ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Đồ án Tốt Nghiệp Đại Học

PHÂN TÍCH ẢNH HƯỞNG CỘNG ĐỒNG TRONG DỮ LIỆU MẠNG LƯỚI

KHOA TOÁN-TIN HỌC
NGÀNH KHOA HỌC DỮ LIỆU
KHÓA 2021

Sinh viên thực hiện	MSSV
Nguyễn Thị Lan Diệp	21280123
Huỳnh Lê Minh Thư	21280110
Nguyễn Thị Yến Như	21280082
Phạm Ngọc Phương Uyên	21280119
Trần Thị Bích Tuyền	21280059

Giảng viên hướng dẫn: TS. Tô Đức Khánh

Tp. Hồ Chí Minh, xx-xx, 2018

Lời cảm ơn

Trong suốt quá trình thực hiện đồ án tốt nghiệp này, nhóm chúng em đã nhận được sự giúp đỡ quý báu từ nhiều cá nhân và tổ chức. Trước hết, nhóm chúng em xin bày tỏ lòng biết ơn sâu sắc đến TS. Tô Đức Khánh, người đã tận tâm hướng dẫn, góp ý chi tiết và truyền cảm hứng để nhóm chúng em vượt qua những khó khăn trong quá trình nghiên cứu. Sự kiên nhẫn, sự nghiêm khắc trong khoa học và những lời khuyên quý giá của Thầy/Cô đã giúp nhóm chúng em hoàn thiện đề tài một cách tốt nhất.

Nhóm chúng em cũng xin chân thành cảm ơn các thầy cô trong khoa Toán - Tin học, trường Đại học Khoa học Tự Nhiên, ĐHQG-HCM, đặc biệt là các giảng viên đã giảng dạy, truyền đạt kiến thức nền tảng trong suốt thời gian nhóm chúng em theo học. Những bài giảng, những buổi seminar và sự hỗ trợ từ quý thầy cô đã giúp nhóm chúng em có đủ hành trang để thực hiện công trình này.

Về phía gia đình, nhóm chúng em xin gửi lời cảm ơn sâu sắc đến bố mẹ, anh chị em và người thân – những người luôn bên cạnh động viên nhóm chúng em cả về tinh thần lẫn vật chất. Dù không trực tiếp tham gia vào quá trình nghiên cứu, nhưng sự hy sinh và niềm tin của gia đình chính là động lực lớn nhất giúp nhóm chúng em kiên trì đến phút cuối cùng.

Cuối cùng, nhóm chúng em xin gửi lời tri ân đến những người bạn đồng hành, những anh chị khóa trên đã chia sẻ kinh nghiệm, tài liệu và luôn sẵn sàng lắng nghe những khó khăn của nhóm chúng em. Cảm ơn tất cả vì đã cùng nhóm chúng em trải qua hành trình đầy thử thách nhưng cũng vô cùng ý nghĩa này.

Dù đã cố gắng hết sức, nhưng do hạn chế về thời gian và kinh nghiệm, đồ án tốt nghiệp không tránh khỏi những thiếu sót. nhóm chúng em rất mong nhận được sự góp ý từ quý thầy cô và độc giả để công trình được hoàn thiện hơn trong tương lai.

Các thành viên nhóm

Tóm tắt nội dung

Trong bối cảnh dữ liệu lớn và sự phát triển mạnh mẽ của các nền tảng số, việc phân tích mức độ ảnh hưởng của các thực thể trong mạng lưới trở nên ngày càng quan trọng, đặc biệt trong các lĩnh vực như tiếp thị, tài chính - kinh tế và truyền

thông - giải trí,... Đề tài *Phân Tích Ảnh Hưởng Cộng Đồng trong Dữ Liệu Mạng (Community Influence Analysis for Network Data)* tập trung vào việc xây dựng mô hình đánh giá ảnh hưởng trong các mạng lưới phức tạp, với ứng dụng cụ thể trên hai lĩnh vực: dữ liệu âm nhạc từ nền tảng Spotify và dữ liệu cổ phiếu từ thị trường chứng khoán.

Nghiên cứu sử dụng các kỹ thuật phân tích đồ thị để khám phá và hiểu rõ cấu trúc của mạng lưới. Cụ thể, các chỉ số đo lường trung tâm được áp dụng nhằm xác định những thực thể (như nghệ sĩ hoặc cổ phiếu) có vai trò nổi bật trong hệ thống. Thêm vào đó, các thuật toán phát hiện cộng đồng được triển khai để làm rõ mối liên kết giữa các nhóm tương đồng, trong khi mô hình lan truyền ảnh hưởng được sử dụng để phân tích cách thức thông tin, xu hướng hoặc mức độ phổ biến lan rộng trong mạng.

Ngoài phần lý thuyết, đề tài còn phát triển một ứng dụng trực quan nhằm hỗ trợ quá trình ra quyết định dựa trên mức độ ảnh hưởng của từng thực thể. Ứng dụng này có thể được triển khai trong các tình huống thực tế, chẳng hạn như việc đề xuất nghệ sĩ trên nền tảng nhạc số, phân tích xu hướng thị trường hoặc đánh giá ảnh hưởng tương quan giữa các mã cổ phiếu.

Với cách tiếp cận liên ngành, kết hợp giữa phân tích mạng xã hội và dữ liệu cổ phiếu – âm nhạc, đề tài không chỉ mang lại góc nhìn mới về ảnh hưởng trong mạng lưới mà còn đóng góp giá trị thực tiễn cho các hệ thống hỗ trợ phân tích và ra quyết định.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh sách ký hiệu

ML	Maximum likelihood
	Hợp lí cực đại
REML	Restricted Maximum Likelihood
	Hợp lí cực đại giới hạn
FS	Fisher scoring
NR	Newton-Raphson

Chương 1

GIỚI THIỆU

1.1 Lý do chọn đề tài

Trong bối cảnh phát triển nhanh chóng của các mạng xã hội và hệ thống kết nối trực tuyến, việc phân tích ảnh hưởng của các cá nhân hoặc tổ chức trong mạng đã trở thành một vấn đề quan trọng. Việc xác định các nút có ảnh hưởng trong mạng xã hội không chỉ có ý nghĩa trong lĩnh vực tiếp thị số, mà còn trong các lĩnh vực như kinh tế, y tế công cộng và khoa học chính trị. Tuy nhiên, hầu hết các phương pháp đo lường ảnh hưởng hiện tại chỉ dựa trên thông tin về cấu trúc mạng mà không tính đến các thuộc tính cụ thể của từng nút, dẫn đến sự thiếu chính xác trong đánh giá mức độ ảnh hưởng thực sự của các cá nhân. Do đó, đề tài *"Phân tích ảnh hưởng của cộng đồng trong dữ liệu mạng"* nhằm xây dựng một mô hình mới để xác định mức độ ảnh hưởng không đồng nhất của các nút trong mạng xã hội, giúp nâng cao độ chính xác trong việc phát hiện các cá nhân có tầm ảnh hưởng lớn trong một cộng đồng.

1.2 Mục tiêu nghiên cứu

Thông qua việc hiểu rõ vai trò và tầm ảnh hưởng của từng thực thể, nhà nghiên cứu và nhà thực hành có thể khai thác sâu hơn các xu hướng tiềm ẩn, từ đó hỗ trợ quá trình ra quyết định một cách hiệu quả trong các lĩnh vực như tiếp thị số, đầu tư tài chính và hệ thống gợi ý cá nhân hóa.

Xuất phát từ thực tiễn đó, đề tài này được triển khai với mục tiêu phân tích, đánh giá vai trò và sức ảnh hưởng của ác thực thể hoặc tác nhân có trong một mạng lưới đa cộng đồng. Bên cạnh đó, đề tài còn tập trung ứng dụng các mô hình tiên tiến trong lĩnh vực khai phá mạng lưới như phát hiện cộng đồng và đo lường ảnh hưởng, qua đó góp phần làm sáng tỏ cấu trúc và động lực bên trong của các hệ

thống dữ liệu phức tạp.

1.3 Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài bao gồm hai lĩnh vực chính: âm nhạc và tài chính. Trong lĩnh vực âm nhạc, dữ liệu được thu thập từ nền tảng Spotify, bao gồm thông tin chi tiết về nghệ sĩ, danh sách phát, thể loại nhạc và xu hướng nghe nhạc của người dùng. Những dữ liệu này phục vụ cho việc phân tích cấu trúc mạng âm nhạc cũng như xác định mức độ ảnh hưởng của từng thực thể trong hệ sinh thái nội dung số.

Ở lĩnh vực tài chính, nghiên cứu tập trung vào việc thu thập và phân tích dữ liệu từ thị trường chứng khoán, với mục tiêu xác định mức độ ảnh hưởng tương hỗ giữa các cổ phiếu trong mạng lưới tài chính. Việc này nhằm làm rõ vai trò của từng cổ phiếu trong quá trình dẫn dắt thị trường và hỗ trợ quá trình xây dựng chiến lược đầu tư.

Bên cạnh đó, nghiên cứu áp dụng các mô hình phân tích mạng xã hội và các phương pháp phát hiện cộng đồng để khám phá các đặc điểm cấu trúc trong mạng lưới, từ đó phục vụ cho việc đánh giá mức độ ảnh hưởng và xác định các cụm thực thể có mối liên kết chặt chẽ.

1.4 Phương pháp nghiên cứu

Nghiên cứu được thực hiện thông qua việc thu thập dữ liệu từ API của Spotify và các nguồn tài chính uy tín, bao gồm thông tin về nghệ sĩ, thể loại âm nhạc, danh sách phát, cùng với dữ liệu giao dịch cổ phiếu. Để khám phá cấu trúc mạng lưới, các thuật toán phát hiện cộng đồng tiên tiến như mô hình khối ngẫu nhiên (Stochastic Block Model - SBM), Girvan-Newman, Louvain và Leiden được áp dụng. Các thuật toán này giúp phân tích và xác định các nhóm thực thể có mối liên kết chặt chẽ trong mạng.

Việc đánh giá mức độ ảnh hưởng của các thực thể trong mạng lưới được thực hiện thông qua mô hình phân tích ảnh hưởng cộng đồng (Community Influence Model - CIM). Cụ thể, chỉ số λ (community influence) được tính toán từ mô hình CIM, đóng vai trò là thước đo chính để xác định tầm ảnh hưởng của từng thực thể trong mạng. Chỉ số này phản ánh mức độ tác động của các thực thể dựa trên cấu trúc và động lực học của mạng lưới.

Để đảm bảo độ tin cậy trong phân tích, mô hình được kiểm định bằng các tiêu chí

thống kê và phương pháp đánh giá hiệu suất. Đồng thời, phương pháp tối ưu hóa ADMM (Alternating Direction Method of Multipliers) được sử dụng để tinh chỉnh các tham số trong mô hình CIM, đảm bảo kết quả phân tích chính xác và đáng tin cậy. Kết quả nghiên cứu mang lại giá trị thực tiễn cao, hỗ trợ ra quyết định trong các lĩnh vực như tiếp thị số, phân tích xu hướng truyền thông và đầu tư tài chính.

Chương 2

TỔNG QUAN VỀ DỮ LIỆU MẠNG LƯỚI VÀ ẢNH HƯỞNG CỘNG ĐỒNG

2.1 Khái niệm về dữ liệu mạng lưới

2.1.1 Định nghĩa và đặc điểm

Dữ liệu mạng lưới (network data) là loại dữ liệu thể hiện mối quan hệ hoặc sự kết nối giữa các thực thể khác nhau. Không giống như dữ liệu dạng bảng truyền thống, dữ liệu mạng lưới tập trung vào cách các đối tượng liên kết với nhau. Chẳng hạn như giữa người với người, địa điểm với địa điểm, hoặc tổ chức với tổ chức. Loại dữ liệu này đặc biệt hữu ích khi phân tích các tương tác trong không gian, vì nó giúp khám phá các mô hình kết nối và dòng chảy giữa các thực thể trong một bối cảnh cụ thể.

2.1.2 Cách biểu diễn dữ liệu mạng

Dữ liệu mạng lưới thường được trực quan hóa dưới dạng đồ thị, trong đó:

- Các nút (nodes) đại diện cho các thực thể như người, địa điểm hoặc đối tượng.
- Các cạnh (edges) thể hiện mối quan hệ hoặc sự tương tác giữa các nút.

Hình thức biểu diễn này giúp người phân tích dễ dàng nhận diện các mẫu hoặc xu hướng trong cách các thực thể liên kết với nhau.

Ví dụ, trong một mạng xã hội, mỗi người dùng là một nút, và nếu hai người là bạn bè, sẽ có một cạnh nối giữa họ. Từ đó, ta có thể phân tích ảnh hưởng xã hội: nếu người dùng B chia sẻ một nội dung, nội dung đó có thể nhanh chóng lan sang

người dùng A và C nếu họ có kết nối trực tiếp với B. Những phân tích kiểu này giúp hiểu rõ hơn về cách thông tin lan truyền, các nhóm tương tác mật thiết, hoặc vai trò trung tâm của từng cá nhân trong cộng đồng.

2.1.3 Ứng dụng của dữ liệu mạng

Dữ liệu mạng được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau nhờ khả năng mô hình hóa các mối quan hệ và dòng tương tác trong hệ thống. Trong quy hoạch đô thị, dữ liệu mạng cho phép mô hình hóa các tuyến giao thông và dòng người di chuyển, từ đó hỗ trợ tối ưu hóa hạ tầng và hệ thống vận tải công cộng. Trong lĩnh vực tiếp thị và quảng cáo, việc phân tích mạng lưới giúp xác định các cá nhân có sức ảnh hưởng trong cộng đồng, góp phần xây dựng các chiến dịch truyền thông lan tỏa hiệu quả hơn. Ngoài ra, trong y tế cộng đồng, dữ liệu mạng được sử dụng để theo dõi đường lây truyền của dịch bệnh và phát hiện các cụm dân cư có nguy cơ cao, hỗ trợ việc can thiệp và kiểm soát dịch kịp thời.

2.2 Khái niệm về phân tích ảnh hưởng cộng đồng

2.2.1 Định nghĩa và tầm quan trọng

Phân tích ảnh hưởng cộng đồng là phương pháp nhằm đo lường mức độ tác động của các cá nhân hoặc nhóm trong việc định hình hành vi, quan điểm và quyết định trong cộng đồng. Trong phân tích mạng xã hội, phương pháp này giúp xác định các thực thể có ảnh hưởng lớn và làm rõ cơ chế lan truyền ảnh hưởng giữa các nhóm và thành viên trong mạng lưới. Trong bối cảnh dữ liệu mạng ngày càng phổ biến từ các nền tảng trực tuyến như Facebook, Twitter hay LinkedIn, phân tích ảnh hưởng cộng đồng trở thành một hướng nghiên cứu quan trọng nhằm khám phá cấu trúc và động lực của các mối quan hệ trong hệ thống kết nối.

2.2.2 Ứng dụng

Phân tích ảnh hưởng cộng đồng là một hướng nghiên cứu có tính ứng dụng cao, đặc biệt trong các hệ thống mà sự tương tác giữa các thực thể đóng vai trò quan trọng trong việc định hình xu hướng chung. Trong phạm vi đề tài, hai lĩnh vực tiêu biểu được xem xét là thị trường chứng khoán và nền tảng âm nhạc trực tuyến.

Trong lĩnh vực chứng khoán, phân tích ảnh hưởng cộng đồng nhằm khám phá mối quan hệ tác động lẫn nhau giữa các cổ phiếu hoặc nhóm cổ phiếu trong một thị trường tài chính. Các thực thể trong mạng có thể là mã cổ phiếu, công ty hoặc ngành nghề, được kết nối với nhau thông qua các đặc điểm như mức độ đồng biến động giá (price co-movement), mối liên hệ ngành nghề hoặc hành vi đầu tư đồng

thời. Việc xác định các cổ phiếu có ảnh hưởng trung tâm trong mạng lưới có thể hỗ trợ xây dựng chiến lược đầu tư, phát hiện rủi ro hệ thống, hoặc hiểu rõ hơn về sự lan truyền của các cú sốc thị trường trong toàn bộ hệ thống tài chính.

Trong lĩnh vực âm nhạc, đặc biệt là trên các nền tảng nghe nhạc trực tuyến như Spotify, phân tích ảnh hưởng cộng đồng tập trung vào việc đánh giá mối quan hệ giữa các nghệ sĩ, bài hát hoặc danh sách phát (playlist). Các thực thể này được kết nối dựa trên hành vi nghe nhạc của người dùng, sự xuất hiện đồng thời trong playlist, hoặc sự tương đồng về thể loại. Mục tiêu là xác định các thực thể âm nhạc có ảnh hưởng lớn đến thị hiếu người nghe, qua đó hỗ trợ đề xuất nội dung, cải thiện thuật toán gợi ý, hoặc tối ưu hóa chiến lược phân phối nội dung. Đồng thời, việc phân tích dòng ảnh hưởng trong cộng đồng người nghe cũng giúp hiểu rõ hơn cách xu hướng âm nhạc lan truyền và hình thành trong cộng đồng.

Tổng thể, việc ứng dụng phân tích ảnh hưởng cộng đồng vào hai lĩnh vực trên không chỉ giúp làm rõ cơ chế lan truyền thông tin hoặc xu hướng mà còn mang lại giá trị thực tiễn cao trong việc hỗ trợ ra quyết định, cả về mặt tài chính lẫn truyền thông nội dung.

Chương 3

CÁC PHƯƠNG PHÁP PHÁT HIỆN CỘNG ĐỒNG

3.1 Các khái niệm cơ bản trong phát hiện cộng đồng

3.1.1 Các khái niệm cơ bản

3.1.1.1 Thuật ngữ và ký hiệu

Trong phần này, chúng tôi tập trung trình bày thuật toán *Stochastic Block Model (SBM)* áp dụng cho *đồ thị vô hướng*.

Xét một đồ thị $G = (N, E)$, trong đó N là tập các đỉnh với số lượng $n := |N|$, và E là tập các cạnh với số lượng $M := |E|$. Ví dụ, trong Hình 3.1, ta có $N = \{1, 2, \ldots, 90\}$, tức $n = 90$, và $M = 1192$. Một cặp đỉnh (p, q) được gọi là một *dyad*, và sự tồn tại hay không tồn tại của cạnh giữa hai đỉnh này được biểu diễn bằng một *ma trận kề* kích thước $n \times n$, ký hiệu là Y.

Vì G là đồ thị vô hướng nên ma trận Y có tính chất đối xứng: $Y_{pq} = Y_{qp} = 1$ nếu tồn tại cạnh nối giữa p và q, và $Y_{pq} = Y_{qp} = 0$ nếu không tồn tại cạnh nào nối hai đỉnh này. Ký hiệu M_{rs} được dùng để chỉ phần tử ở hàng r, cột s của một ma trận M bất kỳ. Do đó, với đồ thị vô hướng, ma trận Y luôn đối xứng qua đường chéo chính.

Trong mô hình SBM, mỗi đỉnh thuộc về một trong K nhóm (với $K < n$). Trong ví dụ minh họa này, ta giả sử $K = 3$. Vì thông tin phân nhóm chưa được biết trước khi mô hình hóa, ta định nghĩa với mỗi đỉnh $p = 1, 2, \ldots, n$ một vector nhị phân K chiều Z_p, trong đó chỉ có đúng một phần tử có giá trị bằng 1 (biểu thị nhóm mà đỉnh p thuộc về), còn lại là 0. Ví dụ, nếu các đỉnh 1, 45 và 90 thuộc về các nhóm 1, 2 và 3 tương ứng thì:

$$Z_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad Z_{45} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad Z_{90} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Toàn bộ các vector Z_p được ghép thành một ma trận $n \times K$:

$$Z := \begin{bmatrix} Z_1 & Z_2 & \dots & Z_n \end{bmatrix}^T,$$

trong đó Z_{pi} là phần tử thứ i của vector Z_p. Kích thước của từng nhóm có thể suy ra từ Z và được biểu diễn bằng vector:

$$N = \begin{bmatrix} N_1 \\ N_2 \\ \dots \\ N_K \end{bmatrix}.$$

Về cơ bản, N_i là tổng số đỉnh thuộc nhóm i, tương đương với tổng các phần tử khác 0 trong cột thứ i của Z. Trong ví dụ này, ta giả sử:

$$N_1 = 25, \quad N_2 = 30, \quad N_3 = 35.$$

3.1.1.2 Ma trận cạnh và xác suất kết nối

Dựa trên ma trận gán nhóm Z và ma trận kề Y, ta có thể xây dựng một ma trận cạnh kích thước $K \times K$, ký hiệu là E. Phần tử E_{ij} biểu thị số lượng cạnh giữa các đỉnh thuộc nhóm i và nhóm j trong đồ thị. Trong trường hợp đồ thị vô hướng, ma trận E là đối xứng, tức $E_{ij} = E_{ji}$. Ví dụ, với đồ thị trong Hình 3.2 được sử dụng để phân tích, ta có:

$$E_{11} = 245, \quad E_{22} = 341, \quad E_{33} = 481,$$

$$E_{12} = E_{21} = 37, \quad E_{23} = E_{32} = 52, \quad E_{31} = E_{13} = 36.$$

Tiếp theo, ta định nghĩa ma trận xác suất kết nối giữa các nhóm, gọi là ma trận khối $C \in [0,1]^{K \times K}$. Với đồ thị vô hướng, C cũng là một ma trận đối xứng, trong đó phần tử C_{ij} đại diện cho xác suất tồn tại một cạnh giữa hai đỉnh bất kỳ thuộc nhóm i và nhóm j. Nói cách khác, C phản ánh mật độ kết nối trung bình giữa các nhóm trong mô hình.

Giả định cơ bản trong SBM là các dyad độc lập có điều kiện, nghĩa là khi đã biết thông tin phân nhóm trong Z, thì sự tồn tại của một cạnh giữa hai đỉnh p và q

là ngẫu nhiên và không phụ thuộc vào các dyad khác. Cụ thể, biến Y_{pq} – biểu thị sự có mặt của một cạnh giữa p và q – tuân theo phân phối Bernoulli với xác suất thành công được xác định bởi tích vô hướng:

$$Y_{pq} \sim \text{Bernoulli}(Z_p^T C Z_q),$$

và độc lập với Y_{rs} khi $(p, q) \neq (r, s)$, với điều kiện biết Z_p và Z_q. Điều này có nghĩa là tổng số cạnh giữa hai khối i và j là một biến ngẫu nhiên phân phối nhị thức với kỳ vọng bằng tích của C_{ij} và số lượng dyad có sẵn. Với đồ thị vô hướng, số lượng dyad sẽ là:

$$\frac{N_i N_j}{2}$$

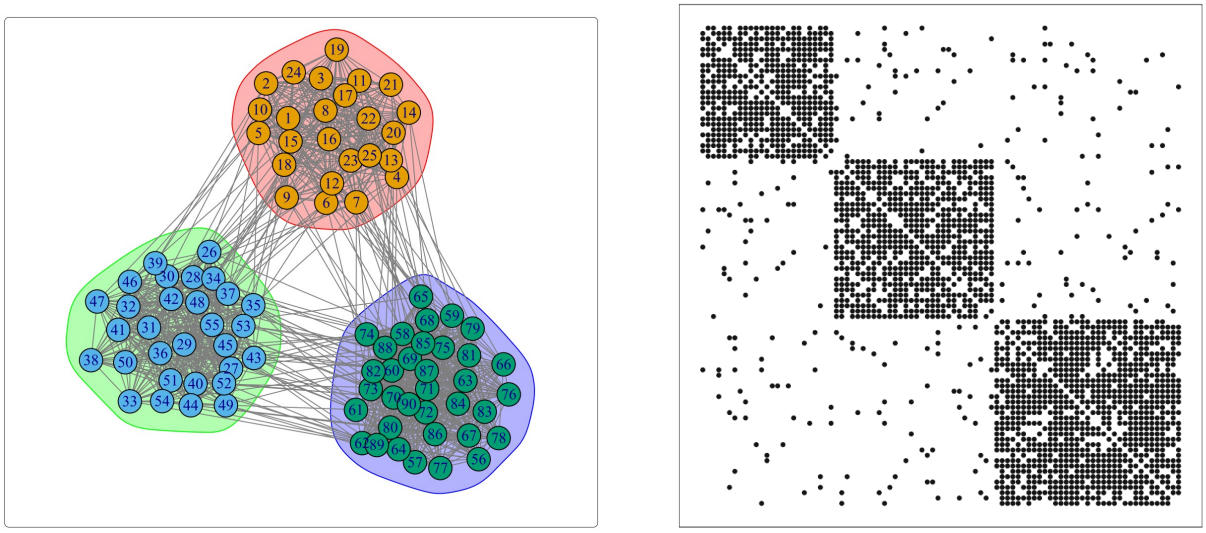
3.2 Phương pháp dựa trên mô hình xác suất

3.2.1 Mô hình Stochastic Block Model (SBM)

Mô hình khối ngẫu nhiên (**Stochastic Block Model – SBM**) là một mô hình sinh dữ liệu thường được sử dụng để phân tích cấu trúc cộng đồng trong mạng. Trong SBM, các nút của đồ thị được phân vào các nhóm (hay khối), và xác suất xuất hiện một cạnh giữa hai nút phụ thuộc vào nhóm mà chúng thuộc về. Cụ thể, các nút trong cùng một nhóm thường có xu hướng kết nối với nhau nhiều hơn so với các nút thuộc các nhóm khác.

Lợi thế của SBM là cung cấp một cách tiếp cận có tính hệ thống để sinh ra dữ liệu mạng, từ đó cho phép đánh giá hiệu quả của các thuật toán phát hiện cộng đồng trong một thiết lập có *"sự thật nền"* (ground truth). Tuy nhiên, SBM cũng có một số hạn chế, đặc biệt là ở khả năng phản ánh chính xác đặc điểm của mạng trong thực tế. Dù không nhất thiết phải là mô hình hoàn hảo, SBM vẫn hữu ích nếu có thể cung cấp góc nhìn sâu sắc về cấu trúc dữ liệu.

Để minh họa, nhóm xét ví dụ được trình bày trong Hình 3.1, gồm một mạng với 90 nút và 1192 cạnh. Các nút được chia thành ba nhóm: nhóm 1 có 25 nút, nhóm 2 có 30 nút, và nhóm 3 có 35 nút. Mô hình thể hiện rõ sự kết nối dày đặc giữa các nút trong cùng một nhóm, trong khi mức độ kết nối giữa các nhóm là thưa thớt hơn. Đáng chú ý, các kết nối trong mỗi nhóm được phân bổ tương đối đồng đều, ví dụ nút 1 không có nhiều kết nối vượt trội so với các nút khác trong cùng nhóm hoặc với các nhóm khác.



(a) Mạng gồm 90 nút (b) Ma trận kề của đồ thị

Hình 3.1: Biểu diễn mạng và ma trận kề tương ứng

3.2.1.1 Tương đương ngẫu nhiên

Một giả định quan trọng trong mô hình SBM là xác suất tồn tại cạnh giữa hai nút chỉ phụ thuộc vào nhóm mà các nút đó thuộc về. Giả định này dựa trên khái niệm *tương đương ngẫu nhiên* (stochastic equivalence). Hiểu đơn giản, nếu hai nút p và q thuộc cùng một nhóm, thì chúng được xem là "tương đương" về mặt xác suất – tức là xác suất mà p kết nối với một nút bất kỳ r sẽ giống hệt (và độc lập) với xác suất mà q kết nối với r.

Điều này phản ánh quan sát thực nghiệm trong ví dụ đã nêu: chẳng hạn, nút 1 không có xu hướng kết nối nhiều hơn hay ít hơn so với các nút khác trong cùng nhóm. Mặc dù xác suất kết nối có thể thay đổi tùy theo nhóm của nút r, nhưng trong phạm vi nhóm của mình, mọi nút đều "bình đẳng" về mặt kết nối – đó chính là tính chất tương đương ngẫu nhiên của SBM.

3.2.1.2 Mô hình SBM và ước lượng hợp lý

Với các tham số Z (nhóm của các nút) và ma trận xác suất kết nối C đã cho, xác suất sinh ra toàn bộ ma trận kề Y trong mô hình SBM (cho đồ thị vô hướng G và không có cạnh tự nối) được tính như sau:

$$\pi(Y|Z,C) = \prod_{p<q}^{n} \pi(Y_{pq}|Z,C) = \prod_{p<q}^{n} \left(Z_p^T C Z_q\right)^{Y_{pq}} \left(1 - Z_p^T C Z_q\right)^{(1-Y_{pq})}. \tag{3.1}$$

Trong đó: $Z_p^T C Z_q$ là xác suất có cạnh giữa nút p và q.

3.2.1.3 Ước lượng tham số

Trong thực tế, cả hai tham số Z và C thường chưa được biết và cần được ước lượng từ dữ liệu. Do đó, một số giả định và phương pháp ước lượng tham số được áp dụng như sau:

1. Ước lượng ma trận xác suất kết nối C:

Khi biết phân nhóm Z, ta có thể ước lượng xác suất kết nối giữa nhóm i và j bằng công thức:

$$\hat{C}ij = \frac{Eij}{N_i N_j}, \tag{3.2}$$

trong đó: E_{ij} là số cạnh quan sát được giữa hai nhóm i và j, N_i, N_j là số lượng nút trong các nhóm tương ứng.

2. Ước lượng phân nhóm Z:

Giả định rằng mỗi nút p được gán ngẫu nhiên vào một trong K nhóm với xác suất $\theta = (\theta_1, \theta_2, \ldots, \theta_K)^T$, thỏa mãn điều kiện:

$$\sum_{i=1}^{K} \theta_i = 1. \tag{3.3}$$

Do đó, biến ẩn Z_p tuân theo phân phối đa thức (Multinomial), và xác suất gán nhóm cho toàn bộ các nút là:

$$\pi(Z|\theta) = \prod_{p=1}^{n} \theta^T Z_p = \prod_{i=1}^{K} \theta_i^{N_i}, \tag{3.4}$$

trong đó N_i là số lượng nút thuộc nhóm i.

Ngoài ra, ta có thể giả định thêm rằng vector xác suất θ tuân theo phân phối Dirichlet với tham số α, tức là:

$$\theta \sim \text{Dirichlet}(\alpha \cdot \mathbf{1}_K), \tag{3.5}$$

với α được chọn từ phân phối tiên nghiệm Gamma(a, b).

3.3 Phương pháp dựa trên phân rã đồ thị

3.3.1 Các khái niệm cơ bản

Xét một đồ thị vô hướng có trọng số được ký hiệu là $G(V, E, w)$, trong đó V là tập hợp các đỉnh, E là tập hợp các cạnh, và $w_{ij} = w_{ji}$ là trọng số tương ứng với mỗi cạnh $(i, j) \in E$.

Trong trường hợp đồ thị không trọng số, ta giả sử mọi trọng số đều bằng 1, tức $w_{ij} = 1$ với mọi cạnh (i, j).Một số khái niệm cơ bản trong đồ thị được định nghĩa như sau: tập láng giềng của một đỉnh i, ký hiệu là J_i, bao gồm tất cả các đỉnh j sao cho tồn tại cạnh nối giữa i và j, tức $J_i = j \mid (i, j) \in E$. Bậc có trọng số của đỉnh i, ký hiệu là K_i, được tính bằng tổng trọng số của các cạnh nối từ đỉnh i đến các đỉnh láng giềng, tức $K_i = \sum_{j \in J_i} w_{ij}$. Tổng số đỉnh trong đồ thị được ký hiệu là $N = |V|$, còn tổng số cạnh là $M = |E|$. Cuối cùng, tổng trọng số của toàn bộ các cạnh trong đồ thị được ký hiệu là m và được tính bằng công thức:

$$m = \frac{1}{2} \sum_{i,j \in V} w_{ij}$$

trong đó hệ số $\frac{1}{2}$ được sử dụng để tránh đếm trùng các cạnh trong đồ thị vô hướng.

3.3.2 Phát hiện cộng đồng

Phát hiện cộng đồng không trùng lặp là quá trình xác định một ánh xạ $C : V \to \Gamma$, trong đó mỗi đỉnh $i \in V$ được gán vào đúng một cộng đồng $c \in \Gamma$. Nói cách khác, mỗi đỉnh chỉ thuộc về một và chỉ một cộng đồng duy nhất. Để mô tả rõ hơn, ta sử dụng một số ký hiệu như sau:

Tập hợp các đỉnh thuộc về một cộng đồng c được ký hiệu là V_c. Cộng đồng mà một đỉnh i thuộc về được ký hiệu là C_i. Tập các đỉnh láng giềng của i nằm trong cộng đồng c được ký hiệu là $J_{i \to c}$, trong đó:

$$J_{i \to c} = \{j \mid j \in J_i \text{ và } C_j = c\}$$

nghĩa là bao gồm tất cả các đỉnh j vừa là láng giềng của i, vừa thuộc về cộng đồng c.

Tổng trọng số các cạnh nối đỉnh i với cộng đồng c được ký hiệu là $K_{i \to c}$, được tính theo công thức:

$$K_{i \to c} = \sum_{j \in J_{i \to c}} w_{ij}$$

Tổng trọng số của tất cả các cạnh nội bộ trong cộng đồng c, tức là các cạnh mà cả hai đầu mút đều thuộc cộng đồng đó, được ký hiệu là σ_c và tính bởi:

$$\sigma_c = \sum_{(i,j)\in E, C_i=C_j=c} w_{ij}$$

Ngoài ra, ta cũng định nghĩa Σ_c là tổng trọng số của tất cả các cạnh mà có ít nhất một đầu mút thuộc cộng đồng c, được cho bởi:

$$\Sigma_c = \sum_{(i,j)\in E, C_i=c} w_{ij}$$

Các khái niệm và ký hiệu này sẽ được sử dụng trong việc định nghĩa các độ đo chất lượng cộng đồng và thuật toán phát hiện cộng đồng trong các phần tiếp theo.

3.3.3 Modularity

Modularity đo lường chất lượng của các cộng đồng được xác định bằng thuật toán heuristic. Nó được tính là sự chênh lệch giữa tỷ lệ các cạnh nằm trong cộng đồng và tỷ lệ kỳ vọng nếu các cạnh được phân bố ngẫu nhiên. Giá trị của modularity nằm trong khoảng $[-0.5, 1]$, trong đó giá trị cao hơn cho thấy cộng đồng có cấu trúc chặt chẽ hơn.

Modularity Q được xác định bằng công thức:

$$Q = \frac{1}{2m} \sum_{(i,j)\in E} \left[w_{ij} - \frac{K_i K_j}{2m} \right] \delta(C_i, C_j) \tag{3.6}$$

trong đó $\delta(x, y)$ là hàm Kronecker delta, nhận giá trị 1 nếu $x = y$ và 0 nếu ngược lại.

Công thức trên có thể viết lại dưới dạng:

$$Q = \sum_{c\in\Gamma} \left[\frac{\sigma_c}{2m} - \left(\frac{\Sigma_c}{2m} \right)^2 \right] \tag{3.7}$$

Delta modularity khi di chuyển một đỉnh i từ cộng đồng d sang cộng đồng c, ký hiệu $\Delta Q_{i:d\to c}$, được tính theo công thức:

$$\Delta Q_{i:d\to c} = \frac{1}{m}(K_{i\to c} - K_{i\to d}) - \frac{K_i}{2m^2}(K_i + \Sigma_c - \Sigma_d) \tag{3.8}$$

3.3.4 Thuật toán Girvan-Newman

Thuật toán Girvan-Newman là một phương pháp cổ điển trong phân tích cấu trúc cộng đồng của mạng, được đề xuất bởi Michelle Girvan và Mark Newman vào năm 2002. Điểm nổi bật của thuật toán là khả năng phát hiện các cộng đồng tiềm ẩn trong mạng mà không cần biết trước số lượng cộng đồng, một ưu điểm lớn so với nhiều phương pháp phân cụm truyền thống.

3.3.4.1 Nguyên lý hoạt động

Thuật toán dựa trên khái niệm *độ trung gian của cạnh* (edge betweenness centrality), được định nghĩa là số lượng đường đi ngắn nhất giữa các cặp đỉnh trong mạng đi qua cạnh đó. Các cạnh có độ trung gian cao thường đóng vai trò là cầu nối giữa các cộng đồng, do đó nếu loại bỏ những cạnh này, mạng sẽ có xu hướng bị chia cắt theo các cụm liên kết chặt chẽ hơn.

Thuật toán được thực hiện theo các bước chính sau:

Bước 1: Tính độ trung gian của cạnh

Với mỗi cạnh e trong đồ thị, ta sẽ tính số lượng đường đi ngắn nhất giữa tất cả các cặp đỉnh (s, t) có đi qua e. Gọi σ_{st} là tổng số đường đi ngắn nhất từ đỉnh s đến t, và $\sigma_{st}(e)$ là số đường đi trong số đó đi qua cạnh e. Khi đó, độ trung gian của cạnh e sẽ được tính theo công thức:

$$C_B(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}$$

Bước 2: Loại bỏ cạnh có độ trung gian cao nhất

Cạnh có độ trung gian cao nhất được coi là "liên kết yếu" giữa các cộng đồng và sẽ bị loại bỏ khỏi đồ thị. Và việc loại bỏ cạnh có thể làm cho mạng bị phân tách thành nhiều thành phần liên thông, tương ứng với các cộng đồng riêng biệt.

Bước 3: Lặp lại quá trình

Sau mỗi lần loại bỏ cạnh, ta tính lại độ trung gian cho tất cả các cạnh còn lại. Quá trình này được lặp lại liên tục cho đến khi không còn cạnh nào trong mạng, hoặc khi đạt đến số lượng cộng đồng mong muốn.

Bước 4: Xác định cộng đồng

Khi mạng bị chia thành các thành phần liên thông riêng biệt (connected components), mỗi thành phần được xem là một cộng đồng.

3.3.4.2 Mô tả thuật toán

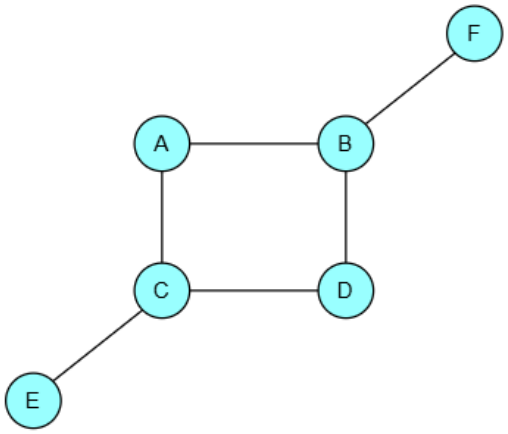
Algorithm 1 Thuật toán Girvan-Newman

Require: Đồ thị đầu vào $G = (V, E)$
Ensure: Phân cụm cộng đồng C
while G *chưa bị phân tách hoàn toàn* **do**
1:
Tính độ trung gian của tất cả các cạnh trong E
2: Xác định cạnh e^* có độ trung gian cao nhất
3: Loại bỏ e^* khỏi đồ thị G
4: Cập nhật cấu trúc của G
5: Kiểm tra số thành phần liên thông trong G
6:
7: $C \leftarrow$ tập hợp các thành phần liên thông còn lại trong G
8: **return** C =0

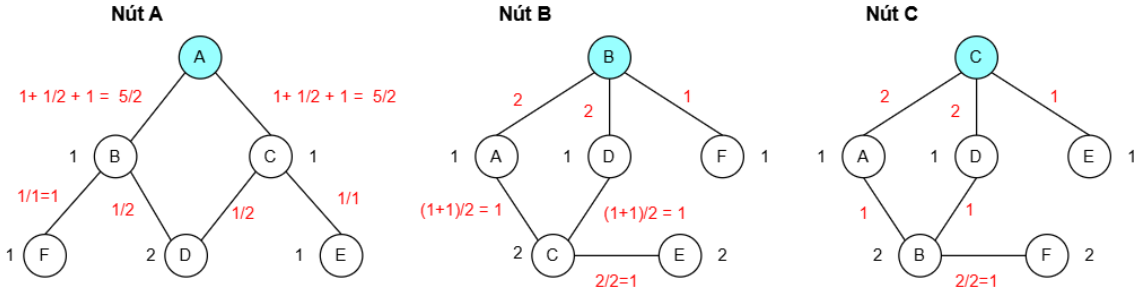
‘

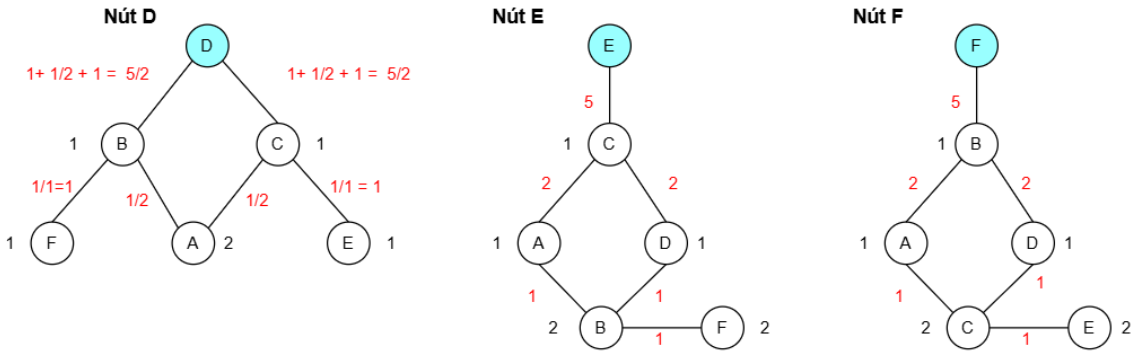
3.3.4.3 Ví dụ mô phỏng

Cho đồ thị như hình vẽ

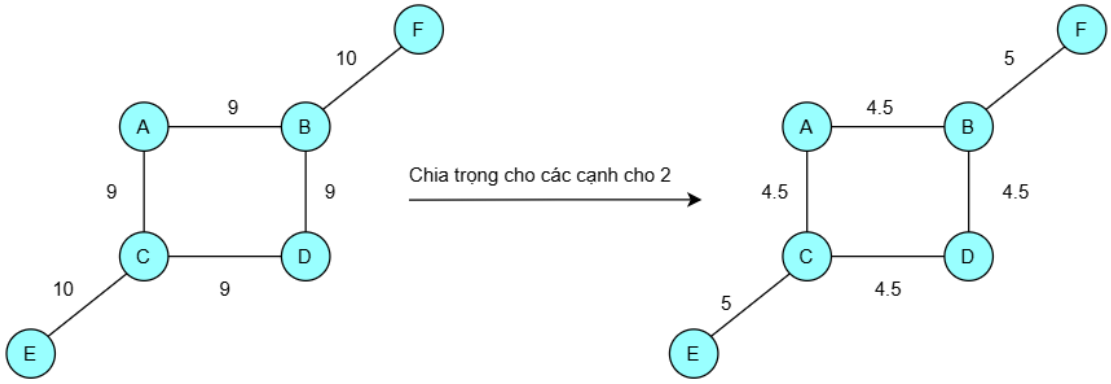


Bước 1: Đếm số lượng đường đi ngắn nhất từ mỗi nút đến các nút khác trong đồ thị và tính betweenness centrality cho mỗi trường hợp

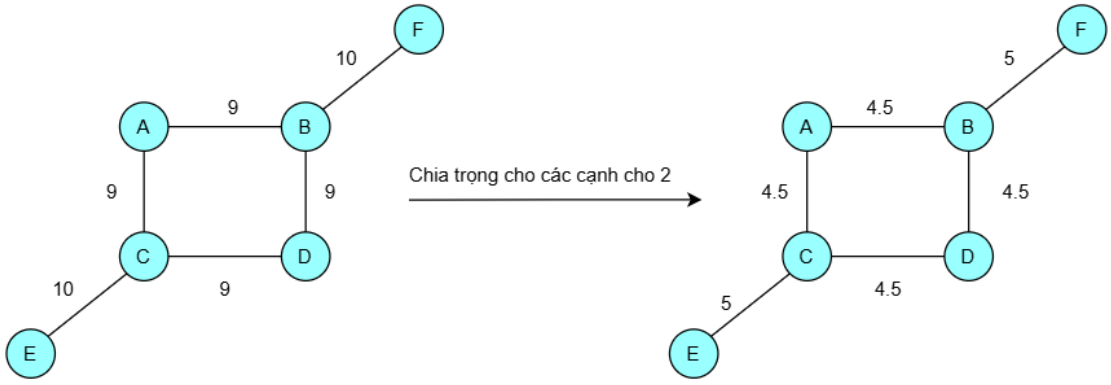




Khi đó ta sẽ được bảng:



Vì tất cả các cạnh trong đồ thị đều là hai chiều do đó giữa tâm của từng cạnh riêng lẻ sẽ giảm đi 1 nửa cho nên ta phải chia trọng số của từng cạnh cho 2.



Bước 2: Loại bỏ các cạnh có betweenness centrality cao nhất khi đó ta sẽ có được đồ thị sau khi loại bỏ.

	A	B	C	D	E	F	Tổng
AB	5/2	2	1	1/2	1	2	9
AC	5/2	1	2	1/2	2	1	9
BD	1/2	2	1	5/2	1	2	9
CD	1/2	1	2	5/2	2	1	9
CE	1	1	1	1	5	1	10
BF	1	1	1	1	1	5	10

Bước 3: Lặp lại bước 1 và bước 2 cho đến khi trọng số không còn thay đổi. Đối với ví dụ này sau khi xóa cạnh CE và BF thì ta thấy rằng trọng số trên từng cạnh không còn thay đổi nên dừng thuật toán. Khi đó ta đã phan biệt được 3 cộng đồng {E}, {A, B, C, D} và {F}.

	A	B	C	D	E	F	Tổng
AB	5/2	2	1	1/2	1	2	9
AC	5/2	1	2	1/2	2	1	9
BD	1/2	2	1	5/2	1	2	9
CD	1/2	1	2	5/2	2	1	9
CE	1	1	1	1	5	1	10
BF	1	1	1	1	1	5	10

3.3.5 Thuật toán Louvain

Thuật toán Louvain, được giới thiệu bởi Blondel và cộng sự vào năm 2008, là một phương pháp hiệu quả để phát hiện cấu trúc cộng đồng trong mạng lưới. Đây là thuật toán lặp mang tính tham lam (greedy), hoạt động bằng cách tối đa hóa độ đo *modularity*.

3.3.5.1 Nguyên lý hoạt động

Thuật toán bao gồm hai giai đoạn chính, được lặp đi lặp lại cho đến khi modularity hội tụ:

1. *Giai đoạn 1 – Gán lại cộng đồng*: Ban đầu, mỗi đỉnh được xem là một cộng đồng riêng biệt. Thuật toán duyệt qua từng đỉnh i, và xét việc chuyển đỉnh này sang cộng đồng của các đỉnh kề, sao cho giá trị modularity tăng nhiều nhất. Nếu không có sự cải thiện, đỉnh giữ nguyên cộng đồng ban đầu.

2. *Giai đoạn 2 – Tạo đồ thị rút gọn*: Sau khi không thể cải thiện modularity thêm ở giai đoạn 1, các cộng đồng hiện tại sẽ được nén lại thành các siêu đỉnh (meta-node). Trọng số của các cạnh mới giữa các siêu đỉnh được xác định dựa trên tổng trọng số của các liên kết giữa các cộng đồng ban đầu. Đồ thị rút gọn này được sử dụng làm đầu vào cho vòng lặp tiếp theo.

Quá trình này tiếp tục cho đến khi modularity không còn cải thiện đáng kể nữa. Kết quả cuối cùng là một cấu trúc phân cấp các cộng đồng, từ mức chi tiết đến mức khái quát hơn.

3.3.5.2 Tính toán modularity

Khi đánh giá việc chuyển một đỉnh i từ cộng đồng hiện tại $C(i)$ sang cộng đồng lân cận $C(j)$, mức thay đổi modularity được tính theo công thức:

$$Q_i^{C(j)} = e_i^{C(j)} - e_i^{C(i)} + \frac{m}{2}\left(\frac{k_i a_{C(i)}}{2m} - \frac{k_i a_{C(j)}}{2m}\right)^2 \tag{3.9}$$

Trong đó, $Q_i^{C(j)}$ biểu thị mức thay đổi độ đo modularity khi chuyển đỉnh i sang cộng đồng $C(j)$, và $e_i^{C(j)}$ là số liên kết giữa đỉnh i với các nút trong cộng đồng $C(j)$. Ký hiệu k_i thể hiện bậc (số liên kết) của đỉnh i. Hai đại lượng $a_{C(i)}$ và $a_{C(j)}$ lần lượt là tổng bậc của tất cả các nút trong cộng đồng $C(i)$ và $C(j)$. Cuối cùng, m là tổng trọng số của tất cả các liên kết trong toàn bộ đồ thị.

Đỉnh i sẽ được gán vào cộng đồng $C(j)$ sao cho $Q_i^{C(j)}$ đạt giá trị lớn nhất:

$$C(i) = \arg\max_{C(j)} Q_i^{C(j)}$$

Một trong những điểm mạnh của thuật toán Louvain là khả năng mở rộng tốt với độ phức tạp trung bình trên mỗi vòng lặp là $O(M)$, với M là số cạnh của đồ thị. Nhờ việc tối ưu cấu trúc dữ liệu và chỉ xét các cộng đồng lân cận, quá trình tính toán được thực hiện rất nhanh chóng.

Trên thực tế, Louvain thường hội tụ chỉ sau vài chục vòng lặp và số giai đoạn nén cộng đồng cũng không nhiều. Điều này giúp thuật toán trở thành một trong những lựa chọn phổ biến nhất để phân cụm cộng đồng trong mạng lớn.

3.3.5.3 Ví dụ mô phỏng

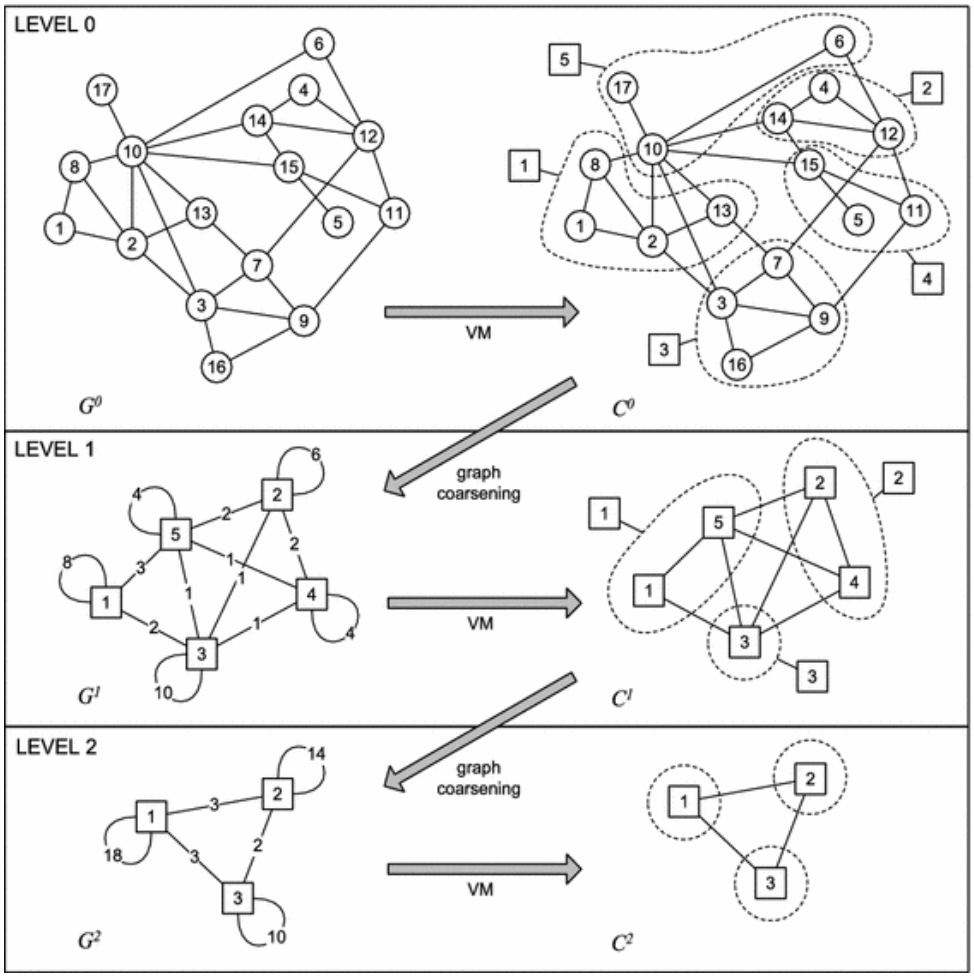
Hình 3.2 minh họa quy trình của thuật toán Louvain trong việc phát hiện cộng đồng trong một đồ thị. Quá trình này diễn ra theo từng cấp độ lặp lại, bao gồm Level 0, Level 1, Level 2, v.v., với mỗi cấp độ tương ứng với một bước trong quá trình tối ưu hóa modularity.

Ở Level 0 - Tối ưu hóa cục bộ ban đầu, mỗi đỉnh trong đồ thị ban đầu G^0 được xem là một cộng đồng riêng biệt. Thuật toán sẽ duyệt qua từng đỉnh và xem xét việc di chuyển đỉnh đó sang cộng đồng của các đỉnh kề nếu hành động này giúp cải

thiện độ đo modularity. Quá trình này lặp lại cho đến khi không còn sự cải thiện nào, từ đó tạo ra một phân cụm tạm thời C^0.

Sang Level 1 - Rút gọn đồ thị (Graph Coarsening), các cộng đồng C^0 được nén lại thành các siêu nút để tạo thành một đồ thị mới G^1, trong đó mỗi siêu nút đại diện cho một cộng đồng đã phát hiện ở cấp độ trước. Trên đồ thị G^1, thuật toán tiếp tục tối ưu hóa modularity như bước trước, thu được phân cụm C^1.

Ở Level 2 và các cấp độ tiếp theo, quy trình này tiếp tục được lặp lại: các cộng đồng được phát hiện sẽ tiếp tục được thu gọn thành siêu nút để tạo đồ thị mới, và thuật toán lại thực hiện tối ưu hóa modularity. Quá trình lặp này dừng lại khi giá trị modularity hội tụ, tức là không còn cải thiện đáng kể nào xảy ra.



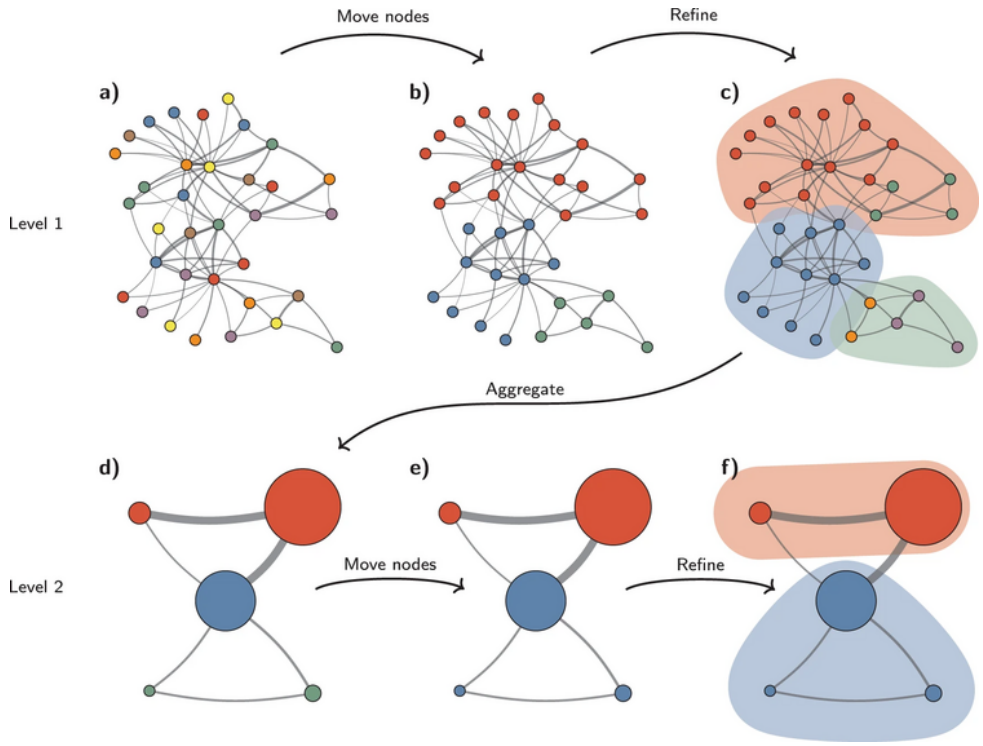
Hình 3.2: Mô tả cách hoạt động của thuật toán Louvain Yang et al. (2016)

3.3.6 Thuật toán Leiden

3.3.6.1 Nguyên lý hoạt động

Thuật toán *Leiden* được phát triển nhằm khắc phục các hạn chế của thuật toán Louvain, đặc biệt là vấn đề cộng đồng bị chia cắt hoặc có mức kết nối kém. Leiden

cải thiện hiệu quả phát hiện cộng đồng bằng cách bổ sung các bước giúp đảm bảo tính kết nối tốt hơn của các cộng đồng.



Hình 3.3: Vấn đề của thuật toán

Thuật toán *Leiden* bắt đầu với một đồ thị gồm các nút chưa được tổ chức (Hình 3.3a) và phân vùng chúng nhằm tối đa hóa tính mô-đun, tức là sự khác biệt về chất lượng giữa phân vùng được tạo ra và một phân vùng ngẫu nhiên giả định của các cộng đồng. Phương pháp được sử dụng tương tự như thuật toán Louvain, ngoại trừ việc sau khi di chuyển mỗi nút, thuật toán còn xem xét các nút lân cận chưa thuộc về cộng đồng mà nút đó được đưa vào. Quá trình này tạo ra phân vùng đầu tiên $\mathcal{P}$ (Hình 3.3b)

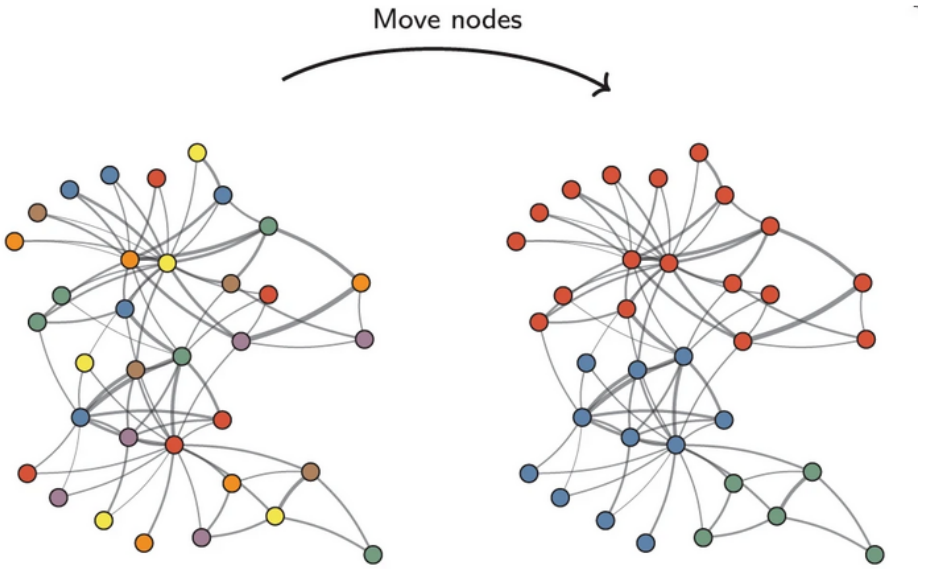
Tiếp theo, Leiden thực hiện bước tinh chỉnh phân vùng bằng cách phân tích lại các cộng đồng trong $\mathcal{P}$. Cụ thể, mỗi nút sẽ được tách ra thành một cộng đồng riêng biệt, rồi được lần lượt di chuyển sang cộng đồng khác nếu giúp cải thiện modularity hoặc đảm bảo tính liên thông bên trong cộng đồng. Quá trình này tạo ra phân vùng tinh chỉnh $\mathcal{P}_{\text{refined}}$ (Hình 3.3c), trong đó mỗi cộng đồng được đảm bảo là liên thông.

Sau đó, một mạng tổng hợp được xây dựng (Hình 3.3d), trong đó mỗi cộng đồng từ phân vùng tinh chỉnh sẽ được gộp thành một nút. Đồ thị tổng hợp này sẽ là đầu vào cho vòng lặp tiếp theo. Trong các vòng lặp tiếp theo, quá trình phân vùng

và tinh chỉnh tiếp tục được lặp lại (Hình 3.3e), cho đến khi không còn sự thay đổi nào (Hình 3.3f). Khi đó, thuật toán đạt đến trạng thái hội tụ.

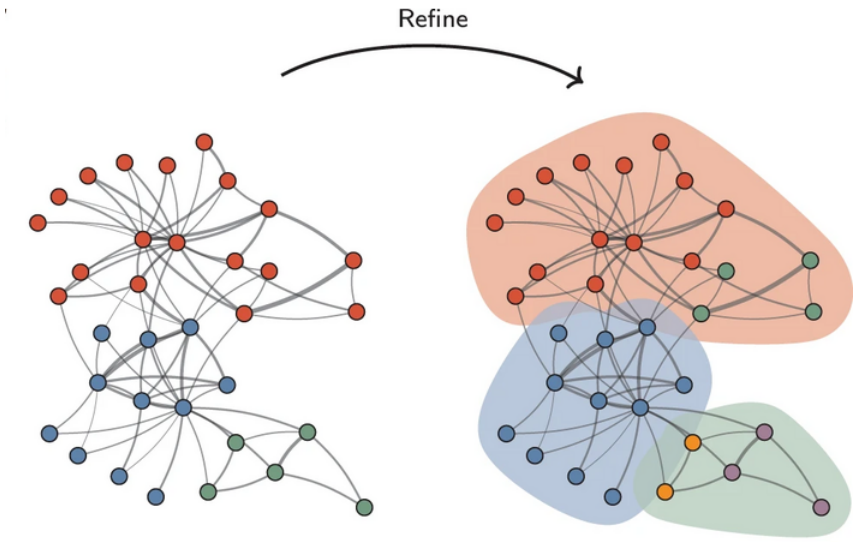
3.3.6.2 Ví dụ mô phỏng

Thuật toán Leiden hoạt động dựa trên ba bước chính nhằm phát hiện cộng đồng hiệu quả hơn so với Louvain. Thứ nhất, trong bước di chuyển cục bộ các nút, các nút sẽ được chuyển sang cộng đồng lân cận nếu việc di chuyển đó giúp cải thiện độ đo modularity. Điểm cải tiến của Leiden so với Louvain nằm ở chỗ: sau mỗi lần di chuyển, thuật toán sẽ cập nhật lại cộng đồng của các nút lân cận, từ đó giúp quá trình hội tụ nhanh hơn và kết quả chính xác hơn.



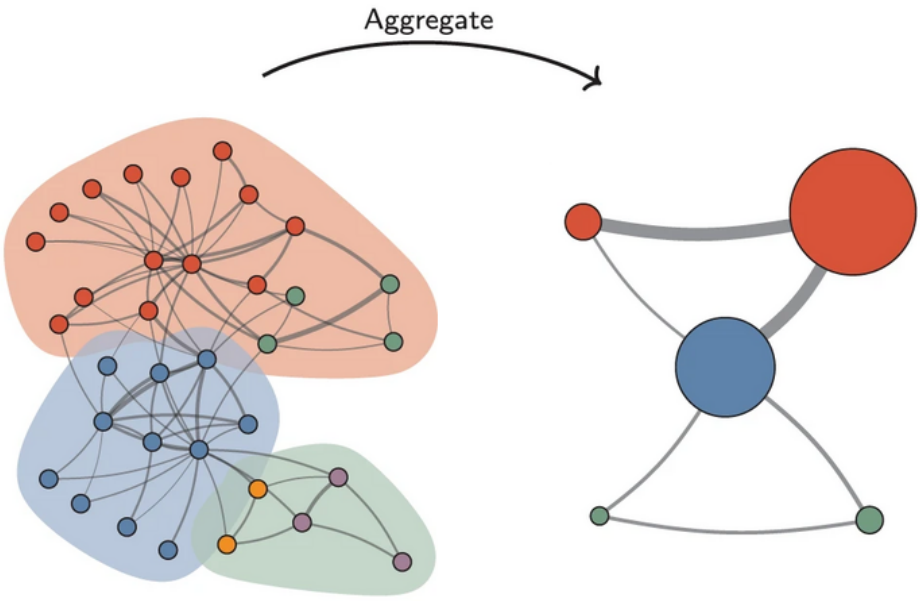
Hình 3.4: Di chuyển nút cục bộ như Lauvain

Thứ hai, thuật toán thực hiện tinh chỉnh phân vùng, nhằm đảm bảo rằng mỗi cộng đồng được phát hiện là một cấu trúc liên thông, tránh hiện tượng cộng đồng rời rạc vốn thường gặp trong Louvain. Trong hình minh họa, các nút ban đầu chưa thuộc về cộng đồng nào (bên trái), sau bước này đã được gán vào ba cộng đồng riêng biệt (bên phải), thể hiện bằng các màu sắc khác nhau như đỏ, xanh dương và xanh lá cây.



Hình 3.5: Tinh chỉnh phân

Cuối cùng, trong bước tổng hợp mạng, mỗi cộng đồng được nén lại thành một nút mới, tạo thành một đồ thị rút gọn. Thuật toán tiếp tục lặp lại ba bước trên với đồ thị mới này cho đến khi modularity không còn cải thiện đáng kể.



Hình 3.6: Tổng hợp

Tất cả các bước trên được thực hiện trong vòng lặp chính của thuật toán Leiden, trong đó phương pháp Louvain được tối ưu hóa bằng kỹ thuật "Fast Louvain" như được đề xuất trong nghiên cứu Ozaki et al. (2016).

Chương 4

MÔ HÌNH PHÂN TÍCH ẢNH HƯỞNG CỘNG ĐỒNG

4.1 Mô hình phân tích ảnh hưởng cộng đồng

4.1.1 Định nghĩa và công thức tính toán

Mô hình phân tích ảnh hưởng cộng đồng (**Community Influence Model - CIM**) là một mô hình nghiên cứu nhằm đo lường và đánh giá mức độ ảnh hưởng của các cá nhân, nhóm hoặc sự kiện trong một cộng đồng nhất định. Mô hình này dựa trên lý thuyết mạng lưới xã hội và lan truyền thông tin, trong đó các tác nhân trong cộng đồng tương tác với nhau thông qua các kênh truyền thông, dẫn đến sự thay đổi nhận thức, thái độ hoặc hành vi.

CIM được sử dụng để xác định các yếu tố ảnh hưởng chính, phân tích cấu trúc cộng đồng và đánh giá cơ chế lan truyền tác động thông qua các mô hình toán học và thống kê. Nhờ đó, mô hình này có ứng dụng rộng rãi trong các lĩnh vực như khoa học xã hội, truyền thông, tiếp thị, kinh tế và chính trị, hỗ trợ việc hiểu rõ hơn về động lực tác động và sự lan tỏa thông tin trong cộng đồng.

4.1.2 Thuật toán

Mô hình ảnh hưởng cộng đồng (*CIM*) dùng để mô hình hóa các ảnh hưởng không đồng nhất trong một mạng lưới. Cụ thể, trước tiên ta xây dựng một mạng lưới n nút và định nghĩa ma trận kề $A = (a_{ij})_{n\times n} \in \mathbb{R}^{n\times n}$ để mô tả mối quan hệ giữa 2 nút bất kỳ, trong đó $a_{ij} = 1$ nếu có một cạnh nối từ nút i sang nút j hoặc bằng 0 nếu không có và $a_{ii} = 0$. Tiếp theo, ta thu thập các biến phản hồi $Y_i \in \mathbb{R}^1$ và các biến giải thích p-chiều liên quan $X_i = (X_{i1}, \ldots, X_{ip})^\top \in \mathbb{R}^p$. Định nghĩa, $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top \in \mathbb{R}^n$ là vecto phản hồi và $\mathbf{X} = (X_1, \ldots, X_n)^\top \in \mathbb{R}^{p\times n}$ là ma trận

phụ thuộc.

Ta giả định rằng các nhóm mạng lưới có thể được phân loại thành K nhóm không chồng chéo lên nhau $\{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$ với $1 \leq K < \infty$, trong đó $\mathcal{G}_k \subseteq [n]$ và $\bigcup_{k=1}^{K} \mathcal{G}_k = [n]$. Nghĩa là các nút này chỉ thuộc một nhóm và các nút trong cùng một nhóm sẽ cùng chia sẻ một tham số ảnh hưởng chung. Thêm vào đó với mỗi nút i, g_i là nhóm của nút i với $i \in \mathcal{G}_{g_i}$. Cuối cùng, ta định nghĩa công thức CIM như sau:

$$Y_i = \sum_{j=1}^{n} \lambda^{(g_j)} w_{ij} Y_j + X_i^\top \alpha + \epsilon_i, \tag{4.1}$$

hay

$$\mathbf{Y} = \mathbf{W}\Lambda\mathbf{Y} + \alpha X + \mathcal{E}, \tag{4.2}$$

Trong đó, $\mathbf{W} = (w_{ij}) \in \mathbb{R}^{n \times n}$ là ma trận kề có trọng số với $w_{ij} = \frac{a_{ij}}{\sum_{j=1}^{n} a_{ij}}$, $\bigcup_{i=1}^{n} g_i = [K]$, $\Lambda = \text{diag}\{\lambda^{(g_1)}, \ldots, \lambda^{(g_n)}\}$, $\alpha = (\alpha_1, \ldots, \alpha_p)^\top \in \mathbb{R}^p$ là vecto hệ số hồi quy p-chiều và $\mathcal{E} = (\epsilon_1, \ldots, \epsilon_n)^\top \in \mathbb{R}^n$ là vecto sai số. Giả định rằng các ϵ_i độc lập và phân phối giống nhau với trung bình 0 và phương sai σ^2. Theo mô hình (4.1), hiệu ứng ảnh hưởng của Y_j lên Y_i là $\lambda^{(g_j)} w_{ij}$. Do đó, cho kết nối mạng w_{ij}, $\lambda^{(g_j)}$ càng lớn, ảnh hưởng của nó lên Y_i càng mạnh. Thêm vào đó, $\lambda^{(g_j)} w_{ij}$ là ma trận ảnh hưởng của cộng đồng và các nút trong nhóm k có một tham số ảnh hưởng chung λ^k với $k = 1, \ldots, K$. Điều đáng chú ý là mô hình (4.2) chứa mô hình hồi quy tự động mạng cổ điển (NAR) của Manski (1993) như một trường hợp đặc biệt khi chỉ có một nhóm và tất cả các nhóm đều giống nhau. Bằng cách tích hợp khái niệm của một mô hình ngẫu nhiên cổ điển với mô hình tự hồi quy cổ điển, mô hình ảnh hưởng cộng đồng đề xuất một phương pháp mới cho việc phân tích mạng lưới.

Để ước lượng mô hình (4.2) và xác định số lượng nhóm K, chúng tôi định nghĩa $\lambda = (\lambda_1, \ldots, \lambda_n)^\top$ trong đó $\lambda_i = \lambda^{(g_i)}$, $S(\lambda) = I_n - \mathbf{W}\Lambda$ và $\mathcal{E} = S(\lambda)\mathbf{Y} - \lambda\alpha$ dựa trên giả định rằng $\mathcal{E}$ là một vector sai số với trung bình 0 và ma trận hiệp phương sai $\sigma^2 I_n$. Ta xem xét hàm Quasi Log-Likelihood bên dưới.

4.1.3 Hàm Quasi Log-Likelihood

Quasi Log-Likelihood *(QLL)*là một phương pháp ước lượng trong thống kê, thường được sử dụng khi phân phối xác suất thực tế của dữ liệu không được biết hoặc không phù hợp với giả định phân phối chuẩn. Thay vì tối đa hóa log-likelihood thực tế, ta sử dụng một hàm hợp lý gần đúng để ước lượng tham số.

Do $E \sim \mathcal{N}(0, \sigma^2 I_n)$, hàm mật độ xác suất của E là:

$$p(\mathcal{E}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}\mathcal{E}^\top \mathcal{E}\right).$$

Thay $\mathcal{E} = S(\lambda)Y - X\alpha$ vào, ta có:

$$p(Y \mid \alpha, \lambda, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(S(\lambda)Y - X\alpha)^\top (S(\lambda)Y - X\alpha)\right).$$

Lấy log-likelihood:

$$\ell_n(\alpha, \lambda, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(S(\lambda)Y - X\alpha)^\top (S(\lambda)Y - X\alpha).$$

Bỏ qua hằng số $-\frac{n}{2}\log(2\pi)$ không ảnh hưởng đến tối ưu hóa, ta có:

$$\ell_n(\alpha, \lambda, \sigma^2) = -\frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(S(\lambda)Y - X\alpha)^\top (S(\lambda)Y - X\alpha).$$

Vì $S(\lambda)$ thay đổi định thức, cần thêm điều chỉnh Jacobian. Do đó, hàm *quasi log-likelihood* (bỏ qua hằng số không ảnh hưởng đến tối ưu) được viết dưới dạng:

$$\ell_n(\alpha, \lambda, \sigma^2) = -\frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(S(\lambda)Y - X\alpha)^\top (S(\lambda)Y - X\alpha) + \log|\det S(\lambda)|. \quad (4.3)$$

4.1.4 Hàm Objective Function

Để xác định cấu trúc nhóm của tham số ảnh hưởng $\boldsymbol{\lambda}$, chúng tôi áp dụng một hàm phạt hợp nhất lồi-concave cho sự khác biệt từng cặp của $\boldsymbol{\lambda}$. Khi đó, hàm objective function được đề xuất có dạng:

$$Q_n(\boldsymbol{\theta}) = \ell_n(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \sigma^2) - \sum_{1 \le i < j \le n} p_\gamma(|\lambda_i - \lambda_j|), \quad (4.4)$$

trong đó $p_\gamma(\cdot)$ là một hàm phạt concave có tham số điều chỉnh $\gamma > 0$, và vector tham số đầy đủ được xác định bởi $\boldsymbol{\theta} = (\boldsymbol{\alpha}^\top, \boldsymbol{\lambda}^\top, \sigma^2)^\top$.

Hàm phạt trong phương trình (4.4) được điều chỉnh từ Fused Lasso Tibshirani et al. (2005) và các nghiên cứu trước đó như Ke et al. (2015), Qian and Su (2016), Wang et al. (2018) đã xem xét các hàm phạt tương tự nhưng chủ yếu áp dụng cho hệ số hồi quy hoặc hằng số chặn thay vì tham số ảnh hưởng.

Dựa trên (4.4), ước lượng *quasi-maximum likelihood estimator (QMLE)* của $\boldsymbol{\theta}$ được xác định thông qua bài toán tối ưu:

$$\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\alpha}}^\top, \hat{\boldsymbol{\lambda}}^\top, \hat{\sigma}^2)^\top = \arg\max_{\boldsymbol{\alpha},\boldsymbol{\lambda},\sigma^2} Q_n(\boldsymbol{\theta}). \tag{4.5}$$

Các kết quả lý thuyết trong phần 3 chứng minh rằng ước lượng $\hat{\boldsymbol{\lambda}}$ thu được có thể xác định được cấu trúc nhóm. Gọi $\{\hat{\lambda}^{(1)}, \ldots, \hat{\lambda}^{(\hat{K})}\}$ là các giá trị khác nhau của $\hat{\boldsymbol{\lambda}}$, trong đó $\hat{K}$ là số nhóm được ước lượng. Khi đó, nhóm thứ k có thể được xác định bởi:

$$\hat{\mathcal{G}}_k = \{i \mid \hat{\lambda}_i = \hat{\lambda}^{(k)}, 1 \le i \le n\}, \quad \forall 1 \le k \le \hat{K}. \tag{4.6}$$

Như đã đề cập, để xác định cấu trúc nhóm của tham số ảnh hưởng λ, ta áp dụng một hàm phạt lồi-concave đối với sự khác biệt từng cặp của λ. Hàm phạt này giúp phát hiện và phân nhóm các giá trị λ có tính chất tương đồng. Cụ thể, nhóm thứ k được xác định bởi:

$$\hat{G}_k = \{i|\hat{\lambda}_i = \hat{\lambda}^{(k)}, 1 \le i \le n\}, \quad \forall 1 \le k \le \hat{K}, \tag{4.7}$$

trong đó $\hat{K}$ là số nhóm được ước lượng.

Hàm phạt $p_\gamma(|\lambda_i - \lambda_j|)$ có vai trò thúc đẩy sự hội tụ của các giá trị λ về một số lượng hữu hạn nhóm, từ đó xác định cộng đồng có ảnh hưởng tương tự trong mô hình. Các phương pháp như fused lasso (Tibshirani et al., 2005) hay các biến thể của nó đã chứng minh hiệu quả trong việc nhóm các tham số mô hình. Như vậy, hàm objective function không chỉ giúp tìm kiếm các tham số tối ưu mà còn hỗ trợ phân cụm các thực thể có ảnh hưởng tương đồng trong cộng đồng.

4.2 Thuật toán tính toán trong CIM

4.2.1 Hàm mục tiêu ước lượng theo λ

Như vậy, ước lượng QMLE của tham số $\theta = (\alpha^\top, \lambda^\top, \sigma^2)^\top$ được xác định thông qua bài toán tối ưu:

$$\hat{\theta} = \arg\max_{\alpha,\lambda,\sigma^2} Q_n(\theta),$$

trong đó hàm mục tiêu $Q_n(\theta)$ được định nghĩa như trong phương trình (4.4).

Dưới giả định $\mathcal{E} \sim N(0, \sigma^2 I_n)$

$$Q_n(\alpha, \sigma^2, \lambda) = \ell_n(\alpha, \sigma^2, \lambda) - \sum_{1 \le i < j \le n} p_\gamma(|\lambda_i - \lambda_j|)$$

Mục tiêu: tối đa hoá $Q_n(\alpha, \sigma^2, \lambda)$ theo λ. Tức là ta đưa α, σ^2 theo λ, và tìm $\arg\max_\lambda Q_n(\lambda)$

Sử dụng phương pháp khả năng hợp lý giả cực đại tập trung (concentrated quasi-maximum likelihood approach) để ước lượng vectơ tham số chưa biết θ. Với một λ đã cho, bằng cách tối đa hóa $Q_n(\theta)$ theo α, σ^2, ta thu được:

$$\alpha(\lambda) = (X^T\hat{X})^{-1}X^T S(\lambda)Y, \quad \hat{\sigma^2}(\hat{\alpha}(\lambda), \lambda) = \frac{1}{n}Y^T S(\lambda)^T \mathcal{M}_X S(\lambda)Y \tag{4.8}$$

Trong đó $\mathcal{M}_X = I_n - X(X^TX)^{-}1X^T$. Sau đó, bằng cách cắm hai ước lượng trên vào 4.2, chúng ta ước lượng λ bằng cách tối thiểu hóa Q_{nc} bằng $-Q_n$ được định nghĩa trong phương trình sau

$$Q_{nc} = \frac{n}{2}\log\sigma^2(\hat{\alpha}(\lambda), \lambda) - \log|\det\{S(\lambda)\}| + \sum_{1\le i<j\le n} p_\gamma(|\lambda_i - \lambda_j|) \tag{4.9}$$

Để giải quyết 7.3, ta sửa đổi thuật toán phương pháp nhân tử hướng xen kẽ (ADMM) của Boyd et al. (2011) và Ma và Ma and Huang (2017) thông qua hai bước sau.

4.2.2 Áp dụng phương pháp ADMM

ADMM được sử dụng để tối ưu hóa $Q_{nc}(\lambda)$ qua các bước sau:

Algorithm 2 Phương pháp ADMM có chuyển đổi

Bước 1: Biến đổi bài toán và xây dựng hàm Lagrange tăng cường
Biến đổi bài toán về dạng tối ưu có ràng buộc
Xây dựng hàm Lagrange tăng cường với biến dual
Bước 2: Thực hiện các bước lặp ADMM
Khởi tạo λ^0, ζ^0, v^0
while *chưa hội tụ* **do**
  Cập nhật λ^{m+1} bằng phương pháp **gradient descent**
  Cập nhật ζ^{m+1} bằng phương pháp **soft thresholding**
  Cập nhật v^{m+1} theo công thức có sẵn
  Kiểm tra điều kiện dừng
end
Trả về nghiệm tối ưu λ^*

4.2.2.1 Bước 1

1. **Chuyển bài toán về dạng tối ưu có ràng buộc**
   Ta bắt đầu với hàm mục tiêu từ phương trình 4.9

$$\arg\min_\lambda \frac{n}{2}\log\sigma^2 - \log|\det\{S(\lambda)\}| + \sum_{1\le i<j\le n} p_\gamma(|\lambda_i - \lambda_j|)$$

Nhưng do hàm phạt $p_\gamma(|\lambda_i - \lambda_j|)$ không trơn (non-smooth), việc giải bài toán trở nên khó khăn. Để xử lý, ta đưa bài toán về dạng tối ưu có ràng buộc bằng cách thêm biến phụ $\zeta_{ij} = \lambda_i - \lambda_j$. Khi đó hàm mục tiêu trở thành:

$$\arg\min_{\lambda,\zeta} \quad \frac{n}{2}\log\hat{\sigma}^2(\hat{\alpha}(\lambda),\lambda) - \log|\det(S(\lambda))| + \sum_{i<j} p_\gamma(|\zeta_{ij}|) \tag{4.10}$$

với ràng buộc bằng: $\lambda_i - \lambda_j - \zeta_{ij} = 0$, $\forall 1 \le i < j \le n$

2. Xây dựng hàm Lagrange tăng cường (Augmented Lagrangian)
Hàm Lagrange tăng cường giúp chuyển đổi bài toán tối ưu có ràng buộc thành dạng không ràng buộc, có thể tìm điểm cực trị bằng đạo hàm thông thường. Có được bằng cách kết hợp cả nhân tử Lagrange và hàm phạt.

$$\begin{aligned} Q_{ncc}(\lambda,\zeta,v) = {} & \frac{n}{2}\log\hat{\sigma}^2(\hat{\alpha}(\lambda),\lambda) - \log|\det(S(\lambda))| + \sum_{i<j} p_\gamma(|\zeta_{ij}|) \\ & + \underbrace{\sum_{i<j} v_{ij}(\lambda_i - \lambda_j - \zeta_{ij})}_{\text{Hàm nhân tử Lagrange}} + \underbrace{\frac{\vartheta}{2}\sum_{i<j}(\lambda_i - \lambda_j - \zeta_{ij})^2}_{\text{Hàm phạt}} \end{aligned} \tag{4.11}$$

Trong đó: v_{ij} là các biến nhân tử Lagrange. và ϑ là tham số phạt (có thể điều chỉnh)

4.2.2.2 Bước 2: Quá trình lặp của ADMM

Quá trình lặp của ADMM với mong muốn tối ưu hóa hàm Quasi Log-Likelihood (QLL) giúp tìm ra các giá trị tham số tốt nhất cho mô hình cũng như mô phỏng chính xác các quan hệ trong dữ liệu mạng. Diễn ra quá trình cập nhật các thông số cho đến khi hội tụ.

$$\begin{aligned} \lambda^{(m+1)} &= \arg\min_{\lambda} \mathcal{Q}_{ncc}(\lambda,\zeta^{(m)},v^{(m)}), \\ \zeta^{(m+1)} &= \arg\min_{\zeta} \mathcal{Q}_{ncc}(\lambda^{(m+1)},\zeta,v^{(m)}), \\ v_{ij}^{(m+1)} &= v_{ij}^{(m)} + \vartheta(\lambda_i^{(m+1)} - \lambda_j^{(m+1)} - \zeta_{ij}^{(m+1)}). \end{aligned} \tag{4.12}$$

1. Cập nhật λ

$$\lambda^{(m+1)} = \arg\min_{\lambda} Q_{ncc}(\lambda,\zeta^{(m)},v^{(m)}) \tag{4.13}$$

Bài toán trên không có nghiệm đóng, nên ta sử dụng thuật toán gradient descent để tìm nghiệm. Hàm mục tiêu $f(\lambda)$ được định nghĩa như sau:

$$f(\lambda) = \frac{n}{2}\log\hat{\sigma}^2(\hat{\alpha}(\lambda),\lambda) - \log|\det S(\lambda)| + \frac{\vartheta}{2}\|\Delta\lambda - \zeta^{(m)} + \vartheta^{-1}v^{(m)}\|_2^2$$

(xem thêm mục 6.2.2 a)

Trong đó:

- $\hat{\sigma}^2(\hat{\alpha}(\lambda), \lambda)$ là một hàm có thể liên quan đến phương sai dự đoán.
- $S(\lambda)$ là một ma trận liên quan đến λ.
- $\zeta^{(m)}$ và $v^{(m)}$ là các giá trị từ các bước trước trong thuật toán ADMM.
- $\Delta\lambda$ là một vectơ chứa các hiệu giữa các thành phần của λ, cụ thể là các hiệu giữa các cặp chỉ số i, j.
- ϑ là một tham số trong ADMM điều chỉnh trọng số giữa các phần trong hàm mục tiêu.

Khi đó ta áp dụng thuật toán gradient descent tìm nghiệm với đạo hàm của hàm mục tiêu theo λ là:

$$\frac{\partial f(\lambda)}{\partial \lambda_i} = -\frac{1}{\hat{\sigma}^2(\hat{\alpha}(\lambda), \lambda)} \left[YY^\top S(\lambda)^\top \mathcal{M}_X W\right]_{ii} + \left[S(\lambda)^{-1} W\right]_{ii} + \vartheta \left[\Delta^\top (\Delta\lambda - \zeta^{(m)} + \vartheta^{-1} v^{(m)})\right]_i$$

(các bước đạo hàm được trình bày tại mục 6.2.2.a)

Trong đó:

- Các ký hiệu như $Y, S(\lambda), \mathcal{M}, X, W$ là các ma trận và vectơ được tính toán từ dữ liệu và các giá trị của λ.
- Δ là một ma trận chứa các hiệu giữa các thành phần của λ, như đã đề cập ở trên.

Sau khi có đạo hàm của $f(\lambda)$ theo λ, ta sử dụng Gradient Descent để cập nhật giá trị của λ trong mỗi vòng lặp.

$$\lambda^{(m+1)} = \lambda^{(m)} - \eta \frac{\partial f(\lambda)}{\partial \lambda}$$

Với: η là tốc độ học (learning rate).

Giá trị ước lượng khởi đầu $\lambda^{(0)}$ được xác định thông qua việc tối thiểu hóa hàm log-likelihood giả âm $-\ell_n(\alpha, \lambda, \sigma^2)$, như trình bày trong Phương trình (4.3). Quá trình tối ưu hóa này được thực hiện bằng phương pháp BFGS, một thuật toán tối ưu hóa gradient quasi-Newton, được triển khai thông qua hàm minimize trong thư viện `scipy.optimize`. Phương pháp này được lặp lại nhiều lần nhằm cải thiện nghiệm ước lượng. Trong bước khởi tạo, các giá trị ban đầu của các tham số được thiết lập bằng cách ước lượng mô hình tự

hồi quy mạng (Network Autoregressive - NAR) với giả định rằng tất cả các hệ số λ_i đều bằng nhau. Giả định này không những giúp đơn giản hóa bài toán mà còn cung cấp một điểm xuất phát hợp lý, hỗ trợ quá trình lặp đạt được nghiệm hội tụ ổn định.

2. Cập nhật ζ
Viết lại bài toán tối ưu Q_{nnc} theo ζ

$$\zeta_{ij}^{(m+1)} = \arg\min_{\zeta_{ij}} \left\{ \frac{\vartheta}{2} \left(\lambda_i^{(m+1)} - \lambda_j^{(m+1)} - \zeta_{ij} + \vartheta^{-1} v_{ij}^{(m)} \right)^2 + p_\gamma |\zeta_{ij}| \right\}. \tag{4.14}$$

(xem thêm tại mục 6.2.2.b)

Với:

- p_γ là hệ số điều chỉnh của Lasso, kiểm soát mức độ phạt đối với giá trị của ζ_{ij}.
- ϑ là hệ số điều chỉnh trong phương pháp Augmented Lagrangian.
- v_{ij} là nhân tử Lagrange.

Nếu sử dụng hàm phạt Lasso, nghiệm có thể tìm bằng thuật toán coordinate descent được diễn giải bên dưới.

Đạo hàm và điều kiện tối ưu
Xét đạo hàm riêng của hàm mục tiêu theo ζ_{ij}:

$$\frac{\partial}{\partial \zeta_{ij}} \left(\frac{\vartheta}{2} (\zeta_{ij} - a)^2 + p_\gamma |\zeta_{ij}| \right) = \vartheta(\zeta_{ij} - a) + p_\gamma \cdot \text{sign}(\zeta_{ij}), \tag{4.15}$$

với

$$u_{ij}^{(m+1)} = \lambda_i^{(m+1)} - \lambda_j^{(m+1)} + \vartheta^{-1} v_{ij}^{(m)} \tag{4.16}$$

Công thức cập nhật
Giải phương trình đạo hàm bằng điều kiện tối ưu bậc nhất, ta thu được nghiệm có dạng *soft-thresholding*:

$$\zeta_{ij}^{(m+1)} = \begin{cases} u_{ij}^{(m+1)} & \text{if } |u_{ij}^{(m+1)}| > \tau\gamma, \\ \frac{\mathbf{ST}(u_{ij}^{(m+1)}, \gamma/\vartheta)}{\left(1 - \frac{1}{\tau\vartheta}\right)} & \text{if } |u_{ij}^{(m+1)}| \leq \tau\gamma. \end{cases} \tag{4.17}$$

Ý nghĩa của nghiệm

- Nếu a đủ lớn ($a > \lambda/\vartheta$), ta trừ đi λ/ϑ, làm giảm giá trị tuyệt đối của ζ_{ij}.

- Nếu $|a| \leq \lambda/\vartheta$, ta đặt $\zeta_{ij} = 0$, tức là loại bỏ thành phần đó, tạo ra tính chọn lọc của Lasso.
- Nếu a âm ($a < -\lambda/\vartheta$), ta cộng thêm λ/ϑ, tương tự như trên nhưng theo hướng ngược lại.

3. Cập nhật v

$$v_{ij}^{(m+1)} = v_{ij}^{(m)} + \vartheta(\lambda_i^{(m+1)} - \lambda_j^{(m+1)} - \zeta_{ij}^{(m+1)}). \tag{4.18}$$

4. Điều kiện dừng thuật toán

Thuật toán trên sẽ dừng khi hai điều kiện sau được thỏa mãn:

$$\|\mathbf{r}^{(m+1)}\|_2 \leq \epsilon_p, \quad \|\mathbf{s}^{(m+1)}\|_2 \leq \epsilon_d \tag{4.19}$$

trong đó:

$$\epsilon_p = \sqrt{\frac{n(n-1)}{2}}\epsilon_{\text{abs}} + \epsilon_{\text{rel}} \max\{\|\Delta\boldsymbol{\lambda}^{(m+1)}\|_2, \|\boldsymbol{\zeta}^{(m+1)}\|_2\} \tag{4.20}$$

$$\epsilon_d = \sqrt{n}\epsilon_{\text{abs}} + \epsilon_{\text{rel}}\|\Delta^\top \boldsymbol{v}^{(m+1)}\|_2 \tag{4.21}$$

với:

- ϵ_{abs} và ϵ_{rel} là các giá trị nhỏ, được chọn trước để đảm bảo độ chính xác mong muốn.
- ϵ_p và ϵ_d điều chỉnh điều kiện hội tụ theo kích thước của bài toán.

4.2.2.3 Tính chất hội tụ và độ chính xác

Các kết quả lý thuyết chứng minh rằng ước lượng $\hat{\lambda}$ thu được từ phương pháp này có thể xác định chính xác cấu trúc nhóm dưới điều kiện phù hợp về tính lồi của hàm phạt. Ngoài ra, nếu lựa chọn tham số điều chỉnh γ hợp lý, mô hình có thể đạt được sự cân bằng tốt giữa tính chính xác và tính tổng quát của việc phân cụm.

Như vậy, phương pháp ước lượng **QMLE** không chỉ giúp tìm kiếm các tham số tối ưu mà còn hỗ trợ phát hiện cộng đồng có ảnh hưởng tương đồng trong dữ liệu mạng lưới.

4.2.3 Thử nghiệm mẫu

4.2.3.1 Ví dụ 1

Chúng tôi mô phỏng dữ liệu theo mô hình CIM, trong đó

$$X_i = (x_{i1}, x_{i2}, x_{i3})^\top$$

với $x_{i1} \equiv 1$ và x_{i2}, x_{i3} được tạo độc lập và phân phối giống nhau theo phân phối chuẩn chuẩn hóa $\mathcal{N}(0,1)$. Các tham số hồi quy tương ứng với ba biến đồng biến là

$$\boldsymbol{\alpha} = (5,1,1)^\top.$$

Ngoài ra, các sai số ngẫu nhiên ε_i được tạo độc lập và phân phối giống nhau theo phân phối chuẩn $\mathcal{N}(0,0.5^2)$.Hơn nữa, tồn tại $K = 2$ nhóm với $\lambda^{(1)} = 0.6$ và $\lambda^{(2)} = 0.3$. Sau đó, chúng tôi gán ngẫu nhiên các nút vào $R_d/2$ khối với tham số ảnh hưởng $\lambda^{(1)}$, và phần còn lại $R_d/2$ khối với tham số ảnh hưởng $\lambda^{(2)}$. Tức là, một khối sẽ được gán hoàn toàn là $\lambda^{(1)}$ hoặc $\lambda^{(2)}$, và các nút thuộc cùng một khối sẽ chia sẻ cùng tham số ảnh hưởng tương ứng.Trong ví dụ này, chúng tôi xét $R_d \in \{6,12\}$ và $m \in \{15,20\}$, và mạng được tạo ra bởi DBN với kích thước mạng là $n = m \times R_d$.

Rd	**m**	**K**				**Rand**		**aRand**	
		mean	sem	median	MAD	mean	sem	mean	sem
6	15	3.12	0.112993	3.0	1.0	0.647705	0.017558	0.293145	0.035
6	20	3.84	0.115564	4.0	0.5	0.611908	0.009218	0.221096	0.0185
12	15	6.00	0.257143	5.0	1.0	0.564298	0.005747	0.125525	0.0115
12	20	7.44	0.469	6.0	1.0	0.558	0.0043	0.1142	0.0086

Nhìn chung, mô hình dự đoán khá tốt (Rand 0.64) khi kích thước mạng nhỏ (Rd=6), mean và median của K đều xấp xỉ 3 hoặc 4 (chỉ sai lệch 1 so với số nhóm cộng đồng thực). Ta có thể thấy rằng: Khi tăng m từ 15 → 20 (cùng Rd = 6) thì: trung bình $\hat{K}$ tăng từ 3.12 → 3.84.Tương tự, khi tăng Rd từ 6 → 12 (cùng m = 15) thì: Trung bình $\hat{K}$ tăng từ 3.12 → 6.00. Khi số block hoặc m tăng, CIM dễ overestimate - ước lượng quá cao số nhóm K, khả năng cao là do nhiễu lớn dẫn đến việc mô hình phá vỡ cấu trúc nhóm cộng đồng thực thành nhiều nhóm con tiềm ẩn không cần thiết. Bên cạnh đó, nhận thấy rằng với Rand index (tỉ lệ cặp đúng):

» Với (6,15): 0.6478 (±0.0176) → khoảng 65%

» Với (6,20): 0.6119 → giảm nhẹ

» Với (12,15): 0.5643 → hiệu quả giảm khi mạng lớn hơn

» Với (12,20): 0.5584 → tiếp tục giảm

Còn đối với Adjusted Rand index:

» Với (6,15): 0.2931 (±0.035)

» Giảm xuống 0.1255, 0.1142... khi Rd hay m tăng

Tương tự khi thống kê về số nhóm K, hiệu quả clustering đánh giá thông qua 2 chỉ số Rand/aRand đều kém đi khi mạng lớn hơn, số block tăng hoặc m tăng. Mặc

dù Rand index vẫn ở mức khá **(0.65–0.55)**, tuy nhiên aRand lại còn khá thấp **(0.29 → 0.11)** –> điều này cho thấy mô hình vẫn bị ảnh hưởng nhiều bởi *các yếu tố ngẫu nhiên*. MAD ở đây đo độ phân tán hoặc biến thiên của số cộng đồng ước lượng $\hat{K}$ quanh median, tức là khoảng cách trung vị của mỗi lần lặp so với median của 100 vòng lặp. Ví dụ, với (6,15), MAD = 1.0 (như trong bảng) cho thấy trong 50%+1 lần lặp, $\hat{K}$ bị sai lệch khoảng 1 so với median. MAD lớn hơn cho khi kích thước mạng tăng (12,20) cho thấy kết quả $\hat{K}$ hiệu suất mô hình ngày càng bất ổn và biến động mạnh hơn. Nhận định tổng quan thì các chỉ số đánh giá đều giảm khi kích thước mạng lớn, cho thấy mô hình gặp vấn đề trong việc dự đoán các tập dữ liệu với độ nhiễu cao hoặc kích thước lớn. Đề xuất khắc phục cần áp dụng thêm các phương pháp tunning parameters (sử dụng grid search, silhouette score) để tìm ra các bộ tham số (gamma, tau, theta) phù hợp cho từng bộ dữ liệu tương ứng, giúp nâng cao khả năng tổng quát của mô hình và detect nhiễu tốt hơn.

Chương 5

THỰC NGHIỆM VÀ PHÂN TÍCH KẾT QUẢ

5.1 Mô tả bộ dữ liệu

Trong phần này thực hiện phân tích dựa trên 2 tập dữ liệu là dữ liệu dữ liệu cổ phiếu có cấu trúc nhóm theo ngành (gợi ý bằng phân loại ngành nghề). Tập thứ hai là dữ liệu âm nhạc Spotify đồng nghệ sĩ và thể loại— không có cấu trúc nhóm rõ ràng.

5.1.1 Phân tích dữ liệu cổ phiếu

Để ứng dụng và minh họa cho hoạt động của phương pháp CIM đã phân tích thị trường chứng khoán ở thị trường Mỹ gồm từ 4 ngành có số lượng mã ngành nhiều nhất:

Bảng 5.1: Số mã cổ phiếu theo ngành

Ngành	**Số lượng cổ phiếu**
Finance	494
Technology Services	166
Health Technology	103
Electronic Technology	96

Mạng lưới các cổ phiếu được xây dựng bằng cách kết nối hai cổ phiếu nếu chúng cùng ngành và tương đồng về chỉ số Volume(khối lượng giao dịch),Market Cap(vốn hóa thị trường), Change(mức thay đổi theo phiên). Với mạng này, các phương pháp phân cụm truyền thống thường chia cổ phiếu theo ngành, tạo thành 4 nhóm nhưng trên thực tế khi áp dụng phương pháp CIM thì sẽ phân loại cổ phiếu theo **sức ảnh hưởng trong mạng lưới**, chứ không chỉ ngành.

5.1.1.1 Phân tích kết quả

Bảng 5.2: Hệ số ước lượng, sai số chuẩn và giá trị p từ hồi quy với EPS là biến phụ thuộc

Biến	(λ)	σ^2	**const**	**Close**	**Volume**	**Market Cap**	**EPS**	**Beta (1Y)**
est	(0.226, 0.713)	0.043	0.484	-0.000	0.000	-0.0	-0.001	0.001
s.e.	(0.016, 0.922)	—	0.011	0.000	0.000	0.0	0.001	0.010
p-value	(0.275, 0.693)	—	$< 10^{-3}$	0.076	0.699	0.8	0.147	0.925

Thông qua phương pháp CIM để đánh giá mức độ λ ảnh hưởng của 4 nhóm cổ phiếu, nhóm chúng em đã sửa dụng các biến tài chính bao gồm: Close(giá đóng cửa của cổ phiếu theo phiên),Volumne(Khối lượng giao dịch), Market cap(Vốn hóa thị trường), EPS(lợi nhuận trên mỗi cổ phiếu) và Beta(1Y) - hệ số beta đo lường mức độ biến động so với thị trường chung. Nhận thấy rằng các biến tài chính trong bộ dữ liệu lần này nhóm đã sử dụng là các biến phổ biến được xem xét nhiều để đánh giá hiệu suất cổ phiếuKhi đó thông qua bảng 5.4 trình bày hệ số ước lượng, sai số chuẩn và giá trị p tương ứng của từng đặc trưng, cùng với miền giá trị quan sát của Λ.

Kết quả hiện thỉ ở bảng trên, cho ta thấy rằng khoảng giá trị lambda dao động trong khoảng từ **(0.226,0.713** và hình thành 2 nhóm phản ánh mức độ ảnh hưởng của cố phiếu trong mạng lưới. Bên cạnh đó phương sai phần dư $\sigma^2 = 0.043$ phản ánh mức độ nhiễu khá thấp nhưng khi thống kê ở mức 1% với p$< 10^{-3}$ tuy thể hiện được tính tuyến tính tổng thể của các biến đầu vào so với Λ. Tuy vậy khi thống kê ở mức 5% thì các biến đầu vào riêng lẻ gần như chưa đủ để giải thích sự biến thiên của Λ. Điều này phản ánh rằng độ ảnh hưởng cổ phiếu trong mạng lưới có thể không chỉ phụ thuộc vào các đặc trưng tài nguyên sẵng có, mà còn chịu ảnh hưởng bởi các yếu tố phi tuyến và cấu trúc mạng phức tạp hơn khác hay hành vi đầu tư, biến động xã hội, chính trị. Như vậy sau khi thực nghiệm trên dữ liệu thực tế,chỉ số Λ từ mô hình CIM cung cấp cho nhóm góc nhìn mới mẻ hơn về tầm ảnh hưởng của một cổ phiếu trong cả một mạng lưới rộng lớn từ đó mở ra một hướng nghiên cứu mới mở rộng nghiên cứu cấu trúc mạng và động lực thị trường trong tương lại.

5.1.2 Phân tích dữ liệu âm nhạc

Đối với dữ liệu âm nhạc phi cấu trúc thì để minh họa cho hoạt động phương pháp CIM với dữ liệu hơn 500 bài hát random được lấy từ nền tảng SPotify thông qua API với 12 thể loại bài hát:

5.1.2.1 Phát hiện cộng đồng theo Thể loại

Khác với dữ liệu cổ phiếu thì dữ liệu âm nhạc lại có cấu trúc phức tạp hơn với cộng đồng tìm ẩn về thông tin thể loại âm nhạc (genre) đóng vai trò quan trọng trong quá trình nhận diện cộng đồng có tính tương đồng về mặt phong cách âm

Bảng 5.3: Số lượng bài hát theo thể loại

Thể loại	Số lượng bài hát
R&B	128
Pop	122
K-Pop	120
Indie	117
Hip-Hop	117
Disco	109
Electropop	109
Ballad	108
Rock	106
Synthwave	104
Dance	104
Soul	103

thanh. Tuy nhiên, dữ liệu thể loại thu thập từ nền tảng Spotify tồn tại nhiều bất cập với nguyên nhân chính là:

- Một số bài hát có **quá nhiều thể loại**, gây nhiễu cho quá trình phân tích.
- Một số bài hát lại không thể lấy được thông tin thể loại.

Do đó để đảm bảo tính nhất quán cho toàn bộ dữ liệu cũng như là chất lượng đầu vào cho mô hình CIM sau này, nhóm đã thực hiện quá trình phát hiện cộng đồng và từ đó điền khuyết và chuẩn hóa dữ liệu tốt hơn.

5.1.2.1.1 Phát hiện cộng đồng thể loại âm nhạc Như đã đề cập bên trên chương 3 trước hết ta xây dựng mạng các tính chất âm thanh, bài hát và thể loại, trong đó mỗi bài hát sẽ được liên kết với với thể loại và được chia 3 trường hợp có, có nhiều và chưa có thể loại và sử dụng thuật toán 3.3.5 **Louvain, Leiden** - thuật toán tốt nhất trong các thuật toán phát hiện cộng đồng đã đề cập

- Khi đó với những bài hát không có thông tin thể loại, có thể gán thể loại theo mức độ tương đồng âm nhạc thông qua các tính nhất gần với nhưng
- Còn với những bài hát có quá nhiều thể loại cũng sẽ lọc ra thể loại có mức độ tương đồng cao nhất theo tính chất âm thanh rồi điền lại một thể loại duy nhất

Sau khi hoàn tất các bước trên, toàn bộ tập dữ liệu Spotify sẽ có cột genre — thể loại đã được chuẩn hóa. Cột này được sử dụng làm đầu vào cho các bước kế tiếp trong mô hình Community Influence Model (CIM), chẳng hạn như phân cụm bài hát, xây dựng mạng lan truyền ảnh hưởng theo thể loại, và xác định các nút trung tâm trong từng cộng đồng thể loại.

5.1.2.2 Phân tích kết quả

Bảng 5.4: Kết quả hồi quy tuyến tính với λ là biến phụ thuộc

Biến	**est**	**s.e.**	**p-value**
λ	(0.193, 0.772)	(-0.034, 0.999)	(0.194, 0.767)
σ^2	0.051	—	—
const	0.529	0.070	$< 10^{-3}$
energy	-0.000	0.001	0.746
danceability	-0.001	0.001	0.060
happiness	0.001	0.000	0.108
acousticness	-0.000	0.000	0.992
instrumentalness	0.000	0.000	0.715
liveness	-0.001	0.001	0.195
speechiness	0.000	0.001	0.997
artist_followers	-0.000	0.000	0.931
playlist_followers	0.000	0.000	0.933
BPM	0.000	0.000	0.664

Nhận xét: Để đánh giá mức độ ảnh hưởng của đặc trưng âm nhạc đến hệ số ảnh hưởng Λ, nhóm chúng em đã thực hiện hồi quy tuyến tính với biến phụ thuộc là Λ và các biến độc lập bao gồm: năng lượng (energy), khả năng khiêu vũ (danceability), mức độ vui vẻ (happiness), tính mộc mạc (acousticness), tính nhạc cụ (instrumentalness), sự sống động (liveness), khả năng nói (speechiness), số lượng người theo dõi nghệ sĩ (artist_total_followers), số lượng người theo dõi playlist (playlist_num_followers), và tốc độ nhịp (BPM). Khi đó thông qua bảng 5.4 trình bày hệ số ước lượng, sai số chuẩn và giá trị p tương ứng của từng đặc trưng, cùng với miền giá trị quan sát của Λ. Kết quả cho thấy hầu hết các biến giải thích không có ý nghĩa thống kê ở mức 5%, ngoại trừ hệ số chặn (*const*) có giá trị $p < 10^{-3}$, cho thấy có mối quan hệ tuyến tính tổng thể giữa các đặc trưng và hệ số ảnh hưởng Λ. Các đặc trưng như energy, happiness,speechiness hay BPM tuy có hệ số ước lượng khác 0, nhưng đều không có ý nghĩa thống kê, với giá trị p-value lần lượt là **0.772, 0.108, 0.193 và 0.664**.

Đáng chú ý, khoảng giá trị của Λ quan sát được nằm trong khoảng **(0.193, 0.772)**, cho thấy có sự khác biệt về mức độ ảnh hưởng giữa các bài hát trong tập dữ liệu. Tuy nhiên, không có đặc trưng nào nổi bật đóng vai trò quyết định đến sự biến thiên của Λ, điều này có thể phản ánh rằng ảnh hưởng cộng đồng của một bài hát được hình thành từ nhiều yếu tố phi tuyến hoặc phụ thuộc vào các yếu tố mạng phức tạp khác ngoài các đặc trưng âm nhạc thuần túy. Tóm lại, mô hình CIM cho phép lượng hóa mức độ ảnh hưởng Λ của từng bài hát và cung cấp thông tin mô tả hữu ích. Tuy nhiên, các đặc trưng đầu vào trong tập dữ liệu hiện tại chưa đủ để giải thích rõ ràng sự thay đổi trong mức độ ảnh hưởng và từ đó tiếp tục hướng đi phân tích vai trò và tầm ảnh hưởng quan của mỗi cá nhân bài hát, thể loại, ca sĩ trong mạng lưới để có phương pháp mới ổn định và mở rộng thị trường âm nhạc

tránh bị bão hòa, khó tiếp cận người nghe.

Chương 6

Ứng dụng triển khai trực quan hóa

6.1 Quy trình xây dựng

6.2 Mô tả sử dụng

Chương 7

Appendix

7.1 Community influence model

Ta đã định nghĩa công thức CIM ở 4.1 như sau:

$$Y_i = \sum_{j=1}^{n} \lambda^{(g_j)} w_{ij} Y_j + X_i^\top \alpha + \epsilon_i,$$

hay

$$\mathbf{Y} = \mathbf{W}\Lambda\mathbf{Y} + X\alpha + \mathcal{E},$$

Với

$$\Lambda = \begin{bmatrix} x_1 & x_2 & 0 & \dots \\ 0 & \dots & & \end{bmatrix}$$

Để hiểu hơn lý do từ công thức trên được triển khai sang công thức Quasi Log-Likelihood:

$$S(\boldsymbol{\lambda})\mathbb{Y} = \mathbb{X}\boldsymbol{\alpha} + \mathcal{E},$$

Ta có:

$$\begin{aligned} Y &= W\Lambda Y + X\alpha + \mathcal{E} \\ \Rightarrow \mathcal{E} &= Y - W\Lambda Y - X\alpha \\ \Leftrightarrow \mathcal{E} &= (I_n - W\Lambda)Y - X\alpha \end{aligned}$$

Đặt $S(\lambda) = I_n - W\Lambda$

$$\Rightarrow \mathcal{E} = S(\lambda)Y - X\alpha \sim (0, \sigma^2 I_n)$$

Tương tự, ta được

$$\begin{aligned} Y &= S^{-1}(\lambda)(X\alpha + \mathcal{E}) \\ Y &\sim \mathcal{N}(S^{-1}(\lambda)X\alpha, [S^{-1}(\lambda)]\sigma^2[S^{-1}(\lambda)]^T) \\ X\alpha + \varepsilon &\sim \mathcal{N}(X\alpha, \sigma^2 I) \end{aligned}$$

7.1.1 Xây dựng hàm Quasi Log-Likelihood

Do $E \sim \mathcal{N}(0, \sigma^2 I_n)$, hàm mật độ xác suất của E là:

$$p(\mathcal{E}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}\mathcal{E}^\top \mathcal{E}\right).$$

Thay $\mathcal{E} = S(\lambda)Y - X\alpha$ vào, ta có:

$$p(Y \mid \alpha, \lambda, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(S(\lambda)Y - X\alpha)^\top (S(\lambda)Y - X\alpha)\right).$$

Lấy log-likelihood:

$$\ell_n(\alpha, \lambda, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(S(\lambda)Y - X\alpha)^\top (S(\lambda)Y - X\alpha).$$

Bỏ qua hằng số $-\frac{n}{2}\log(2\pi)$ không ảnh hưởng đến tối ưu hóa, ta có:

$$\ell_n(\alpha, \lambda, \sigma^2) = -\frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(S(\lambda)Y - X\alpha)^\top (S(\lambda)Y - X\alpha).$$

Vì $S(\lambda)$ thay đổi định thức, cần thêm điều chỉnh Jacobian:

$$\ell_n(\alpha, \lambda, \sigma^2) = -\frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(S(\lambda)Y - X\alpha)^\top (S(\lambda)Y - X\alpha) + \log|\det S(\lambda)|.$$

7.1.2 Xây dựng Objective Function

Thông thường hàm objective có dạng: objective=average_loss+regularizer. Khi đó các thành phần trong công thức sẽ là:

- *Hàm loss:* $\ell_n(\alpha, \lambda, \sigma^2)$ là log-likelihood của mô hình, tương ứng với phần "loss" trong tối ưu hóa.

- *Regularizer:* $\sum_{1 \le i < j \le n} p_\gamma(|\lambda_i - \lambda_j|)$ đóng vai trò như một hàm phạt để kiểm soát độ phức tạp của mô hình.

Để xác định cấu trúc nhóm của tham số ảnh hưởng λ, chúng tôi áp dụng một hàm phạt hợp nhất lồi-concave cho sự khác biệt từng cặp của λ. Khi đó, hàm *objective*

function được đề xuất có dạng:

$$Q_n(\theta) = \ell_n(\alpha, \lambda, \sigma^2) - \sum_{1 \le i < j \le n} p_\gamma(|\lambda_i - \lambda_j|),$$

trong đó $p_\gamma(\cdot)$ là một hàm phạt concave có tham số điều chỉnh $\gamma > 0$, và vector tham số đầy đủ được xác định bởi $\theta = (\alpha^\top, \lambda^\top, \sigma^2)^\top$.

7.1.3 QMLE Estimator

Từ hàm $Q_n(\theta)$, ta tiến hành ước lượng tham số bằng cách giải bài toán tối ưu:

$$\hat{\theta} = \arg\max_\theta Q_n(\theta).$$

Dưới giả định $\mathcal{E} \sim N(0, \sigma^2 I_n)$

$$Q_n(\alpha, \sigma^2, \lambda) = \ell_n(\alpha, \sigma^2, \lambda) - \sum_{1 \le i < j \le n} p_\gamma(|\lambda_i - \lambda_j|)$$

Mục tiêu: tối đa hoá $Q_n(\alpha, \sigma^2, \lambda)$ theo λ. Tức là ta đưa α, σ^2 theo λ, và tìm $\arg\max_\lambda Q_n(\lambda)$

7.1.3.1 Tìm ước lượng $\alpha(\lambda)$

Lấy đạo hàm Q_n theo α:

$$\begin{aligned}
\frac{\partial Q}{\partial \alpha} &= -\frac{1}{2\sigma^2}\left[\frac{\partial}{\partial \alpha}(S(\lambda)Y - X\alpha)^T(S(\lambda)Y - X\alpha) + (S(\lambda)Y - X\alpha)^T\frac{\partial}{\partial \alpha}(S(\lambda)Y - X\alpha)\right] \\
&= -\frac{1}{2\sigma^2}\left[-X^T(S(\lambda)Y - X\alpha) - (S(\lambda)Y - X\alpha)^T X\right] \\
&= -\frac{1}{2\sigma^2}\left[-2X^T(S(\lambda)Y - X\alpha)\right] \\
&= \sigma^2 X^T(S(\lambda)Y - X\alpha)
\end{aligned}$$

Đặt đạo hàm bằng 0:

$$\begin{aligned}
& \sigma^2 X^T(S(\lambda)Y - X\alpha) = 0 \\
\Leftrightarrow \quad & X^T S(\lambda)Y - X^T X\alpha = 0 \\
\Leftrightarrow \quad & X^T X\alpha = X^T S(\lambda)Y
\end{aligned}$$

Vậy:

$$\alpha = (X^T X)^{-1} X^T S(\lambda)Y \tag{7.1}$$

7.1.3.2 Tìm ước lượng $\sigma^2(\alpha(\lambda), \lambda)$

Lấy đạo hàm Q_n theo σ^2:

$$\frac{\partial Q}{\partial \sigma^2} = \frac{\partial \ell_n}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(S(\lambda)Y - X\alpha)^T(S(\lambda)Y - X\alpha)$$

Đặt đạo hàm bằng 0:

$$-\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(S(\lambda)Y - X\alpha)^T(S(\lambda)Y - X\alpha) = 0$$
$$\Rightarrow \quad \hat{\sigma}^2 = \frac{1}{n}(S(\lambda)Y - X\hat{\alpha})^T(S(\lambda)Y - X\hat{\alpha}) \qquad (*)$$

Thay $\hat{\alpha} = (X^TX)^{-1}X^TS(\lambda)Y$ vào (*), ta được:

$$\hat{\sigma^2} = \frac{1}{n}\{[I_n - X(X^TX)^{-}1X^T]S(\lambda)Y\}^T\{[I_n - X(X^TX)^{-}1X^T]S(\lambda)Y\} \qquad (**)$$

Đặt $M_X = I_n - X(X^TX)^{-1}X^T$. Vì M_X là ma trận chiếu, có tính chất đối xứng:

$$M_X^T = M_X, \quad M_X^2 = M_X$$
$$\Rightarrow M_X^T M_X = M_X$$

Từ (*), (**)

$$\hat{\sigma}^2 = \frac{1}{n}Y^TS(\lambda)^TM_X^TM_XS(\lambda)Y$$

Vậy:

$$\hat{\sigma}^2 = \frac{1}{n}Y^TS(\lambda)^TM_XS(\lambda)Y \qquad (7.2)$$

7.1.3.3 Chuyển đổi hàm Objective Function

Đặt $Q_{nc} = -Q_n$

$$Q_{nc} = \frac{n}{2}\log\sigma^2(\alpha(\lambda), \lambda) + \frac{1}{2\sigma^2(\hat{\alpha}(\lambda), \lambda)}\{S(\lambda)Y - X\alpha)\}^T\{S(\lambda)Y - X\alpha)\}$$
$$- \log|det\{S(\lambda)\}| + \sum_{1\le i<j\le} p_\gamma(|\lambda_i - \lambda_j|)$$

Mà khi nãy đạo hàm Q_n theo σ^2, ta có được:

$$\hat{\sigma}^2 = \frac{1}{n}(S(\lambda)Y - X\hat{\alpha})^T(S(\lambda)Y - X\hat{\alpha})$$

Thay $\hat{\sigma}^2$ vào công thức:

$$Q_{nc} = \frac{n}{2}\log\left(\frac{1}{n}(S(\lambda)Y - X\hat{\alpha})^T(S(\lambda)Y - X\hat{\alpha})\right)$$
$$+ \frac{1}{2\hat{\sigma}^2}(S(\lambda)Y - X\alpha)^T(S(\lambda)Y - X\alpha)$$
$$- \log|\det S(\lambda)| + \sum_{1\le i<j\le n} p_\gamma(|\lambda_i - \lambda_j|).$$

Tách logarit:

$$\log\left(\frac{1}{n}A\right) = \log A - \log n.$$

Sử dụng tính chất logarit, ta có:

$$\frac{n}{2}\log\left(\frac{1}{n}(S(\lambda)Y - X\hat{\alpha})^T(S(\lambda)Y - X\hat{\alpha})\right) = \frac{n}{2}\log(S(\lambda)Y - X\hat{\alpha})^T(S(\lambda)Y - X\hat{\alpha}) - \frac{n}{2}\log n.$$

Đồng thời:

$$\begin{aligned}
&\frac{1}{2\hat{\sigma}^2}(S(\lambda)Y - X\hat{\alpha})^\top(S(\lambda)Y - X\hat{\alpha}) \\
&= \frac{1}{2(S(\lambda)Y - X\hat{\alpha})^\top(S(\lambda)Y - X\hat{\alpha})} \cdot (S(\lambda)Y - X\hat{\alpha})^\top(S(\lambda)Y - X\hat{\alpha}) \\
&= \frac{n}{2}
\end{aligned}$$

Nên cuối cùng ta có thể viết lại là:

$$Q_{nc} = \frac{n}{2}\log\hat{\sigma}^2(\hat{\alpha}(\lambda), \lambda) - \log|\det\{S(\lambda)\}| + \sum_{1\le i<j\le n} p_\gamma(|\lambda_i - \lambda_j|) \tag{7.3}$$

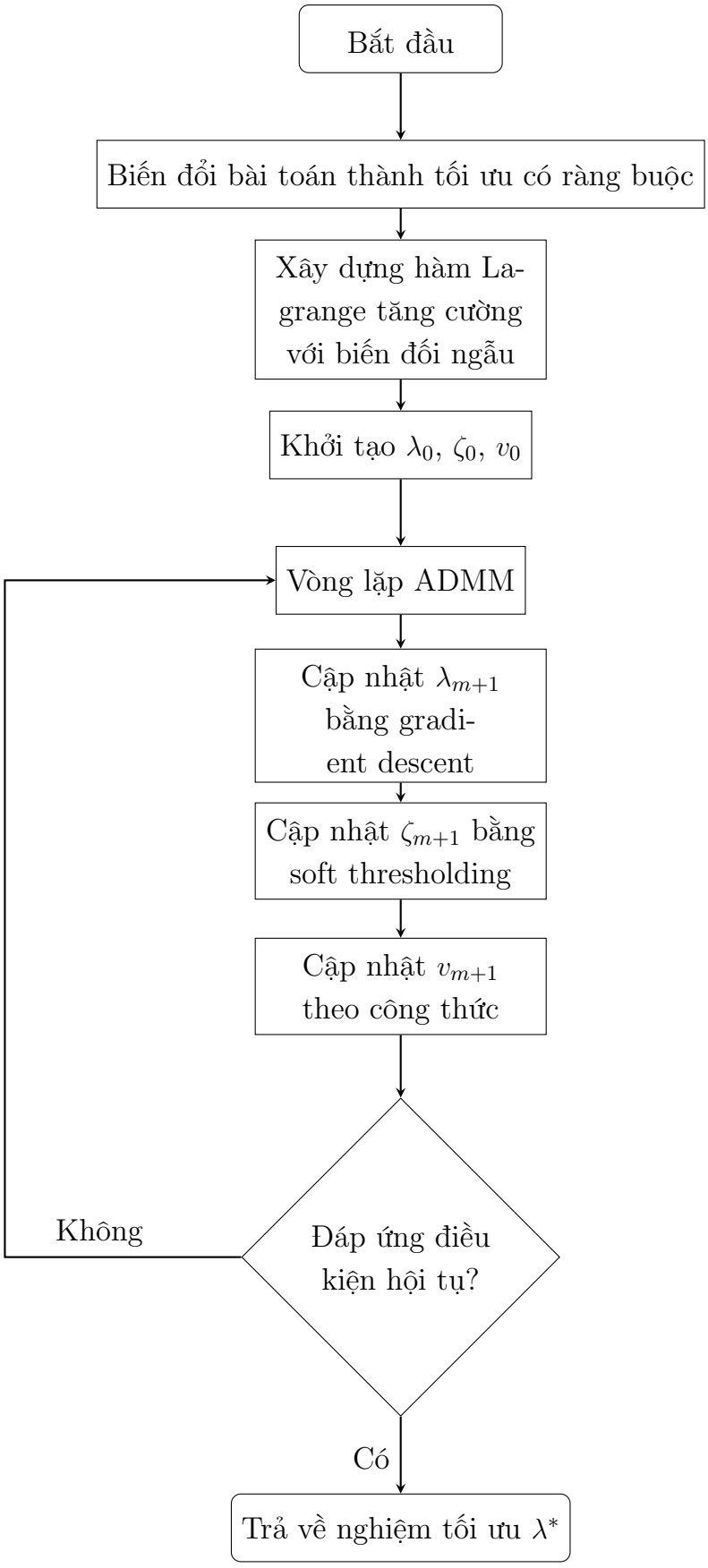
Lúc này, bài toán sẽ trở thành ước lượng tham số λ bằng cách tối thiểu hoá hàm Q_{nc}. Objective function:

$$\arg\min_{\lambda} Q_{nc}(\lambda)$$

Để giải quyết 7.3, sửa đổi thuật toán phương pháp nhân tử hướng xen kẽ "Alternating Direction Method of Multipliers" (ADMM) của Boyd et al. (2011) và Ma and Huang (2017) theo hai bước chính được trình bày dưới đây.

7.2 Tối ưu hóa hàm mục tiêu Q_{nc} bằng ADMM

ADMM được sử dụng để tối ưu hóa $Q_{nc}(\lambda)$ qua các bước sau:



Cụ thể các bước được trình bày sau đây

7.2.1 Bước 1

1. *Chuyển bài toán về dạng tối ưu có ràng buộc*
Ta bắt đầu với hàm mục tiêu từ phương trình 7.3

$$\arg\min_{\lambda} \frac{n}{2}\log\sigma^2 - \log|\det\{S(\lambda)\}| + \sum_{1\le i<j\le n} p_\gamma(|\lambda_i - \lambda_j|)$$

Nhưng do hàm phạt $p_\gamma(|\lambda_i - \lambda_j|)$ không trơn (non-smooth), việc giải bài toán trở nên khó khăn. Để xử lý, ta đưa bài toán về dạng tối ưu có ràng buộc bằng cách thêm biến phụ $\zeta_{ij} = \lambda_i - \lambda_j$. Khi đó hàm mục tiêu trở thành:

$$\arg\min_{\lambda,\zeta} \quad \frac{n}{2}\log\hat{\sigma}^2(\hat{\alpha}(\lambda),\lambda) - \log|\det(S(\lambda))| + \sum_{i<j} p_\gamma(|\zeta_{ij}|) \tag{7.4}$$

với ràng buộc bằng: $\lambda_i - \lambda_j - \zeta_{ij} = 0$, $\forall 1 \le i < j \le n$ (ζ_{ij} đóng vai trò là cầu nối, giúp ADMM có thể chia bài toán thành các bước nhỏ để dễ dàng giải quyết hơn)

2. *Xây dựng hàm Lagrangian tăng cường*
Hàm Lagrange tăng cường giúp chuyển đổi bài toán tối ưu có ràng buộc thành dạng không ràng buộc, có thể tìm điểm cực trị bằng đạo hàm thông thường. Phương pháp nhân tử Lagrange cổ điển bằng cách giải bài toán

$$\arg\min_{\lambda,\zeta} \quad \frac{n}{2}\log\hat{\sigma}^2(\hat{\alpha}(\lambda),\lambda) - \log|\det(S(\lambda))| + \sum_{i<j} p_\gamma(|\zeta_{ij}|) + \sum_{i<j} v_{ij}(\lambda_i - \lambda_j - \zeta_{ij})$$

Với $v = \{v_{ij}, 1 \le i < j \le n\}$, v_{ij} là các nhân tử Lagrange (hoặc các biến đối ngẫu - dual variables)

Tuy nhiên, phương pháp này gặp khó khăn trong hội tụ khi sử dụng các phương pháp số học để tìm nghiệm, đặc biệt là với các bài toán phi tuyến. Một cách tiếp cận khác là sử dụng hàm phạt (penalty function) thuần thuần tuý

$$\arg\min_{\lambda,\zeta} \quad \frac{n}{2}\log\hat{\sigma}^2(\hat{\alpha}(\lambda),\lambda) - \log|\det(S(\lambda))| + \sum_{i<j} p_\gamma(|\zeta_{ij}|) + \frac{\vartheta}{2}\sum_{i<j}(\lambda_i - \lambda_j - \zeta_{ij})^2$$

Với ϑ là tham số phạt được chỉ định trước, với giá trị dương nhỏ.

Cách này giúp hội tụ nhanh hơn. Nhưng vấn đề là nếu ϑ quá nhỏ thì ràng buộc sẽ không được thoả mãn tốt, ngược lại, nếu ϑ quá lớn thì bài toán trở nên khó tối ưu.

Do đó, ta đề xuất sử dụng hàm Lagrange tăng cường có được bằng cách kết hợp cả nhân tử Lagrange và hàm phạt. Các tham số ước tính có được bằng cách cực tiểu hàm mục tiêu Q_{ncc}

$$Q_{ncc}(\lambda, \zeta, v) = \frac{n}{2}\log\hat{\sigma}^2(\hat{\alpha}(\lambda), \lambda) - \log|\det(S(\lambda))| + \sum_{i<j} p_\gamma(|\zeta_{ij}|) + \sum_{i<j} v_{ij}(\lambda_i - \lambda_j - \zeta_{ij}) + \frac{\vartheta}{2}\sum_{i<j}(\lambda_i - \lambda_j - \zeta_{ij})^2 \tag{7.5}$$

7.2.2 Bước 2: Lặp lại các bước và cập nhật tham số

1. *Cập nhật* λ

$$\lambda^{(m+1)} = argmin_\lambda Q_{nc}(\lambda, \zeta^{(m)}, \vartheta^{(m)})$$

Việc cập nhật λ được thực hiện bằng phương pháp Gradient Descent, với

$$\lambda^{(m+1)} = \lambda^{(m)} - \eta \cdot \frac{\partial f}{\partial \lambda}$$

Với η là bước học (learning rate)

Tham số λ_i tại thời điểm tiếp theo phụ thuộc vào tham số λ_i^0 ban đầu và giá trị của $\frac{\partial f}{\partial \lambda_i}$. Cách tính hai tham số trên được trình bày dưới đây.

(a) *Tính đạo hàm* $\frac{\partial f(\lambda)}{\partial \lambda_i}$

Tính $f(\lambda)$

Bỏ qua các thành phần không phụ thuộc vào tham số λ trong Q_{ncc}, ta được công thức cho $f(\lambda)$

$$\begin{aligned} f(\lambda) &= \frac{n}{2}\log\sigma^2(\alpha(\lambda), \lambda) - \log|\det\{S(\lambda)\}| + \sum_{i<j} v_{ij}(\lambda_i - \lambda_j - \zeta_{ij}) + \frac{\vartheta}{2}\sum_{i<j}(\lambda_i - \lambda_j - \zeta_{ij})^2 \\ &= \frac{n}{2}\log\sigma^2(\alpha(\lambda), \lambda) - \log|\det\{S(\lambda)\}| + \sum_{i<j}\frac{\vartheta}{2}[(\lambda_i - \lambda_j - \zeta_{ij} + \frac{v_{ij}}{\vartheta})^2 - (\frac{v_{ij}}{\vartheta})^2] \end{aligned}$$

Vì $(\frac{v_{ij}}{\vartheta})^2$ là một hằng số, nên $\frac{\vartheta}{2}(\frac{v_{ij}}{\vartheta})^2$ cũng là một hằng số không phụ thuộc vào biến λ. Ta có thể viết lại hàm $f(\lambda)$ thành

$$f(\lambda) = \frac{n}{2}\log\sigma^2(\alpha(\lambda), \lambda) - \log|\det\{S(\lambda)\}| + \frac{\vartheta}{2}\sum_{i<j}(\lambda_i - \lambda_j - \zeta_{ij} + \frac{v_{ij}}{\vartheta})^2$$

Vì tổng các cặp (i, j) là tổng từng phần trong vec-tơ, nên:

$$f(\lambda) = \frac{n}{2}\log\sigma^2(\alpha(\lambda), \lambda) - \log|\det\{S(\lambda)\}| + \frac{\vartheta}{2}\left\|\Delta\lambda - \zeta + \vartheta^{-1}v\right\|_2^2$$

Vì là current state nên

$$f(\lambda) = \frac{n}{2}\log\sigma^2(\alpha(\lambda),\lambda) - \log|\det\{S(\lambda)\}| + \frac{\vartheta}{2}\left\|\Delta\lambda - \zeta^{(m)} + \vartheta^{-1}v^{(m)}\right\|_2^2 \tag{7.6}$$

Với $\Delta = \{(e_i - e_j), 1 \le i < j \le n\}^T$, $e_i s$ là vec-tơ $nx1$ với thành phần thư i bằng 1, các thành phần khác bằng 0.

Tính đạo hàm $\frac{\partial f(\lambda)}{\partial\lambda_i}$ Đạo hàm từng thành phần của $f(\lambda)$ theo λ_i

- Đạo hàm $\frac{n}{2}\log\sigma^2(\alpha(\lambda),\lambda)$
  có
  $$\sigma^2 = \frac{1}{n}Y^T S(\lambda)^T M_X S(\lambda) Y$$
  suy ra
  $$\log\sigma^2 = \log(Y^T S(\lambda)^T M_X S(\lambda) Y) - log(n)$$
  Sử dụng quy tắc đạo hàm:
  $$\frac{\partial}{\partial\lambda_i}\log f(\lambda) = \frac{1}{f(\lambda)}\frac{\partial}{\partial\lambda_i}f(\lambda)$$

  $$\begin{aligned}\frac{\partial}{\partial\lambda_i}(\frac{n}{2}\log\sigma^2) &= \frac{n}{2}\frac{1}{\sigma^2}\frac{\partial}{\partial\lambda_i}(Y^T S(\lambda)^T M_X S(\lambda) Y)\\ &= \frac{n}{2\sigma^2}[Y^T\frac{\partial S(\lambda)^T}{\partial\lambda_i}M_X S(\lambda)Y + Y^T S(\lambda)^T M_X\frac{\partial S(\lambda)}{\partial\lambda_i}Y]\\ &= \frac{n}{2\sigma^2}Y^T[\frac{\partial S(\lambda)^T}{\partial\lambda_i}M_X S(\lambda) + S(\lambda)^T M_X\frac{\partial S(\lambda)}{\partial\lambda_i}]Y\end{aligned}$$

  mà
  $$S(\lambda) = I_n - W\Lambda$$
  nên
  $$\frac{\partial S(\lambda)}{\partial\lambda_i} = \frac{\partial}{\partial\lambda_i}(I_n - W\Lambda) = -W_{ii}$$
  tương tự
  $$\frac{\partial S(\lambda)^T}{\partial\lambda_i} = \frac{\partial}{\partial\lambda_i}(I_n - \Lambda^T W^T) = -W_{ii}^T$$
  do đó
  $$\begin{aligned}\frac{\partial}{\partial\lambda_i}(\frac{n}{2}\log\sigma^2) &= \frac{n}{2\sigma^2}Y^T[-W^T M_X S(\lambda) - S(\lambda)^T M_X W]_{ii}Y\\ &= \frac{n}{2\sigma^2}Y^T[-2S(\lambda)^T M_X W]_{ii}Y\\ &= -\frac{n}{\sigma^2}Y^T[S(\lambda)^T M_X W]_{ii}Y\\ &= -\frac{n}{\sigma^2}YY^T[S(\lambda)^T M_X W]_{ii}\end{aligned} \tag{7.7}$$

- **Đạo hàm** $\log|\det\{S(\lambda)\}|$
  Áp dụng công thức: $d|U| = |U|.tr(U^{-1}dU)$

$$\begin{aligned}\frac{\partial}{\partial\lambda_i}\log|\det\{S(\lambda)\}| &= \frac{\partial}{\partial\lambda_i}\log(\det\{S(\lambda)\}) \\ &= \frac{\det\{S(\lambda)\}}{\det\{S(\lambda)\}}tr(S(\lambda)^{-1}\frac{\partial S(\lambda)}{\partial\lambda_i} \\ &= -tr(S(\lambda)^{-1}W_{ii}) \\ &= -[S(\lambda)^{-1}W]_{ii} \end{aligned} \tag{7.8}$$

- **Đạo hàm** $\frac{\vartheta}{2}\left\|\Delta\lambda - \zeta^{(m)} + \vartheta^{-1}v^{(m)}\right\|_2^2$

  Dựa trên công thức $d_{u_i}\sum_{i=1}^{n}(u_i)^2 = 2u_i^T u'$

$$\begin{aligned}\frac{\partial}{\partial\lambda_i}\frac{\vartheta}{2}\left\|\Delta\lambda - \zeta^{(m)} + \vartheta^{-1}v^{(m)}\right\|_2^2 &= \frac{\vartheta}{2}.2(\Delta\lambda - \zeta^{(m)} + \vartheta v^{(m)})_i^T\Delta \\ &= \vartheta\Delta^T[\Delta\lambda - \zeta^{(m)} + \vartheta v^{(m)}]_i \end{aligned} \tag{7.9}$$

Từ 7.7, 7.8, 7.9, ta có được

$$\frac{\partial f(\lambda)}{\partial\lambda_i} = -\frac{1}{\hat{\sigma}^2}\left[YY^\top S(\lambda)^\top M_X W\right]_{ii} + \left[S(\lambda)^{-1}W\right]_{ii} + \vartheta\left[\Delta^\top\left(\Delta\lambda - \zeta^{(m)} + \vartheta^{-1}v^{(m)}\right)\right]_i \tag{7.10}$$

Với $i = \overline{1,n}$, $[.]_{ii}$ là thành phần thứ (i,i) của ma trận, $[.]_i$ là thành phần thứ (i) của vec-tơ

Khi đó từ phươngg trình trên:

$$\lambda_i^{m+1} = (\Delta^\top\Delta)^{-1}\left(\frac{1}{\hat{\sigma}^2\vartheta}[YY^\top S(\lambda)^\top\mathcal{M}_X W]_{ii} - \frac{1}{\vartheta}[S(\lambda)^{-1}W]_{ii} + [\zeta^{(m)} - \vartheta^{-1}v^{(m)}]_i\right) \tag{7.11}$$

(b) **Tính** λ^0

Để tính giá trị ước lượng ban đầu λ^0,chúng ta cần lưu ý rằng giá trị này được thu được bằng cách tối thiểu hóa hàm log-likelihood giả âm $\ell_n(\alpha,\lambda,\sigma^2)$ trong phương trình 4.3. Để thực hiện việc tối thiểu hóa hàm $-\ell_n(\alpha,\lambda,\sigma^2)$, ta sẽ áp dụng một quy trình lặp. Trong quy trình này, chúng ta sẽ sử dụng các giá trị khởi tạo của các tham số bằng cách khớp mô hình tự hồi quy mạng với tất cả các λ_i được đặt bằng nhau.

Nói cách khác, là ta có thể dùng thư viện scipy.optimize.minimize trong python để tìm λ^0

2. Cập nhật ζ

(a) **Viết lại bài toán tối ưu Q_{ncc} theo ζ**

$$L(\zeta) = \sum_{i<j} p_\gamma(|\zeta_{ij}|) + \sum_{i<j} v_{ij}(\lambda_i - \lambda_j - \zeta_{ij}) + \frac{\vartheta}{2}\sum_{i<j}(\lambda_i - \lambda_j - \zeta_{ij})^2$$
$$= \sum_{i<j} p_\gamma(|\zeta_{ij}|) + v_{ij}(\lambda_i - \lambda_j - \zeta_{ij}) + \frac{\vartheta}{2}(\lambda_i - \lambda_j - \zeta_{ij})^2 \tag{7.12}$$

Nhắc lại công thức

$$ax + \frac{b}{2}x^2 = \frac{b}{2}(x^2 + \frac{2a}{b}x) = \frac{b}{2}[(x + \frac{a}{b})^2 - (\frac{a}{b})^2]$$

Với $a = v_{ij}, \quad b = \vartheta, \quad x = \lambda_i - \lambda_j - \zeta_{ij}$, ta biến đổi 7.12 thành

$$L(\zeta) = \sum_{i<j} p_\gamma(|\zeta_{ij}|) + \frac{\vartheta}{2}[(\lambda_i - \lambda_j - \zeta_{ij} - \frac{v_{ij}}{\vartheta})^2 - (\frac{v_{ij}}{\vartheta})^2]$$

Mà $\frac{\vartheta}{2}(\frac{v_{ij}}{\vartheta})^2$ không ảnh hưởng tới việc tối ưu hoá ζ, nên ta có thể bỏ qua.

$$\Rightarrow L(\zeta) = \sum_{i<j} p_\gamma(|\zeta_{ij}|) + \frac{\vartheta}{2}(\lambda_i - \lambda_j - \zeta_{ij} - \frac{v_{ij}}{\vartheta})^2$$

Và nếu tách riêng biến ζ_{ij} trong hàm trên, ta được

$$L(\zeta_{ij}^{m+1}) = p_\gamma(|\zeta_{ij}|) + \frac{\vartheta}{2}(\lambda_i - \lambda_j - \zeta_{ij} - \frac{v_{ij}}{\vartheta})^2$$

Trong ADMM, việc cập nhật tham số λ, ζ, v phải theo thứ tự. Tại vòng lặp thứ $m+1$, ta cập nhật ζ bằng cách giữ $\lambda^{(m+1)}$ và $v^{(m)}$ cố định, và tối ưu hóa L theo ζ.

$$\zeta_{ij}^{m+1} = \arg\min_{\zeta_{ij}}\{\frac{\vartheta}{2}(\lambda_i^{(m+1)} - \lambda_j^{(m+1)} - \zeta_{ij} - \vartheta^{-1}v_{ij}^{(m)})^2 + \rho_\gamma(|\zeta_{ij}|)\}$$

(b) **Giải bài toán tối ưu**

Đặt $u_{ij}^{(m+1)} = \lambda_i^{(m+1)} - \lambda_j^{(m+1)} + \vartheta^{-1}v_{ij}^{(m)}$. Ta viết lại:

$$\lambda_i^{(m+1)} - \lambda_j^{(m+1)} - \zeta_{ij} = u_{ij}^{(m+1)} - \vartheta^{-1}v_{ij}^{(m)} - \zeta_{ij}$$

Thay vào $L_{\zeta_{ij}}$, ta có:

$$L_{\zeta_{ij}} = \rho_\gamma(|\zeta_{ij}|) + \frac{\vartheta}{2}(u_{ij}^{(m+1)} - \vartheta^{-1}v_{ij}^{(m)} - \zeta_{ij})^2$$

Bỏ qua các hằng số không phụ thuộc vào ζ_{ij}, ta cần tối ưu:

$$f(\zeta_{ij}) = \rho_\gamma(|\zeta_{ij}|) + \frac{\vartheta}{2}(u_{ij}^{(m+1)} - \zeta_{ij})^2$$

(c) **Tối ưu hóa $f(\zeta_{ij})$ theo ζ_{ij}**
Để tìm $\zeta_{ij}^{(m+1)}$, ta lấy đạo hàm của $f(\zeta_{ij})$ theo ζ_{ij} và đặt bằng 0:

$$\frac{\partial f}{\partial \zeta_{ij}} = \rho'_\gamma(|\zeta_{ij}|) \cdot \text{sign}(\zeta_{ij}) - \vartheta(u_{ij}^{(m+1)} - \zeta_{ij}) = 0$$

trong đó:

- $\rho'_\gamma(|\zeta_{ij}|)$: Đạo hàm bậc nhất của hàm MCP:

$$\rho'_\gamma(|t|) = \gamma \left(1 - \frac{|t|}{\tau\gamma}\right)_+$$

với $(x)_+ = x$ nếu $x > 0$, và $(x)_+ = 0$ nếu $x \leq 0$. Cụ thể:

  - Nếu $|t| \leq \tau\gamma$, thì $\rho'_\gamma(|t|) = \gamma \left(1 - \frac{|t|}{\tau\gamma}\right)$.
  - Nếu $|t| > \tau\gamma$, thì $\rho'_\gamma(|t|) = 0$.

- $\text{sign}(\zeta_{ij})$: Dấu của ζ_{ij}, bằng 1 nếu $\zeta_{ij} > 0$, -1 nếu $\zeta_{ij} < 0$, và 0 nếu $\zeta_{ij} = 0$.

Phương trình trở thành:

$$\rho'_\gamma(|\zeta_{ij}|) \cdot \text{sign}(\zeta_{ij}) = \vartheta(u_{ij}^{(m+1)} - \zeta_{ij})$$

(d) **Giải phương trình theo hai trường hợp**
Trường hợp 1: $|\zeta_{ij}| > \tau\gamma$
Nếu $|\zeta_{ij}| > \tau\gamma$, thì $\rho'_\gamma(|\zeta_{ij}|) = 0$. Phương trình trở thành:

$$0 = \vartheta(u_{ij}^{(m+1)} - \zeta_{ij})$$

$$\zeta_{ij}^{(m+1)} = u_{ij}^{(m+1)}$$

Ý nghĩa: Khi sự khác biệt giữa λ_i và λ_j (được điều chỉnh bởi u_{ij}) quá lớn, hàm phạt MCP không áp dụng lực phạt, và ζ_{ij} chỉ đơn giản là u_{ij}.

Trường hợp 2: $0 < |\zeta_{ij}| \leq \tau\gamma$
Nếu $|\zeta_{ij}| \leq \tau\gamma$, thì $\rho'_\gamma(|\zeta_{ij}|) = \gamma \left(1 - \frac{|\zeta_{ij}|}{\tau\gamma}\right)$. Phương trình:

$$\gamma \left(1 - \frac{|\zeta_{ij}|}{\tau\gamma}\right) \text{sign}(\zeta_{ij}) = \vartheta(u_{ij}^{(m+1)} - \zeta_{ij})$$

Đặt $s = \text{sign}(\zeta_{ij})$, $z = |\zeta_{ij}|$, thì $\zeta_{ij} = sz$, và phương trình trở thành:

$$\gamma \left(1 - \frac{z}{\tau\gamma}\right) s = \vartheta(u_{ij}^{(m+1)} - sz)$$

Nhân cả hai vế với s:

$$\gamma\left(1-\frac{z}{\tau\gamma}\right)=\vartheta(su_{ij}^{(m+1)}-z)$$

Giải cho z:

$$\gamma-\frac{z}{\tau}=\vartheta su_{ij}^{(m+1)}-\vartheta z$$

$$\gamma-\vartheta su_{ij}^{(m+1)}=z\left(\frac{1}{\tau}-\vartheta\right)$$

$$z=\frac{\gamma-\vartheta su_{ij}^{(m+1)}}{\frac{1}{\tau}-\vartheta}$$

Vì $\zeta_{ij}=sz$, ta có:

$$\zeta_{ij}=s\cdot\frac{\gamma-\vartheta su_{ij}^{(m+1)}}{\frac{1}{\tau}-\vartheta}$$

Nhân tử và mẫu với -1:

$$\zeta_{ij}=\frac{\vartheta u_{ij}^{(m+1)}-s\gamma}{\vartheta-\frac{1}{\tau}}=\frac{\vartheta u_{ij}^{(m+1)}-s\gamma}{\vartheta\left(1-\frac{1}{\tau\vartheta}\right)}$$

(e) **Sử dụng toán tử Soft Thresholding (ST)**
Toán tử soft thresholding được định nghĩa:

$$\mathrm{ST}(t,\lambda)=\mathrm{sign}(t)(|t|-\lambda)_+$$

Nhận thấy $\vartheta u_{ij}^{(m+1)}-s\gamma$ có thể được viết lại:

$$\vartheta u_{ij}^{(m+1)}-s\gamma=\vartheta\left(u_{ij}^{(m+1)}-\mathrm{sign}(u_{ij}^{(m+1)})\frac{\gamma}{\vartheta}\right)$$

Nếu $s=\mathrm{sign}(u_{ij}^{(m+1)})$, thì:

$$\mathrm{ST}(u_{ij}^{(m+1)},\gamma/\vartheta)=\mathrm{sign}(u_{ij}^{(m+1)})(|u_{ij}^{(m+1)}|-\gamma/\vartheta)_+$$

Do đó:

$$\zeta_{ij}^{(m+1)}=\frac{\mathrm{ST}(u_{ij}^{(m+1)},\gamma/\vartheta)}{1-\frac{1}{\tau\vartheta}}$$

Kết hợp hai trường hợp, ta có công thức cập nhật $\zeta_{ij}^{(m+1)}$:

$$\zeta_{ij}^{(m+1)}=\begin{cases}u_{ij}^{(m+1)} & \text{nếu } |u_{ij}^{(m+1)}|>\tau\gamma,\\ \frac{\mathrm{ST}(u_{ij}^{(m+1)},\gamma/\vartheta)}{1-\frac{1}{\tau\vartheta}} & \text{nếu } |u_{ij}^{(m+1)}|\le\tau\gamma,\end{cases}\tag{7.13}$$

trong đó:

- $u_{ij}^{(m+1)} = \lambda_i^{(m+1)} - \lambda_j^{(m+1)} + \vartheta^{-1} v_{ij}^{(m)}$,
- $\mathrm{ST}(t, \lambda) = \mathrm{sign}(t)(|t| - \lambda)_+$.

3. Cập nhật ϑ

$$\vartheta_{ij}^{(m+1)} = \vartheta_{ij}^{(m)} + \vartheta(\lambda_i^{(m+1)} - \lambda_j^{m+1} - \zeta_{ij}^{(m+1)}) \tag{7.14}$$

4. Xét điều kiện để thuật toán hội tụ

Tính phần dư nguyên thủy (primal residual) r^{m+1} và phần dư đối ngẫu (dual residual) s^{m+1}:

$$r^{(m+1)} = \Delta\lambda^{(m+1)} - \zeta^{(m+1)}, s^{(m+1)} = -\vartheta\Delta^\top(\zeta^{(m+1)} - \zeta^{(m)})$$

Dừng thuật toán khi:

$$\|r^{(m+1)}\|_2 \leq \epsilon_p, \quad \|s^{(m+1)}\|_2 \leq \epsilon_d$$

Trong đó: ϵ_p và ϵ_d là các ngưỡng dừng.

Tài liệu tham khảo

V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.*, 10008:6, 2008. doi: 10.1088/1742-5468/2008/10/P10008.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. doi: 10.1561/2200000016.

U. et al. Brandes. On modularity clustering. *IEEE Trans. Knowl. Data Eng.*, 20: 172–188, 2008. doi: 10.1109/TKDE.2007.190689.

A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004. doi: 10.1103/PhysRevE.70.066111.

J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Phys. Rev. E*, 72:027104, 2005. doi: 10.1103/PhysRevE.72.027104.

S. Fortunato. Community detection in graphs. *Phys. Rep.*, 486:75–174, 2010. doi: 10.1016/j.physrep.2009.11.002.

R. Guimerà and L. A. Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900, 2005. doi: 10.1038/nature03288.

L. Hubert and P. Arabie. Comparing partitions. *J. Classif.*, 2:193–218, 1985.

P. Ji and J. Jin. Coauthorship and citation networks for statisticians. *Ann. Appl. Stat.*, pages 1779–1812, 2016.

J. Jin. Fast community detection by score. *Ann. Stat.*, 43:57–89, 2015.

Z.T. Ke, J. Fan, and Y. Wu. Homogeneity pursuit. *J. Am. Stat. Assoc.*, 110: 175–194, 2015.

E.D. Kolaczyk and G. Csárdi. *Statistical Analysis of Network Data with R*, volume 65. Springer, 2014.

W.T. Lai, R.B. Chen, Y. Chen, and T. Koch. Variational bayesian inference for network autoregression models. *Comput. Stat. Data Anal.*, 169:107406, 2022.

A. Lancichinetti and S. Fortunato. Community detection algorithms: a comparative analysis. *Phys. Rev. E*, 80:056117, 2009.

Clement Lee and Darren J Wilkinson. A review of stochastic block models and extensions for graph clustering. *Applied Network Science volume 4, Article number: 122 (2019)*, 49, 2019. doi: 10.48550/arXiv.1903.00114.

J. LeSage and R.K. Pace. *Introduction to Spatial Econometrics*. Chapman and Hall/CRC, 2009.

T. Li, E. Levina, and J. Zhu. Prediction models for network-linked data. *Ann. Appl. Stat.*, 13:132–164, 2019.

T. Li, L. Lei, S. Bhattacharyya, K. Van den Berge, P. Sarkar, P.J. Bickel, and E. Levina. Hierarchical community detection by recursive partitioning. *J. Am. Stat. Assoc.*, 117:951–968, 2022.

X. Lin and L.f. Lee. Gmm estimation of spatial autoregressive models with unknown heteroskedasticity. *J. Econom.*, 157:34–52, 2010.

S. Ma and J. Huang. A concave pairwise fusion approach to subgroup analysis. *J. Am. Stat. Assoc.*, 112:410–423, 2017.

C.F. Manski. Identification of endogenous social effects: the reflection problem. *Rev. Econ. Stud.*, 60:531–542, 1993.

E.M. Mohamed, T. Agouti, A. Tikniouine, and M. El Adnani. A comprehensive literature review on community detection: approaches and applications. In *Proc. Comput. Sci.*, volume 151, pages 295–302, 2019.

M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104, 2006. doi: 10.1103/PhysRevE.74.036104.

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, 2004. doi: 10.1103/PhysRevE.69.026113.

M.E.J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

K. Nowicki and T.A.B. Snijders. Estimation and prediction for stochastic blockstructures. *J. Am. Stat. Assoc.*, 96:1077–1087, 2001.

Naoto Ozaki, Hiroshi Tezuka, and Mary Inaba. A simple acceleration method for the louvain algorithm. *International Journal of Computer and Electrical Engineering*, 2016. doi: 10.17706/ijcee.2016.8.3.207-218.

M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Not. AMS*, 56:1082–1097, 2009.

J. Qian and L. Su. Shrinkage estimation of common breaks in panel data models via adaptive group fused lasso. *J. Econom.*, 191:86–109, 2016.

W.M. Rand. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.*, 66:846–850, 1971.

J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E*, 74:016110, 2006. doi: 10.1103/PhysRevE.74.016110.

Y. Ren, X. Zhu, X. Lu, and G. Hu. Graphical assistant grouped network autoregression model: a bayesian nonparametric recourse. *J. Bus. Econ. Stat.*, 42: 49–63, 2024.

J. Scott. *What Is Social Network Analysis?* Bloomsbury Academic, 2012.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67:91–108, 2005.

B. Wang, Y. Zhang, W.W. Sun, and Y. Fang. Sparse convex clustering. *Journal of Computational and Graphical Statistics*, 27:393–403, 2018.

Guodong Li. Yuewen Liu. Hansheng Wang. Xuening Zhu., Rui Pan. Network vector autoregression. *Ann. Statist. 45 (3) 1096 - 1123*, June 2017. doi: https://doi.org/10.1214/16-AOS1476.

Zhaoqun Yang, René Algesheimer, and Claudio J. Tessone. Improving the louvain algorithm for community detection with modularity maximization. *Physical Review E*, 94(1):012311, 2016. doi: 10.1103/PhysRevE.94.012311.