MẠNG XÃ HỘI

Bài 3. CỘNG ĐỒNG XÃ HỘI (tiếp theo)

ThS. Lê Nhật Tùng

Mục lục

Nội dung

3.2.4. Thuật toán dựa trên độ tương tự của Node (Node Similarity)

- Độ tương tự của node dựa trên các node láng giềng. Thuật toán này theo hướng Node-Centric Community.
- Hai node có cấu trúc tương tự nếu chúng có chung tập các node láng giềng

Độ tương tự Jaccard

$$Jaccard(v_i, v_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$$

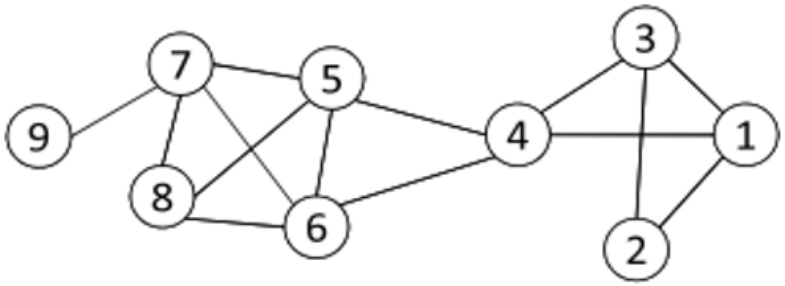
trong đó:

- N_i = Tập hợp các node láng giềng của node i
- N_j = Tập hợp các node láng giềng của node j
- $|N_i \cap N_j|$ = Số lượng node láng giềng chung
- $|N_i \cup N_j|$ = Tổng số node láng giềng duy nhất

Ý nghĩa:

- Giá trị nằm trong khoảng [0,1]
- Bằng 0: hai node không có node láng giềng chung
- Bằng 1: hai node có cùng tập node láng giềng
- Càng gần 1: cấu trúc láng giềng càng tương tự nhau
- Đo lường mức độ chồng lấp của các node láng giềng

Ví dụ: Độ tương tự Jaccard



Hình 3.1. Đồ thị để khám phá cộng đồng

Cho node 4 và node 6:

$$Jaccard(4,6) = \frac{|\{5\}|}{|\{1,3,4,5,6,7,8\}|} = \frac{1}{7}$$

Kết quả này cho thấy node 4 và node 6 chỉ có một node láng giềng chung trong tổng số bảy node láng giềng.

Độ tương tự Cosine

$$Cosine(v_i, v_j) = \frac{|N_i \cap N_j|}{\sqrt{|N_i||N_j|}}$$

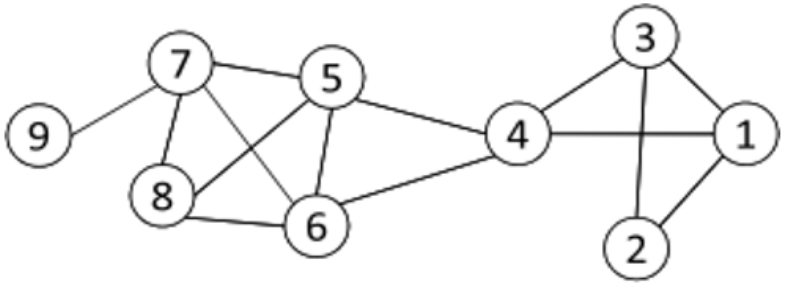
trong đó:

- N_i = Tập hợp các node láng giềng của node i
- N_j = Tập hợp các node láng giềng của node j
- $|N_i \cap N_j|$ = Số lượng node láng giềng chung
- $|N_i|$ = Số lượng node láng giềng của node i
- $|N_j|$ = Số lượng node láng giềng của node j

Ý nghĩa:

- Đo lường góc giữa hai vector láng giềng, không phụ thuộc vào kích thước tuyệt đối; Phù hợp khi so sánh các node có số lượng láng giềng khác biệt nhiều; Cho phép so sánh công bằng hơn giữa các node có độ lớn khác nhau so với Jaccard

Ví dụ: Độ tương tự Cosine



Hình 3.1. Đồ thị để khám phá cộng đồng

Cho node 4 và node 6:

$$Cosine(4,6) = \frac{1}{\sqrt{4 \cdot 4}}$$
$$= \frac{1}{4}$$

Kết quả này cho thấy độ tương tự cosine giữa node 4 và node 6 là 0.25, thể hiện mức độ tương đồng cấu trúc tương đối thấp.

Ứng dụng

Sau khi tính toán các độ đo tương tự:

- Ta thu được ma trận độ tương tự giữa tất cả các node
- Ma trận này có thể được sử dụng làm đầu vào cho các thuật toán phân cụm
- Thuật toán k-means có thể được áp dụng để khám phá các cụm node
- Các cụm đại diện cho các nhóm node có tính chất cấu trúc tương tự nhau

Độ tương tự dựa trên sự gần gũi của node

- Xét sự gần gũi giữa các node thông qua vai trò trung gian
- Đánh giá khả năng truyền tải thông tin giữa các node
- Xem xét đường đi qua các node láng giềng trung gian

Công thức tính độ tương tự

$$S_{ij} = \sum_{z \in T(i) \cap T(j)} \frac{1}{k(z)}$$

trong đó:

- $T(i)$ = Tập các node láng giềng của node i
- $T(j)$ = Tập các node láng giềng của node j
- z = Node chung thuộc cả $T(i)$ và $T(j)$
- $k(z)$ = Số bậc của node z
- $S_{ij} = 0$ khi node i không kết nối trực tiếp với node j

Ý nghĩa của độ đo

- Đo lường mức độ gần gũi thông qua các node trung gian
- Node trung gian có bậc cao sẽ đóng góp ít hơn vào độ tương tự
- Node trung gian có bậc thấp thể hiện kết nối chuyên biệt hơn
- Phản ánh khả năng truyền thông tin giữa hai node
- Càng có nhiều đường đi ngắn qua node trung gian, độ tương tự càng cao

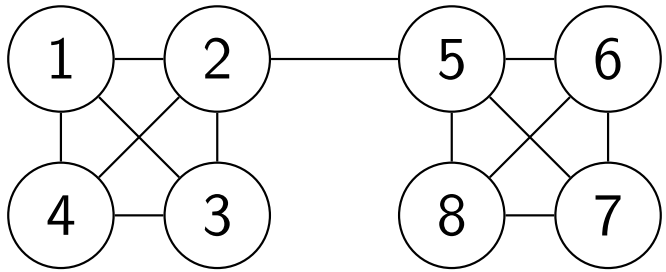
Ma trận độ tương tự

- Với mạng n node, tính S_{ij} cho mọi cặp node (i,j)
- Kết quả được lưu vào ma trận S kích thước n $\times$ n
- S_{ij} là độ tương tự giữa node i và node j
- Ma trận S đối xứng: $S_{ij} = S_{ji}$
- Giá trị S_{ij} phụ thuộc vào:
  - Số lượng node trung gian chung
  - Bậc của các node trung gian

Ví dụ: Tính ma trận độ tương tự S

- Cho mạng xã hội như Hình 3.9
- Tính độ tương tự giữa các node theo công thức:

$$S_{ij} = \sum_{z \in T(i) \cap T(j)} \frac{1}{k(z)}$$



Bước 1: Phân tích bậc của các node

- $k(1) = 3$: kết nối với 2, 3, 4
- $k(2) = 4$: kết nối với 1, 3, 4, 5
- $k(3) = 3$: kết nối với 1, 2, 4
- $k(4) = 3$: kết nối với 1, 2, 3
- $k(5) = 4$: kết nối với 2, 6, 7, 8
- $k(6) = 3$: kết nối với 5, 7, 8
- $k(7) = 3$: kết nối với 5, 6, 8
- $k(8) = 3$: kết nối với 5, 6, 7

Bước 2: Tính S_{12}

Xét node 1 và 2:

- $T(1) = \{2, 3, 4\}$
- $T(2) = \{1, 3, 4, 5\}$
- $T(1) \cap T(2) = \{3, 4\}$
- $k(3) = 3$
- $k(4) = 3$

$$S_{12} = \frac{1}{k(3)} + \frac{1}{k(4)} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

Bước 3: Tính S_{13}

Xét node 1 và 3:

- $T(1) = \{2, 3, 4\}$
- $T(3) = \{1, 2, 4\}$
- $T(1) \cap T(3) = \{2, 4\}$
- $k(2) = 4$
- $k(4) = 3$

$$S_{13} = \frac{1}{k(2)} + \frac{1}{k(4)} = \frac{1}{4} + \frac{1}{3} = \frac{7}{12}$$

Bước 4: Tính S_{14}

Xét node 1 và 4:

- $T(1) = \{2, 3, 4\}$
- $T(4) = \{1, 2, 3\}$
- $T(1) \cap T(4) = \{2, 3\}$
- $k(2) = 4$
- $k(3) = 3$

$$S_{14} = \frac{1}{k(2)} + \frac{1}{k(3)} = \frac{1}{4} + \frac{1}{3} = \frac{7}{12}$$

Bước 5: Tính S_{23}

Xét node 2 và 3:

- $T(2) = \{1, 3, 4, 5\}$
- $T(3) = \{1, 2, 4\}$
- $T(2) \cap T(3) = \{1, 4\}$
- $k(1) = 3$
- $k(4) = 3$

$$S_{23} = \frac{1}{k(1)} + \frac{1}{k(4)} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

Bước 6: Tính S_{24}

Xét node 2 và 4:

- $T(2) = \{1, 3, 4, 5\}$
- $T(4) = \{1, 2, 3\}$
- $T(2) \cap T(4) = \{1, 3\}$
- $k(1) = 3$
- $k(3) = 3$

$$S_{24} = \frac{1}{k(1)} + \frac{1}{k(3)} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

Bước 7: Tính S_{34}

Xét node 3 và 4:

- $T(3) = \{1, 2, 4\}$
- $T(4) = \{1, 2, 3\}$
- $T(3) \cap T(4) = \{1, 2\}$
- $k(1) = 3$
- $k(2) = 4$

$$S_{34} = \frac{1}{k(1)} + \frac{1}{k(2)} = \frac{1}{3} + \frac{1}{4} = \frac{7}{12}$$

Bước 8: Tính S_{56}

Xét node 5 và 6:

- $T(5) = \{2, 6, 7, 8\}$
- $T(6) = \{5, 7, 8\}$
- $T(5) \cap T(6) = \{7, 8\}$
- $k(7) = 3$
- $k(8) = 3$

$$S_{56} = \frac{1}{k(7)} + \frac{1}{k(8)} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

Bước 9: Tính S_{57}

Xét node 5 và 7:

- $T(5) = \{2, 6, 7, 8\}$
- $T(7) = \{5, 6, 8\}$
- $T(5) \cap T(7) = \{6, 8\}$
- $k(6) = 3$
- $k(8) = 3$

$$S_{57} = \frac{1}{k(6)} + \frac{1}{k(8)} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

Bước 10: Tính S_{58}

Xét node 5 và 8:

- $T(5) = \{2, 6, 7, 8\}$
- $T(8) = \{5, 6, 7\}$
- $T(5) \cap T(8) = \{6, 7\}$
- $k(6) = 3$
- $k(7) = 3$

$$S_{58} = \frac{1}{k(6)} + \frac{1}{k(7)} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

Bước 11: Tính S_{67}

Xét node 6 và 7:

- $T(6) = \{5, 7, 8\}$
- $T(7) = \{5, 6, 8\}$
- $T(6) \cap T(7) = \{5, 8\}$
- $k(5) = 4$
- $k(8) = 3$

$$S_{67} = \frac{1}{k(5)} + \frac{1}{k(8)} = \frac{1}{4} + \frac{1}{3} = \frac{7}{12}$$

Bước 12: Tính S_{68}

Xét node 6 và 8:

- $T(6) = \{5, 7, 8\}$
- $T(8) = \{5, 6, 7\}$
- $T(6) \cap T(8) = \{5, 7\}$
- $k(5) = 4$
- $k(7) = 3$

$$S_{68} = \frac{1}{k(5)} + \frac{1}{k(7)} = \frac{1}{4} + \frac{1}{3} = \frac{7}{12}$$

Bước 13: Tính S_{78}

Xét node 7 và 8:

- $T(7) = \{5, 6, 8\}$
- $T(8) = \{5, 6, 7\}$
- $T(7) \cap T(8) = \{5, 6\}$
- $k(5) = 4$
- $k(6) = 3$

$$S_{78} = \frac{1}{k(5)} + \frac{1}{k(6)} = \frac{1}{4} + \frac{1}{3} = \frac{7}{12}$$

Bước 14: Các giá trị đặc biệt

- $S_{16} = 0$: không có node trung gian chung
- $S_{25} = 0$: mặc dù có cạnh nối trực tiếp, nhưng không có node trung gian
- $S_{17} = 0$: không có node trung gian chung
- $S_{18} = 0$: không có node trung gian chung
- $S_{35} = 0$: không có node trung gian chung
- $S_{46} = 0$: không có node trung gian chung

Kết quả: Ma trận S

$$S = \begin{bmatrix} 0 & \frac{2}{3} & \frac{7}{12} & \frac{7}{12} & 0 & 0 & 0 & 0 \\ \frac{2}{3} & 0 & \frac{2}{3} & \frac{2}{3} & 0 & 0 & 0 & 0 \\ \frac{7}{12} & \frac{2}{3} & 0 & \frac{7}{12} & 0 & 0 & 0 & 0 \\ \frac{7}{12} & \frac{2}{3} & \frac{7}{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{2}{3} & \frac{2}{3} \\ 0 & 0 & 0 & 0 & \frac{2}{3} & 0 & \frac{7}{12} & \frac{7}{12} \\ 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{7}{12} & 0 & \frac{7}{12} \\ 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{7}{12} & \frac{7}{12} & 0 \end{bmatrix}$$

Ý tưởng thuật toán phát hiện cộng đồng

- Ý tưởng chính: Lặp di lặp lại việc hợp nhất cộng đồng có chứa một node với các cộng đồng có chứa node tương tự lớn nhất với node đó

Đầu vào và đầu ra

Đầu vào:

- Mạng xã hội $G(V, E)$
  - V là tập nút
  - E là tập cạnh
- Ma trận S đo độ tương tự của các nút trong mạng

Đầu ra:

- Tập các cộng đồng $V_1, V_2, ..., V_k$ (với $\bigcup_{i=1}^{k} V_i = V$)

Quá trình thực hiện

Bước 1: Khởi tạo

- Ban đầu, mỗi node là một cộng đồng
- Chọn một node bất kỳ làm node đầu tiên

Bước 2: Hợp nhất cộng đồng

- Hợp nhất cộng đồng chứa node hiện tại
- Với cộng đồng chứa node có độ tương tự lớn nhất
- Tạo thành cộng đồng mới

Quá trình thực hiện (tiếp)

Bước 3: Xác định node kế tiếp

- Chọn node có độ tương tự lớn nhất với node hiện tại
- Nếu node này không có trong cộng đồng hiện tại:
  - Thực hiện Bước 2
- Ngược lại:
  - Chọn ngẫu nhiên node mới chưa xét
  - Thực hiện Bước 2

Bước 4: Lặp lại

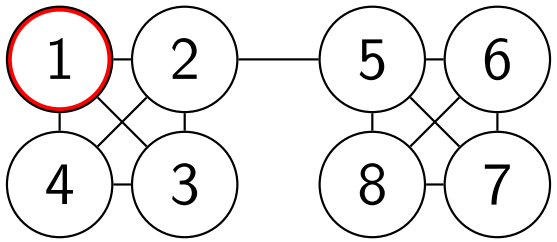
- Quay lại Bước 2 và Bước 3
- Cho đến khi không còn node nào chưa được xét

Bước 1: Khởi tạo

- Ban đầu có 8 cộng đồng riêng lẻ:

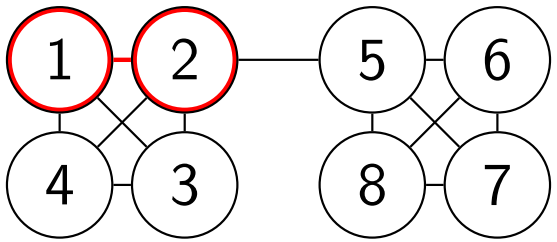
$$\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}$$

- Chọn ngẫu nhiên node 1 làm điểm bắt đầu



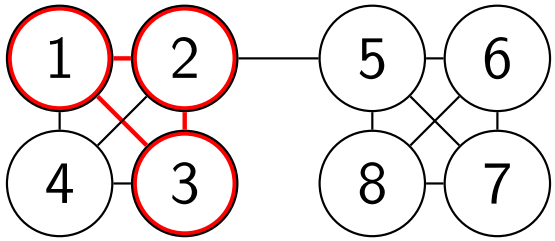
Bước 2: Hợp nhất cộng đồng đầu tiên

- Node 2 có độ tương tự cao nhất với node 1 ($S_{12} = \frac{2}{3}$)
- Hợp nhất cộng đồng $\{1\}$ và $\{2\}$ thành $\{1, 2\}$



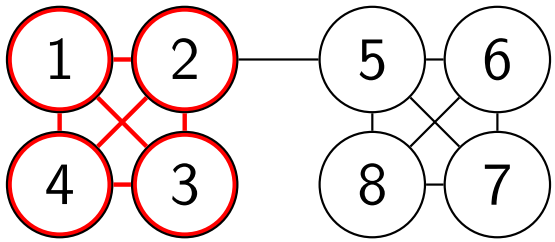
Bước 3: Mở rộng cộng đồng (phần 1)

- Node 1, 3, 4 có độ tương tự bằng nhau với node 2
- Chọn ngẫu nhiên node 3
- Node 3 không thuộc $\{1, 2\}$
- Hợp nhất thành cộng đồng $\{1, 2, 3\}$



Bước 3: Mở rộng cộng đồng (phần 2)

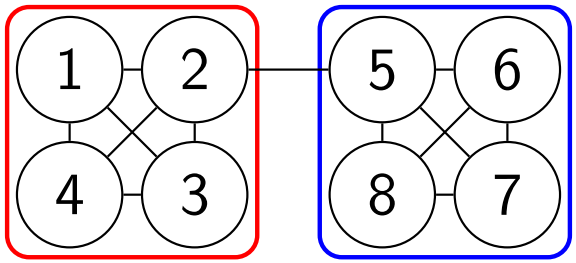
- Chọn ngẫu nhiên node 4 (do node 3 đã thuộc cộng đồng hiện tại)
- Node 2 có độ tương tự cao nhất với node 4
- Hợp nhất thành cộng đồng $\{1, 2, 3, 4\}$



Kết quả cuối cùng

- Thuật toán tiếp tục với node 5 và các node còn lại
- Kết thúc thuật toán, tìm được 2 cộng đồng:

$$\{1, 2, 3, 4\} \text{ và } \{5, 6, 7, 8\}$$



Độ phức tạp thuật toán

Chi phí tính toán:

- Chi phí tính độ tương tự
- Chi phí tìm kiếm các nút tiếp theo
- Chi phí hợp nhất cộng đồng

Phân tích độ phức tạp:

- Tính độ tương đồng cho k node láng giềng: $O(k)$
- Tính toán độ tương đồng cho mạng n node có k node láng giềng: $O(nk)$
- Không gian bộ nhớ cần thiết: $O(nk)$

3.2.5. Tổng quan về thuật toán LPA - phát hiện cộng đồng bằng lan truyền nhãn

Label Propagation Algorithm (LPA):

- Đề xuất bởi Raghavan et al. (2007)
- Ý tưởng chủ đạo:
  - Ban đầu mỗi node có một nhãn riêng biệt
  - Lan truyền nhãn qua các node láng giềng
  - Node chọn nhãn phổ biến nhất từ láng giềng
- Đặc điểm:
  - Đơn giản, hiệu quả
  - Tự động phát hiện số lượng cộng đồng

Nguyên lý hoạt động

- **Khởi tạo:**
  - Mỗi node nhận một nhãn duy nhất
  - $C_x(0) = x$ với mọi node x
- **Lan truyền:**
  - Nhãn được lan truyền qua mạng
  - Node cập nhật nhãn dựa trên láng giềng
  - Chọn nhãn phổ biến nhất trong láng giềng
- **Hội tụ:**
  - Các nhóm node mật độ cao đồng thuận về nhãn
  - Node cùng nhãn tạo thành cộng đồng

Hai cơ chế cập nhật nhãn

1. Cập nhật đồng bộ:

$$C_x(t) = f(C_{n_1}(t-1), ..., C_{n_k}(t-1))$$

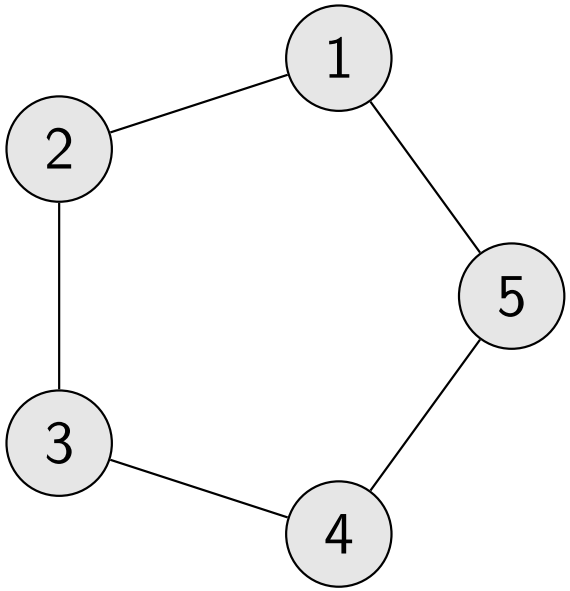
- Nhãn mới dựa trên nhãn láng giềng tại $t-1$
- Cập nhật đồng thời cho tất cả node

2. Cập nhật không đồng bộ:

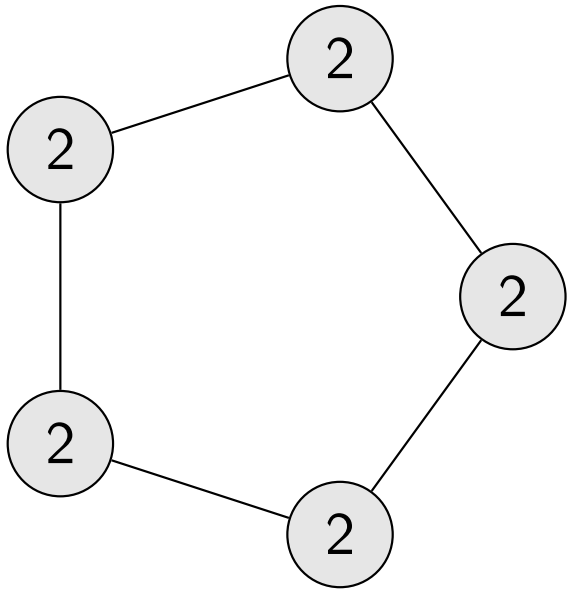
$$C_x(t) = f(C_{n_1}(t), ..., C_{n_p}(t), C_{n_{p+1}}(t-1), ..., C_{n_k}(t-1))$$

- Kết hợp nhãn tại t và $t-1$
- Cập nhật tuần tự theo thứ tự ngẫu nhiên

Quá trình lan truyền nhãn



Bước khởi tạo:
Mỗi node có nhãn riêng



Sau khi hội tụ:
Các node đồng thuận về nhãn 2

Thuật toán chi tiết

Input: Đồ thị $G(V, E)$

Các bước thực hiện:

1. **Khởi tạo:** $C_x(0) = x, \forall x \in V$
2. **Lặp:** $t = 1, 2, ...$ cho đến khi hội tụ
   - Sắp xếp ngẫu nhiên các node
   - Với mỗi node x:
     - $C_x(t) = \arg\max_l [f_l(x)]$
     - $f_l(x)$: tần suất nhãn l trong láng giềng của x

Output: Các nhóm node cùng nhãn = các cộng đồng

Điều kiện dừng và đặc điểm

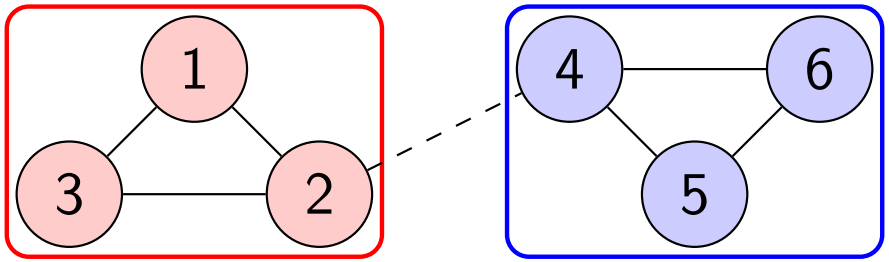
Điều kiện dừng:

- Mỗi node có nhãn phổ biến nhất của láng giềng
- Không có thay đổi nhãn trong một lần lặp

Đặc điểm quan trọng:

- Không cần định trước số cộng đồng
- Kết quả có thể thay đổi giữa các lần chạy
- Độ phức tạp gần tuyến tính với số cạnh
- Phù hợp với mạng lớn

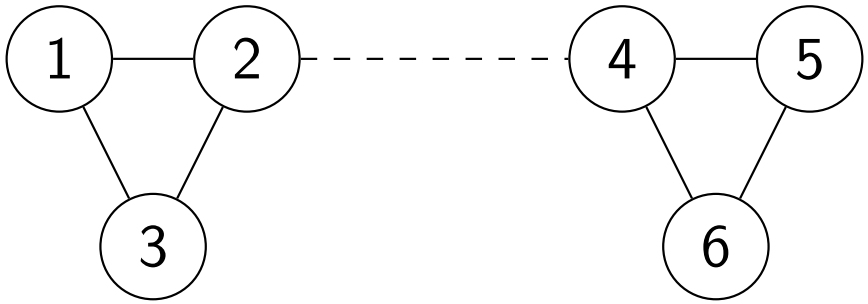
Ví dụ minh họa kết quả



Kết quả:

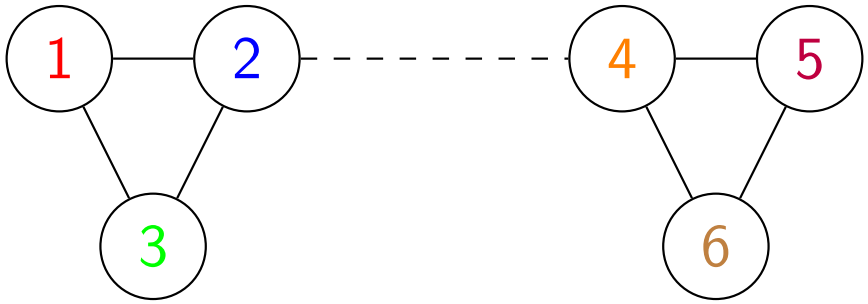
- Hai cộng đồng được phát hiện
- Node trong cùng cộng đồng có mật độ kết nối cao
- Kết nối giữa cộng đồng thưa thớt

Ví dụ minh họa - Đồ thị ban đầu



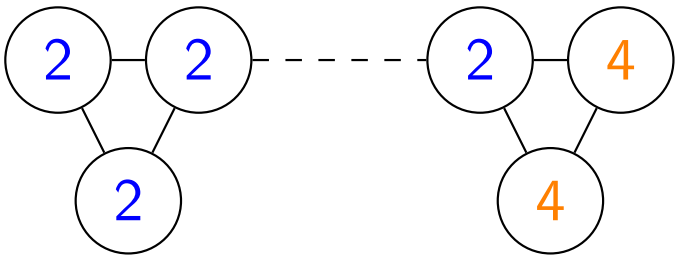
Đồ thị mẫu: 6 node với 2 cụm rõ ràng, có 1 cạnh nối giữa 2 cụm

Bước 1: Khởi tạo nhãn



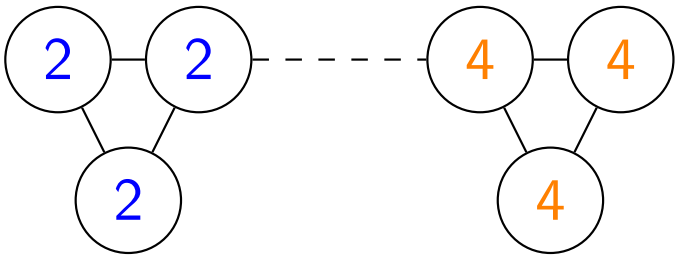
Ban đầu: Mỗi node được gán nhãn là chính số hiệu của nó
$C_x(0) = x$ với mọi node x

Bước 2: Lần lặp thứ nhất (t=1)



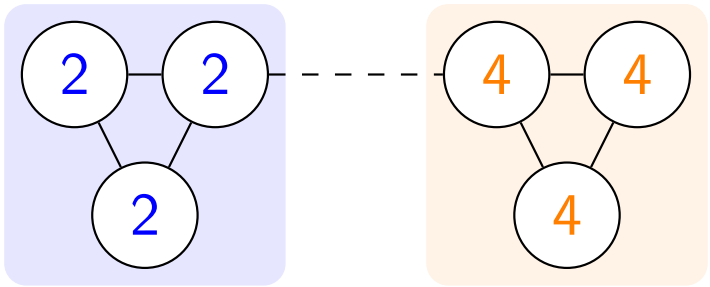
- Node 1 chọn nhãn 2 (láng giềng)
- Node 2 giữ nhãn 2 (phổ biến nhất)
- Node 3 chọn nhãn 2 (phổ biến nhất)
- Node 4 chọn nhãn 2 (láng giềng)
- Node 5, 6 chọn nhãn 4 (láng giềng)

Bước 3: Lần lặp thứ hai (t=2)



- Node 4 chuyển sang nhãn 4 vì:
  - Có 2 láng giềng nhãn 4
  - Chỉ 1 láng giềng nhãn 2
- Các node khác giữ nguyên nhãn
- Đã đạt trạng thái ổn định

Kết quả cuối cùng



Hai cộng đồng được phát hiện:

- Cộng đồng 1 (nhãn 2):
  - Nodes: 1, 2, 3
  - Màu xanh
- Cộng đồng 2 (nhãn 4):
  - Nodes: 4, 5, 6
  - Màu cam

Chúc các bạn học thật tốt!