuyển vào

k_i là là mức độ trọng số của i

k_i, in là tổng trọng số của các liên kết giữa i và khác các nút trong cộng đồng mà i đang di chuyển vào.

m là tổng trọng số của tất cả các liên kết trong mạng.

Sau đó, khi giá trị này được tính cho tất cả các cộng đồng i được kết nối, i được đặt vào cộng đồng dẫn đến sự gia tăng Modularity lớn nhất. Nếu không thể tăng Modularity lên, i vẫn ở trong cộng đồng ban đầu của nó. Quá trình này được áp dụng lặp đi lặp lại và liên tục cho tất cả các nút cho đến khi không có sự gia tăng Modularity nào có thể xảy ra. Khi đạt được mức Modularity tối đa cục bộ này, giai đoạn đầu tiên đã kết thúc.

Trong giai đoạn thứ hai của thuật toán, nó nhóm tất cả các nút trong cùng một cộng đồng và xây dựng một mạng mới trong đó các nút là cộng đồng từ giai đoạn trước. Bất kỳ liên kết nào giữa các nút của cùng một cộng đồng hiện được biểu thị bằng các vòng lặp tự trên nút cộng đồng mới và các liên kết từ nhiều nút trong cùng một cộng đồng đến một nút trong cộng đồng khác được biểu thị bằng các cạnh có trọng số giữa các cộng đồng. Khi mạng mới được tạo, giai đoạn thứ hai đã kết thúc và giai đoạn đầu tiên có thể được áp dụng lại cho mạng mới.

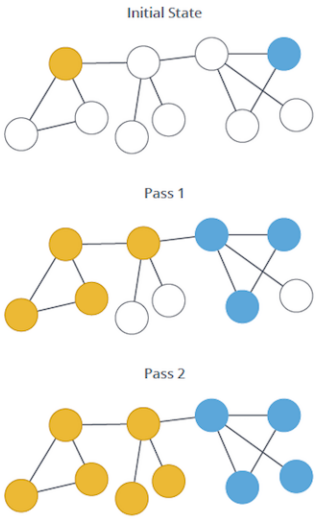
3.2 Label Propagation algorithm

Thuật toán gán nhãn (LPA) là một thuật toán được đề xuất bởi Raghavan và các cộng sự. Ý tưởng cốt lõi của LPA là gán từng nút trong mạng vào một cộng đồng, nơi mà hầu hết các nút láng giềng của nó sẽ quyết định nó có thuộc về cộng đồng mới hay không. Cụ thể, cho một mạng có n nút, trong đó nút x có nhãn ở lần lặp t được ký hiệu là Cx(t), LPA được mô tả như sau:

1. Khởi động nhãn ban đầu Cx (0) = x
2. Đặt t = 1
3. Lấy danh sách các nhãn được cập nhật cho các nút theo thứ tự ngẫu nhiên.
4. Đối với mỗi nút trong danh sách, xác định Cx(t) = f(Cx1(t), . . . , Cxm(t), Cxm + 1(t 1), . . . , Cxn(t 1)).

(Trong đó Cx1(t), . . . , Cxm (t) là nhãn của các nút lân cận đã được cập nhật trong lần lặp hiện tại, còn Cxm + 1(t - 1), . . . , Cxn(t - 1) là nhãn của các nút lân cận trong lần cập nhật trước. Hàm f trả về nhãn được chấp nhận bởi hầu hết các nút lân cận của nó.).

5. LPA đạt đến sự hội tụ nếu mọi nút có nút láng giềng là nhiều nhất có thể có), thì dừng thuật toán. Nếu không , đặt t = t + 1 và quay lại bước (3).
6. LPA dừng nếu đạt được số lần hội tụ hoặc số lần lặp tối đa do người dùng định nghĩa.



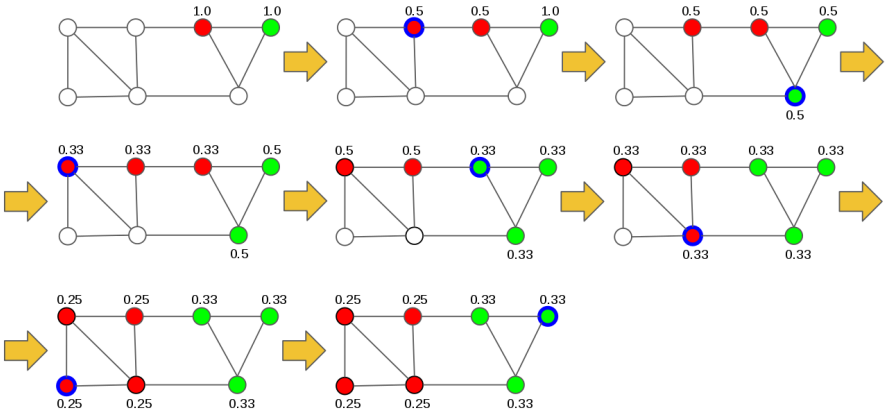
Hình 5: Ví dụ về thuật toán LPA

3.3 Fluid Communities algorithm

Thuật toán dựa trên ý tưởng đơn giản về chất lỏng tương tác trong một môi trường, mở rộng và đẩy lẫn nhau.

Lấy cảm hứng từ hiệu quả của Label Propagation Algorithm (LPA) và phương pháp lan truyền của nó, trong báo cáo này, nhóm đề xuất một thuật toán phát hiện cộng đồng mới gọi là thuật toán Cộng đồng chất lỏng (FluidC). Thuật toán này cố gắng bắt chước hành vi của một số chất lỏng (tức là các cộng đồng) đang mở rộng và đẩy lẫn nhau trong một môi trường khép kín, được chia sẻ (tức là đồ thị), cho đến khi tìm thấy trạng thái cân bằng. Một trong những tính năng phù hợp nhất của FluidC là nó có thể tìm thấy bất kỳ số lượng cộng đồng nào trong biểu đồ (ví dụ: người ta có thể chỉ định số lượng cộng đồng mà FluidC phải tìm) chỉ bằng cách khởi tạo số lượng chất lỏng đó. Theo hiểu biết tốt nhất của nhóm, FluidC là thuật toán dựa trên lan truyền đầu tiên có thuộc tính này. Đồng thời, FluidC tránh tạo ra các cộng đồng quái vật (một hạn chế nổi tiếng của LPA) theo cách tự nhiên, phi tham số.

Quy trình làm việc của thuật toán FluC cho k=2 cộng đồng (đỏ và xanh lục). Mỗi đỉnh được dán nhãn có sức mạnh liên quan của nó. Quy tắc cập nhật nhãn được đánh giá trên đỉnh được đánh dấu (màu xanh lam).

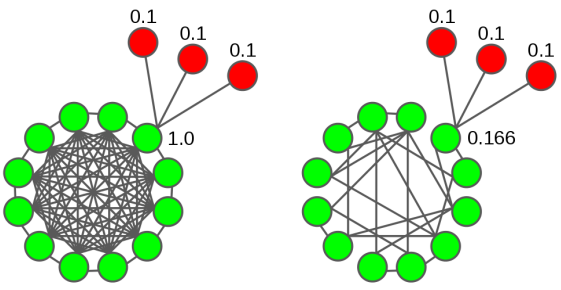


Hình 6: Quy trình làm việc của thuật toán FluC cho k=2

Một cộng đồng chất lỏng sẽ chinh phục các phần của môi trường có cấu trúc liên kết thuận lợi (nghĩa là được kết nối chặt chẽ với các đỉnh của nó) trong khi mất một số phần vào tay các cộng đồng chất lỏng khác. Đáng chú ý, khi một cộng đồng linh hoạt trải rộng qua nhiều đỉnh hơn thì mật độ của nó giảm đi, điều này làm giảm khả năng chinh phục và bảo vệ các đỉnh của nó.

Xét một đồ thị G=(V,E) được hình thành bởi một tập hợp các đỉnh V và một tập hợp các cạnh E. FluidC khởi tạo k cộng đồng chất lỏng trên k đỉnh khác nhau của V, các cộng đồng sẽ bắt đầu mở rộng trong toàn bộ biểu đồ. Tại mọi thời điểm, mỗi cộng đồng chất lỏng có tổng mật độ là 1,0. Khi một cộng đồng chất lỏng được nén thành một đỉnh duy nhất (ví dụ: khi khởi tạo), đỉnh đó chứa mật độ cộng đồng đầy đủ (tức là 1.0), đây cũng là mật độ tối đa mà một đỉnh có thể có. Khi một cộng đồng trải dài qua nhiều đỉnh, mật độ của nó sẽ được phân bổ đồng đều giữa các đỉnh tạo nên nó.

Hai trường hợp cập nhật luật trên đỉnh xanh. Một bên trái thuộc về một cộng đồng nội bộ thực sự tốt trong khi bên phải chứa ít hơn rất nhiều. Cộng đồng màu đỏ sẽ chỉ chinh phục đỉnh màu xanh lục trong ví dụ trên.



Hình 7: Hai trường hợp cập nhật luật trên đỉnh

Luồng công việc của FluidC tuân theo phương pháp lan truyền do LPA giới thiệu. Ở mỗi bước, FluidC lặp lại trên tất cả các đỉnh theo thứ tự ngẫu nhiên, cập nhật cộng đồng mà mỗi đỉnh thuộc về sử dụng quy tắc cập nhật. Nói một cách đơn giản, quy tắc cập nhật tính tổng mật độ của một cộng đồng lân cận đỉnh, bao gồm cả chính nó và trả về cộng đồng có mật độ tối đa. Nếu mật độ tối đa được chia sẻ bởi hai hoặc nhiều cộng đồng nhưng cộng đồng trước đó của đỉnh không nằm trong số đó, thì một cộng đồng ngẫu nhiên sẽ được chọn. Nếu cộng đồng trước đó của đỉnh nằm trong tập hợp các cộng đồng có mật độ tối đa, thì đỉnh đó sẽ giữ cộng đồng trước đó của nó. Lưu ý rằng điều này đảm bảo rằng sẽ không có cộng đồng nào bị loại khỏi biểu đồ, vì khi một cộng đồng được nén vào một đỉnh, cộng đồng này có mật độ tối đa có thể cho quy tắc cập nhật của đỉnh đó. FluidC ban đầu được thiết kế không đồng bộ. Một phiên bản đồng bộ trực tiếp của FluidC sẽ không đảm bảo rằng tất cả các cộng đồng luôn có tổng mật độ là 1.0. Nói cách khác, nó sẽ không đảm bảo rằng một cộng đồng luôn được tạo bởi ít nhất một đỉnh.

FluidC tránh các cộng đồng quái vật thông qua sự phân tán mật độ. Một cộng đồng lớn (khi so sánh với phần còn lại của cộng đồng trong biểu đồ) chỉ có thể giữ nguyên kích thước của nó bằng cách có nhiều cạnh trong cộng đồng tạo nên sự phân tán mật độ lớn hơn. Tính di động của các cộng đồng chất lỏng (chú ý đỉnh ban đầu của một cộng đồng có thể thuộc về một cộng đồng khác khi kết thúc thuật toán) cung cấp một hành vi mạnh mẽ trong bối cảnh khởi tạo ngẫu nhiên. Bất kể các đỉnh cộng đồng ban đầu được đặt ở đâu, các cộng đồng chất lỏng sẽ đẩy nhau về phía các vị trí phù hợp với cấu trúc liên kết biểu đồ.

FluidC cho phép xác định thông số kỹ thuật của số lượng cộng đồng được tìm thấy, đơn giản bằng cách khởi tạo một số lượng chất lỏng khác nhau trong môi trường. Đây là một thuộc tính mạnh mẽ và đáng mong đợi để phân tích dữ liệu, vì nó cho phép nghiên cứu biểu đồ và các thực thể của nó ở một số mức độ cụ thể. Mặc dù một số thuật toán phát hiện cộng đồng có tính năng này (ví dụ: Walktrap), nhưng không có thuật toán nào dựa trên phương pháp lan truyền hiệu quả.

```
0:  G = (V, E), num. communities k > 0, max_iterations > 0
1:  V_rand ⇐ V in random order
2:  for i ∈ {0..k} do
3:      v ⇐ V_rand[i]
4:      C_v ⇐ i
5:      D_v ⇐ 1.0
6:  end for
7:  converged ⇐ False
8:  while num_iterations < max_iterations and converged = False do
9:      converged ⇐ True
10:     for v ∈ V in random order do
11:         v_new_community ⇐ Community Update(G, v)
12:         if v_new_community ≠ C_v then
13:             v_old_community ⇐ C_v
14:             C_v ⇐ v_new_community
15:             Density Update(G, v_old_community)
16:             Density Update(G, v_new_community)
17:             converged ⇐ False
18:         end if
19:     end for
20: end while
```

Hình 8: Pseudo Code cho FluidC

3.4 Greedy Modularity Graph Clustering

Tìm các cộng đồng trong G bằng cách tối đa hóa mô đun theo giải thuật tham lam. Thuật toán này sử dụng tối đa hóa mô-đun tham lam của Clauset-Newman-Moore để tìm phân vùng cộng đồng có mô-đun lớn nhất. Đây là một thuật toán phát hiện cộng đồng dựa trên sự tham lam được đề xuất bởi Newman và cộng sự, (2004). Nó hoạt động dựa trên tối ưu hóa mô-đun và được biểu diễn trong một dendrogram là một chương trình phân cấp phân rã một mạng lưới thành các cộng đồng. Nó bắt đầu phát hiện các cộng đồng bằng cách gán đầu tiên mỗi đỉnh như một cộng đồng. Sau đó, nó liên tục tham gia hai cộng đồng có sự hợp nhất tạo ra

giá trị lớn nhất của giá trị modulity. Thuật toán dừng nối các đỉnh sau n-1 sao cho tham gia giả sử n là số đỉnh vì đây là giai đoạn mà thuật toán chỉ còn lại một cộng đồng duy nhất.

Greedy modularity maximization bắt đầu với mỗi nút trong cộng đồng của chính nó và liên tục tham gia vào cặp cộng đồng dẫn đến mô đun lớn nhất cho đến khi không thể tăng thêm mô đun (tối đa).

Hai đối số từ khóa điều chỉnh điều kiện dừng:

1. cutoff: là giới hạn dưới về số lượng cộng đồng để bạn có thể dừng quá trình trước khi đạt đến mức tối đa (được sử dụng để tiết kiệm thời gian tính toán).

2. best(n): là giới hạn trên đối với số lượng cộng đồng, do đó bạn có thể tiếp tục quy trình cho đến khi còn lại tối đa n cộng đồng ngay cả khi tính mô-đun tối đa xảy ra nhiều hơn. Để có được chính xác n cộng đồng, hãy đặt cả cutoff và best(n) thành n.

Các bước thực hiện thuật toán:

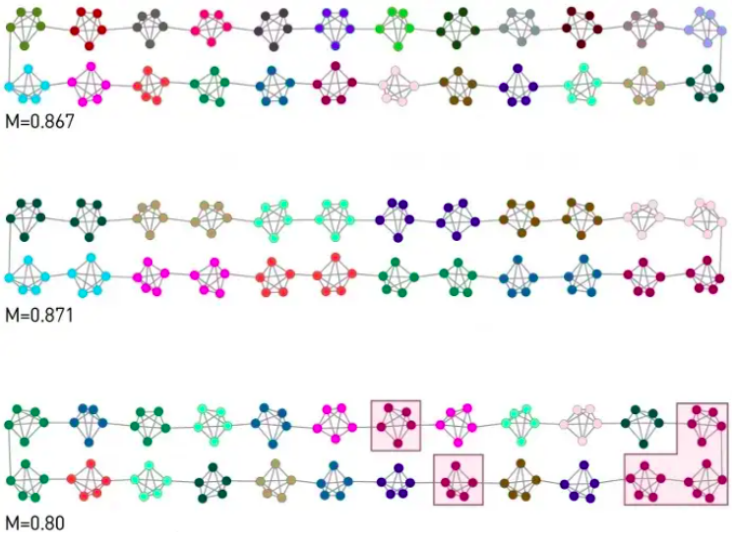
1. Chỉ định mỗi nút là một cộng đồng, tức là sẽ có nhiều cụm bằng số lượng nút.

2. Giảm số lượng cụm bằng cách thêm một cạnh và tính giá trị mô đun tức là khi thêm một cạnh, số lượng cụm sẽ được giảm thiểu từ 'n' xuống 'n-1'.

3. Xây dựng mạng mới được phân vùng bằng cách chọn phân vùng có modularity cao hơn.

4. Lặp lại quy trình từ 2 cho đến khi không còn nút chưa kết nối nào trong mạng (hoặc dừng theo điều kiện dừng).

Nếu độ phân giải của thuật toán nhỏ hơn 1, tính mô đun sẽ ưu tiên các cộng đồng lớn hơn. Nếu độ phân giải lớn hơn 1, ủng hộ các cộng đồng nhỏ hơn.



Hình 9: Ví dụ mô tả thuật toán Greedy Modularity

3.5 Girvan-Newman algorithm

Thuật toán Girvan-Newman là một trong những phương pháp phổ biến nhất trong bài toán phát hiện cộng đồng. Thuật toán phát hiện cộng đồng bằng cách lần lượt loại bỏ một số cạnh nhất định ra khỏi mạng gốc. Các thành phần liên thông còn lại được kết nối với nhau trong mạng gốc sẽ là các cộng đồng. Các cạnh được chọn ở đây dựa theo tính trung tâm cạnh (edge centrality), tập trung vào các cạnh mà có nhiều đường đi ngắn nhất giữa các nốt đi qua nhất.

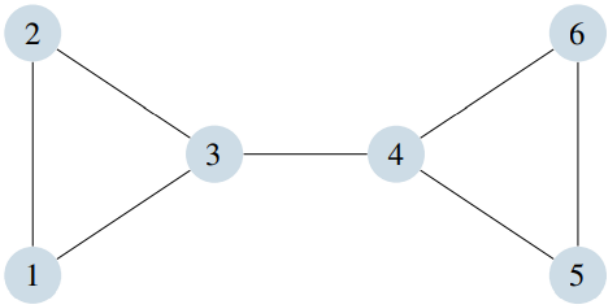
Ý tưởng của thuật toán Girvan-Newman như sau, trong một mạng, nếu các cộng đồng hoặc các nhóm được kết nối lỏng lẻo với nhau bởi một vài cạnh liên nhóm, thì tất cả các đường đi ngắn nhất giữa các cộng đồng khác nhau phải đi dọc theo một trong các cạnh này. Do đó, các cạnh kết nối cộng đồng sẽ có độ giữa của cạnh cao. Bằng cách loại bỏ các cạnh này, các cụm node sẽ được tách ra khỏi nhau và do đó cấu trúc cộng đồng cơ bản của mạng sẽ được hiện ra.

Thuật toán có thể được chia thành 4 bước chính:

1. Tính độ trung tâm cạnh (edge betweenness centrality) của từng cạnh trong mạng.
2. Loại bỏ đi cạnh mà có độ trung tâm cạnh cao nhất.

3. Tính lại độ trung tâm cạnh của từng cạnh còn lại.
4. Lặp lại bước 2 và 3 liên tục cho đến khi trong mạng tạo ra thành phần liên thông mới.

Ví dụ: Cho một đồ thị G = (V,E) với V = 1,2,3,4,5,6 và E = (1,2), (1,3), (2,3), (4,3),(4,5), (4,6), (5,6) được mô tả như Hình 10. Ta áp dụng thuật toán Girvan – Newman để phát hiện cộng đồng cho đồ thị G.



Hình 10: Ví dụ về một đồ thị đơn

Ta tìm đường đi ngắn nhất của tất cả các cặp đỉnh bằng giải thuật Dijkstra và thu được kết quả danh sách các đường đi ngắn nhất như hình 11 :

Cạnh	Đường đi ngắn nhất
(1,2)	$1 \rightarrow 2$
(1,3)	$1 \rightarrow 3$
(1,4)	$1 \rightarrow 3 \rightarrow 4$
(1,5)	$1 \rightarrow 3 \rightarrow 4 \rightarrow 5$
(1,6)	$1 \rightarrow 3 \rightarrow 4 \rightarrow 6$
(2,3)	$2 \rightarrow 3$
(2,4)	$2 \rightarrow 3 \rightarrow 4$
(2,5)	$2 \rightarrow 3 \rightarrow 4 \rightarrow 5$
(2,6)	$2 \rightarrow 3 \rightarrow 4 \rightarrow 6$
(3,4)	$3 \rightarrow 4$
(3,5)	$3 \rightarrow 4 \rightarrow 5$
(3,6)	$3 \rightarrow 4 \rightarrow 6$
(4,5)	$4 \rightarrow 5$
(4,6)	$4 \rightarrow 6$
(5,6)	$5 \rightarrow 6$

Hình 11: Danh sách đường đi ngắn nhất của các cặp cạnh của đồ thị

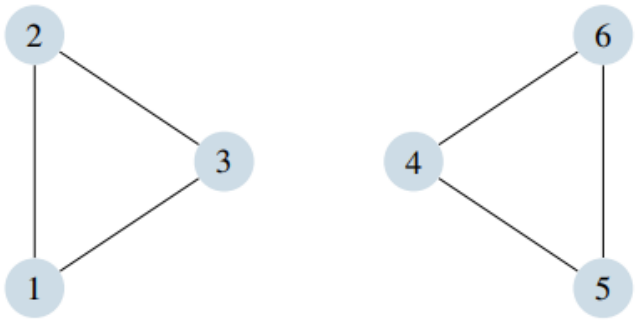
Ta tính độ giữa của cạnh của tất cả các cạnh bằng cách tính số lượng đường đi ngắn nhất đi qua nó. Kết quả thu được như hình 12:

Từ đó, ta chọn cạnh có độ giữa của cạnh cao nhất là cạnh (3,4) và xóa cạnh đó đi. Hình 13 thể hiện đồ thị sau khi đã xóa đi cạnh (3,4). Sau khi loại bỏ cạnh (3,4) trong đồ thị, sô thành phần liên thông tăng lên thành 2

Cạnh	Đường đi ngắn nhất
(1,2)	1
(1,3)	4
(2,3)	4
(3,4)	9
(4,5)	4
(4,6)	4
(5,6)	1

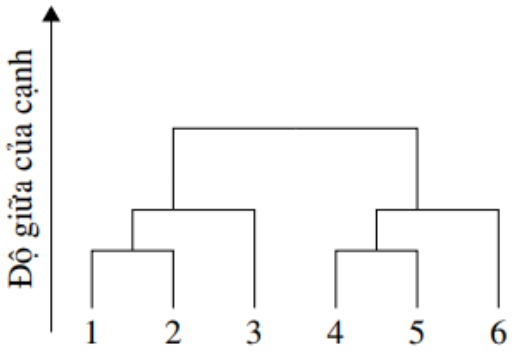
Hình 12: Giá trị độ giữa của cạnh của các cạnh

thành phần liên thông là C1 với các đỉnh là 1,2,3 và thành phần liên thông C2 với các đỉnh là 4,5,6. Lúc này, C1 C2 có thể được xem là các cộng đồng trong đồ thị G.



Hình 13: Đồ thị sau khi xóa cạnh (3,4)

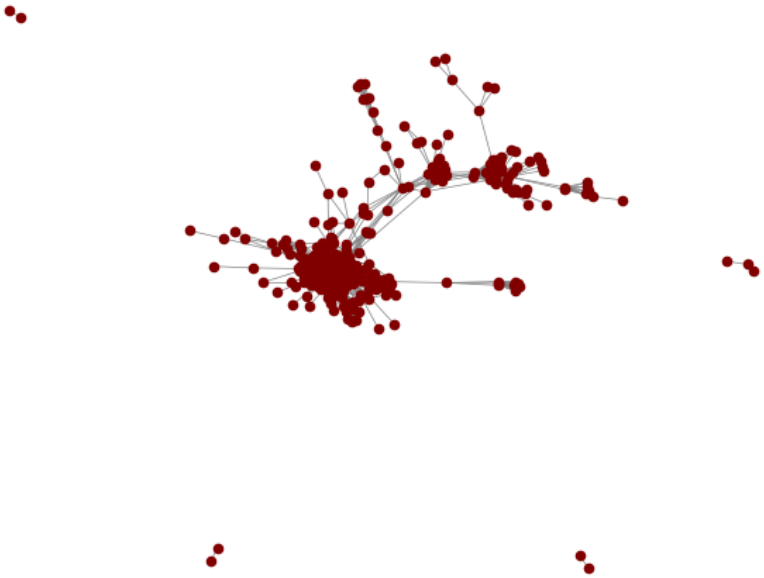
Lặp lại các bước làm trên cho đến khi mỗi đỉnh là một thành phần liên thông (tất cả các cạnh đã bị xóa) ta thu được dendrogram như hình14:



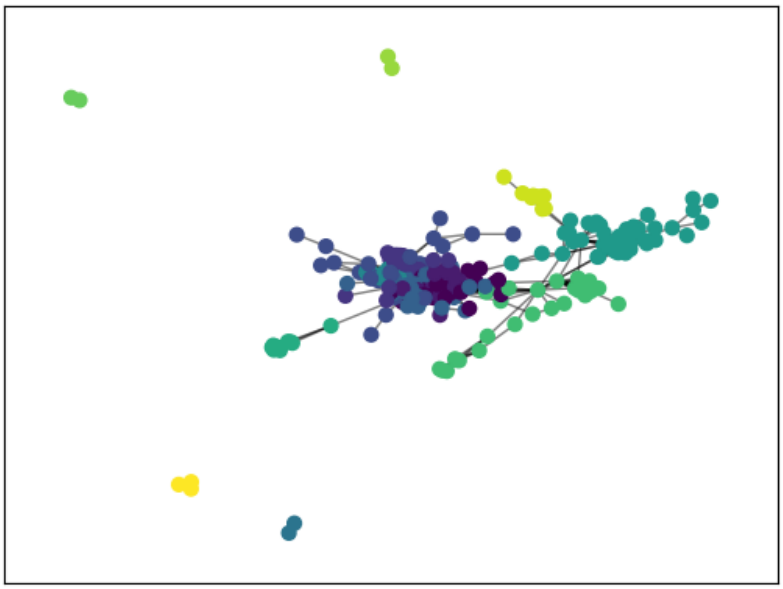
Hình 14: Dendrogram sau khi dùng giải thuật Girvan-Newman trên đồ thị.

4 Experiment

Nhóm thực nghiệm trên node 0 trong dataset ban đầu cho một số thuật toán mà nhóm đã tìm hiểu ở trên.



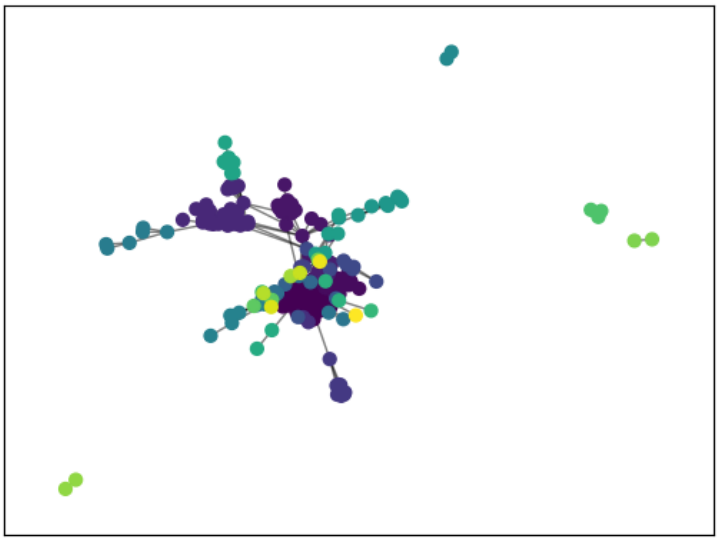
Hình 15: Biểu diễn cộng đồng ban đầu



```
# Number communities for Louvain alogorithm
number_communities(lou)
```

```
Number communities:  14
```

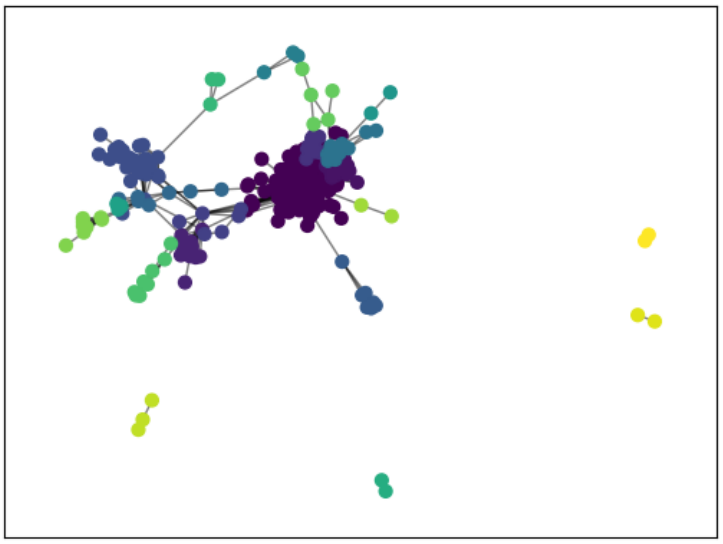
Hình 16: Cộng đồng được phát hiện bằng phương pháp Louvain



```
# Number communities for Girvan-Newman algorithm
number_communities(gn)
```

```
Number communities:  36
```

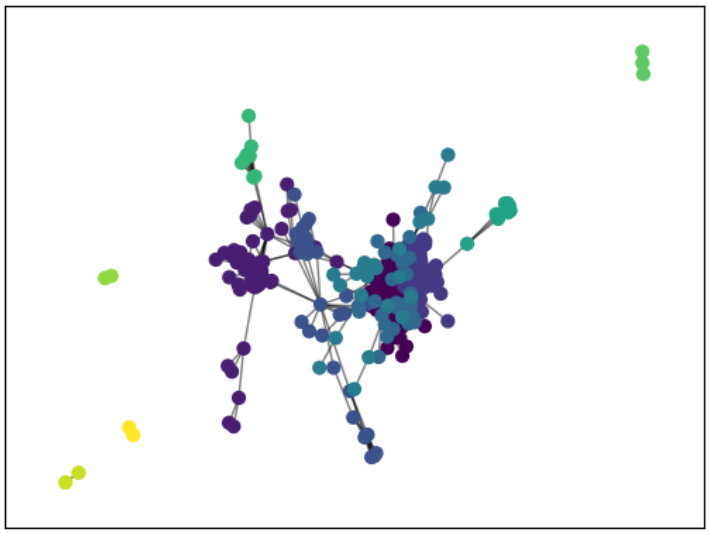
Hình 17: Cộng đồng được phát hiện bằng phương pháp Girvan-Newman



```
# Number communities for Label propagation algorithm
number_communities(lpa)
```

```
Number communities:  21
```

Hình 18: Cộng đồng được phát hiện bằng phương pháp Label propagation



```
# Number communities for Greedy modularity algorithm
number_communities(mod)
```

```
Number communities:  12
```

Hình 19: Cộng đồng được phát hiện bằng phương pháp Greedy Modularity

```
print("Modularity Louvian: ", modularity(lou,G_0))
print("Modularity LPA: ", modularity(lpa,G_0))
print("Modularity Girvan-Newman: ", modularity(gn,G_0))
print("Modularity Greedy algorihm: ", modularity(mod,G_0))
```

```
[64]  ✓ 0.7s
```

```
Modularity Louvian:  0.45865404663343823
Modularity LPA:  0.4014451660039514
Modularity Girvan-Newman:  0.41614614203983036
Modularity Greedy algorihm:  0.4458521745256101
```

Hình 20: Modularity của các thuật toán

5 Phương pháp đánh giá

Sau khi các cộng đồng được phát hiện bằng các thuật toán phát hiện cộng đồng, nhiệm vụ tiếp theo của chúng ta là đánh giá độ tốt của cấu trúc cộng đồng vừa phát hiện. Việc đánh giá sẽ dễ dàng nếu ta có được các cấu trúc cộng đồng thực tế hay thực địa. Trong chương này, chúng tôi sẽtrình bày một vài độ đo của việc phát hiện cộng đồng dựa trên nhãn thực địa. Các độ đo chính là Precision, Recall, F-mesure và BER.

5.1 Độ đo Precision, Recall và F-measure

Ta sử dụng các độ đo sau để đánh giá kết quả của phương pháp so với dữ liệu có nhãn thực địa. Cho đồ thị G= (V,E), gọi C=C1,C2,...,CK là K cụm được gom cụm bằng các giải thuật phát hiện cộng đồng, C'=C'1,C'2,...,C'L là cộng đồng thực địa. Ta có độ đo sau

Precision là độ đo thể hiện độ chính xác trên các tập dự đoán.

$$Precision = \frac{1}{K}\sum_{k=1}^{K} max\frac{|C_k, C'_j|}{|c_k|} \tag{8}$$

Recall là độ đo thể hiện độ nhạy của việc dự đoán.

$$Recall = \frac{1}{L}\sum_{l=1}^{L} max_j\frac{|C_{'l,C_j|}}{|C'_l|} \tag{9}$$

F-measure được tính dựa vào công thức sau:

$$F - measure = \frac{2.Precision.Recall}{Precision + Recall} \tag{10}$$

5.2 Độ đo BER

Độ đo BER là độ đo để đo tỷ lệ lỗi cân bằng của kết quả dữ đoán và nhãn của dữ liệu. Đối với bài toán phát hiện công đồng, độ đo BER được tính như sau. Cho đồ thị G= (V,E),gọi C=C1,C2,...,CK là K cụm được gom cụm bằng các giải thuật phát hiện cộng đồng, C'=C'1,C'2,...,C'L là L cộng đồng thực địa. Cho 2 cụm C và C', ta có độ đo BER(C.C') được tính bằng công thức:

$$BER(C.C') = \frac{1}{2}(\frac{|c'}{|c|} + \frac{|c'|}{|c'|}) \tag{11}$$

Ta sử dụng RBER(C.C') được tính bằng 1 BER(C.C')và tìm được các tập c thuộc C và c' thuộc C' dựa trên phép gán tuyến tính sử dụng thuật toán Hungarian để tìm giá trị tối đa của tổng các cặp theo giá trị RBER.

5.3 Kết quả đánh giá

Để thực hiện đánh giá kết quả các thuật toán, nhóm đề xuất sử dụng bộ dữ liệu (0.circle, 0.edges) từ tập dữ liệu Facebook Ego đã đề cập ở các mục trên, tiến hành đo trên máy tính có CPU Intel Core i5 Coffee Lake - 8300H, RAM 8GB và đánh giá các thuật toán dựa trên điểm NF1, dùng hàm tính NF1 của thư viện nf1 bao gồm: F1-min, F1-max, F1-mean, F1-mode, F1-std. Đầu vào là các tập tin (0.circle, 0.edges), lần lượt cho chạy các thuật toán Louvain, LPA, Girvan-Newman, Greedy modularity, ứng với mỗi thuật toán ta sẽ được kết quả các cộng đồng của từng nút, sau đó truyền tham số là kết quả đầu ra và tập tin 0.circle vào hàm tính NF1, có được kết quả như Bảng 1 của từng thuật toán.

6 Conclusion

Chúng tôi đã tìm hiểu một số thuật toán để phát hiện cộng đồng trên một mạng xã hội thực tế với dữ liệu đầu vào là dữ liệu từ mạng xã hội Facebook. Kết quả đạt được chưa được thật sự tốt vì nhóm triển khai trên đồ thị không có trọng số và không có hướng.

Thuật toán	F1-min	F1-max	F1-mean	F1-mode	F1-std
Louvain	0.03	0.5	0.265333	0.5	0.176592
LPA	0.03	0.61	0.296154	0.03	0.206976
Girvan-Newman	0.03	0.61	0.296154	0.03	0.206976
Greedy modularity	0.03	0.61	0.296154	0.03	0.206976

Bảng 1: Bảng đánh giá NF1 các thuật toán.

Phương pháp	Thời gian thực thi (s)
Greedy modularity	0.9
Louvain	0.8
Girvan-Newman	238
Label propagation	0.6

Hình 21: Thời gian chạy của các thuật toán

Trong tương lai nhóm sẽ cố gắng triển khai trên đồ thị có trọng số và định danh cụ thể cho từng cộng đồng để khai phá được nhiều thông tin ý nghĩa hơn từ tập dữ liệu mạng xã hội.

Tài liệu

[1] Dataset https://networkrepository.com/ego-facebook.php

[2] Thư viện NetworkX https://networkx.guide/algorithms/community-detection/

[3] Giải thuật Girven Newman https://networkx.guide/algorithms/community-detection/girvan-newman/

[4] https://www.arxiv-vanity.com/papers/1703.09307/

[5] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre 2008 J. Stat. Mech. Theory Exp.10 1–12

[6] https://web.stanford.edu/class/cs246/slides/11-graphs1.pdf

[7] Usha Nandini Raghavan, eka Albert and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks

[8] Nguyen Tan Duc. Community detection on social graphs, B. Eng Thesis, HCMUT, VNU-HCM(in vietnamese), 2018