ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN

Báo cáo đồ án
PHÂN TÍCH MẠNG XÃ HỘI
Môn học: Mạng xã hội

Lớp: IS353.O21

Giảng viên: ThS. Thái Bảo Trân

Thành viên (nhóm 8):

Trần Hoàng Phúc 21522479
Nguyễn Việt Hoàng 21522095
Lê Bá Nhất Long.....21522300

Thành Phố Hồ Chí Minh, tháng 06 2024

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Người nhận xét ký tên

MỤC LỤC

LỜI MỞ ĐẦU

Mạng xã hội (social network) đóng một vai trò quan trọng trong cuộc sống hiện đại với nhiều mục đích và lợi ích đa dạng. Một trong những mục đích chính của mạng xã hội là kết nối và giao tiếp, giúp mọi người duy trì và mở rộng các mối quan hệ cá nhân và nghề nghiệp, bất kể khoảng cách địa lý. Nó cung cấp nền tảng để trao đổi thông tin, tin nhắn, hình ảnh và video một cách nhanh chóng và dễ dàng. Còn cho phép người dùng chia sẻ thông tin và nội dung, như suy nghĩ, ý kiến, và tin tức hàng ngày, cũng như tạo và phân phối nội dung để thu hút sự chú ý và tương tác.

Về mặt kinh doanh và tiếp thị, mạng xã hội là công cụ mạnh mẽ để quảng bá sản phẩm, dịch vụ và thương hiệu, đồng thời tạo ra các kênh bán hàng trực tiếp, giúp tăng doanh số và tương tác với khách hàng. Đối với học tập và phát triển cá nhân, mạng xã hội cung cấp nền tảng trao đổi kiến thức và phát triển nghề nghiệp, giúp người dùng kết nối với các chuyên gia và tìm kiếm cơ hội việc làm. Nó cũng tăng cường tương tác cộng đồng thông qua việc tổ chức và tham gia các sự kiện, hoạt động cộng đồng và phong trào xã hội, cũng như hỗ trợ cho các hoạt động từ thiện và cứu trợ.

Sự phát triển mạnh mẽ của các mạng xã hội dẫn đến nguồn thông tin phong phú về mối quan hệ giữa người hoặc các thực thể với nhau.Tuy nhiên, nhiều tri thức trong đó lại thường được ẩn giấu bên trong, trong đó có thông tin về những đối tượng "có ảnh hưởng lớn" trong mạng xã hội. Biết được các đối tượng có ảnh hưởng lớn trong mạng xã hội sẽ giúp ích rất nhiều cho việc tiếp thị, quảng bá sản phẩm, cho phép thông tin và ý tưởng có thể đến với một số lượng lớn người sử dụng trong một thời gian ngắn. Tuy nhiên, bài toán tìm kiếm đối tượng có ảnh hưởng lớn một cách hiệu quả đem tới nhiều thách thức cần được giải quyết

Tối ưu hóa ảnh hưởng của đối tượng trong mạng xã hội là bài toán thời sự, có ý nghĩa. Có nhiều dự án nghiên cứu được hình thành để duy trì và phát triển hướng nghiên cứu này như thông tin lan truyền (Leskovec, 2007), tối ưu hóa ảnh hưởng rời rạc (Kempe, 2003), lan truyền trong tiếp thị (Richardson & Domingos) và lan truyền ảnh hưởng trong dịch tễ học (Wallinga & Teunis,2004). Gần đây, Wei Chen và các cộng sự (2009) đề xuất thuật toán cải tiến dựa trên thuật toán Greedy ban đầu – thuật toán DegreeDiscountIC, giảm bậc dựa trên kinh nghiệm và sau đó Manuel Gomez-Rodriguez và Bernhard Scholkopf, 2012, Bo Liu và cộng sự, 2012 tiến hành các nghiên cứu phát triển giải pháp nói trên

Đồ án "Phân tích mạng xã hội: Từ lý thuyết đồ thị đến ứng dụng" đề cập khái niệm về mạng xã hội và các phương pháp giải bài toán tối ưu hóa ảnh hưởng của đối tượng trong mạng xã hội, tập trung vào lớp giải pháp giảm bậc dựa trên kinh nghiệm.

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

I. Thông tin bài báo

- Tên bài báo: Social Network Analysis: From Graph Theory to Applications with Python (Phân tích mạng xã hội: Từ lý thuyết đồ thị đến ứng dụng)
- Năm xuất bản: 16/1/2021
- Tác giả: Dima Goldenberg
- Xuất bản tại: Trình bày tại PyCon'19 - Hội nghị Python của Israel 2019
- Link bài báo: https://towardsdatascience.com/social-network-analysis-from-theory-to-applications-with-python-d12e9a34c2c7
- Source code: https://github.com/dimgold/pycon_social_networkx
- Video trình bày tại hội nghị: https://youtu.be/px7ff2_Jeqw?si=7X5bDfFYTbnCK8FE

II. Mục đích bài báo

Lý thuyết đồ thị mạng xã hội nhằm mục đích nghiên cứu và phân tích các mối quan hệ và cấu trúc trong các mạng xã hội bằng cách sử dụng các khái niệm và công cụ của lý thuyết đồ thị. Lý thuyết này giúp mô hình hóa và biểu diễn các mối quan hệ xã hội dưới dạng các nút và cạnh, trong đó các nút đại diện cho các cá nhân hoặc tổ chức, và các cạnh biểu thị các mối quan hệ giữa họ. Ngoài ra, còn giúp phân tích cấu trúc mạng, xác định các nhóm, cộng đồng hoặc các cụm trong mạng xã hội, cũng như phát hiện các nút quan trọng có tầm ảnh hưởng lớn

Mục đích của tối đa hóa ảnh hưởng mạng xã hội (influence maximization) là tìm ra một tập hợp nhỏ các đối tượng trong mạng xã hội sao cho việc lan truyền thông tin từ các đối tượng này sẽ ảnh hưởng đến số lượng lớn người dùng nhất có thể. Cụ thể hơn, việc này nhằm tối ưu hóa lan truyền thông tin, đảm bảo rằng một thông điệp, sản phẩm hay ý tưởng được lan truyền rộng rãi và nhanh chóng trong mạng xã hội, từ đó tăng cường nhận thức thương hiệu và hiệu quả của các chiến dịch tiếp thị. Ngoài ra, tối đa hóa ảnh hưởng còn giúp tiết kiệm chi phí tiếp thị bằng cách tập trung vào các đối tượng có khả năng lan truyền cao, thay vì tiếp thị đại trà, và đạt được hiệu quả cao hơn trong các chiến dịch tiếp thị thông qua việc chọn đúng người có ảnh hưởng. Lý thuyết này cũng hỗ trợ phân tích cấu trúc mạng, dự đoán hành vi người dùng và sử dụng mạng xã hội để lan truyền các thông điệp tích cực hoặc kiểm soát sự lan truyền của bệnh dịch và thông tin sai lệch. Cuối cùng, tối đa hóa ảnh hưởng mạng xã hội còn đóng góp vào việc phát triển các mô hình toán học và thuật toán mới, cung cấp hiểu biết sâu sắc về cách con người tương tác và ảnh hưởng lẫn nhau trong mạng xã hội.

III. Mô tả bộ dữ liệu

Bộ dữu liệu chính được sử dụng cho các thuật toán là: **asoiaf-book5-edges.csv**

	A	B	C	D	E
1	Source	Target	Type	weight	book
2	Aegon-I-Targaryen	Daenerys-Targaryen	undirected	4	5
3	Aegon-Targaryen-(son-of-Rhaegar)	Daenerys-Targaryen	undirected	11	5
4	Aegon-Targaryen-(son-of-Rhaegar)	Elia-Martell	undirected	4	5
5	Aegon-Targaryen-(son-of-Rhaegar)	Franklyn-Flowers	undirected	3	5
6	Aegon-Targaryen-(son-of-Rhaegar)	Haldon	undirected	14	5
7	Aegon-Targaryen-(son-of-Rhaegar)	Harry-Strickland	undirected	9	5
8	Aegon-Targaryen-(son-of-Rhaegar)	Jon-Connington	undirected	16	5
9	Aegon-Targaryen-(son-of-Rhaegar)	Lemore	undirected	10	5
10	Aegon-Targaryen-(son-of-Rhaegar)	Rhaegar-Targaryen	undirected	9	5
11	Aegon-Targaryen-(son-of-Rhaegar)	Rhaenys-Targaryen-(daughter-of-Rhaegar)	undirected	4	5
12	Aegon-Targaryen-(son-of-Rhaegar)	Rolly-Duckfield	undirected	11	5
13	Aegon-Targaryen-(son-of-Rhaegar)	Tyrion-Lannister	undirected	23	5
14	Aegon-Targaryen-(son-of-Rhaegar)	Tywin-Lannister	undirected	3	5
15	Aegon-Targaryen-(son-of-Rhaegar)	Viserys-Targaryen	undirected	3	5
16	Aegon-Targaryen-(son-of-Rhaegar)	Yandry	undirected	6	5
17	Aegon-Targaryen-(son-of-Rhaegar)	Ysilla	undirected	4	5
18	Aemon-Targaryen-(Maester-Aemon)	Clydas	undirected	7	5
19	Aemon-Targaryen-(Maester-Aemon)	Gilly	undirected	3	5
20	Aemon-Targaryen-(Maester-Aemon)	Jon-Snow	undirected	12	5
21	Aemon-Targaryen-(Maester-Aemon)	Samwell-Tarly	undirected	16	5
22	Aemon-Targaryen-(Maester-Aemon)	Stannis-Baratheon	undirected	3	5
23	Aenys-Frey	Hosteen-Frey	undirected	6	5
24	Aenys-Frey	Theon-Greyjoy	undirected	3	5
25	Aenys-Frey	Wyman-Manderly	undirected	4	5
26	Aeron-Greyjoy	Euron-Greyjoy	undirected	4	5
27	Aeron-Greyjoy	Victarion-Greyjoy	undirected	3	5
28	Aerys-II-Targaryen	Rhaegar-Targaryen	undirected	5	5

Bộ dữ liệu miêu tả mỗi quan hệ của các nước trong mạng bình chọn

Source: là tên của 1 nhân vật trong sách

Target là tên của 1 nhân vật trong sách

Type: là vô hướng hoặc có hướng

Weight: là trọng số cạnh

Book: Nằm ở tập sách thứ 5

CHƯƠNG 2: LÍ THUYẾT ĐỒ THỊ

I. Các thành phần của mạng

Cạnh (Edges): biểu thị các kết nối giữa các nút và cũng có thể chứa các thuộc tính (chẳng hạn như trọng lượng biểu thị cường độ kết nối, hướng trong trường hợp quan hệ bất đối xứng hoặc thời gian nếu có).

Nút (Nodes): đại diện cho các thực thể trong mạng và có thể chứa các thuộc tính riêng (chẳng hạn như trọng lượng, kích thước, vị trí và bất kỳ thuộc tính nào khác) và các thuộc tính dựa trên mạng (chẳng hạn như Mức độ- số lượng hàng xóm hoặc Cụm- một thành phần được kết nối mà nút thuộc về ...).

⇨ Hai yếu tố cơ bản này có thể mô tả nhiều hiện tượng, chẳng hạn như kết nối xã hội, mạng định tuyến ảo, mạng điện vật lý, mạng lưới đường giao thông, mạng lưới quan hệ sinh học và nhiều mối quan hệ khác.

II. Mạng lưới thế giới thực (Real-world networks):

Real-world networks and in particular social networks have a unique structure which often differs them from random mathematical networks:

- Small world: các mạng thực thường có đường dẫn rất ngắn (về số lượng bước nhảy) giữa bất kỳ thành viên mạng nào được kết nối. Điều này áp dụng cho các mạng xã hội thực và ảo (lý thuyết sáu cái bắt tay) và cho các mạng vật lý như sân bay hoặc điện định tuyến lưu lượng truy cập web.
- Scale Free: có quy mô với sự phân bổ mức độ luật lũy thừa có dân số bị lệch với một vài nút có tính kết nối cao (chẳng hạn như những nút có ảnh hưởng xã hội) và rất nhiều nút có kết nối lỏng lẻo.
- Homophilly: là xu hướng các cá nhân liên kết và gắn bó với những người khác giống nhau, dẫn đến những đặc tính giống nhau giữa những người hàng xóm.

III. Độ đo trung tâm (Centrality Measures)

Các nút trung tâm cao đóng vai trò quan trọng của mạng, đóng vai trò là trung tâm cho các động lực mạng khác nhau. Tuy nhiên, định nghĩa và tầm quan trọng của tính trung tâm có thể khác nhau tùy từng trường hợp và có thể đề cập đến các biện pháp tính trung tâm khác nhau

1. Degree centrality

- Degree centrality của một đỉnh chính là tổng số các liên kết tới đỉnh đó trong đồ thị (tổng số cạnh kề của một đỉnh).
- Trường hợp đồ thị có hướng, degree centrality được tính bởi 2 giá trị: in-degree và out-degree.

  In-degree: tổng số liên kết từ các node khác đến node đang xét.

  Out-degree: tổng số liên kết từ node đang xét đến các node khác.

- Công thức tính degree centrality theo dạng chuẩn:

$$C'_D(u) = \frac{\sum_{v \epsilon V} e_{u,v}}{(n-1)}$$

Trong đó: **u**: đỉnh đang xét, **V**: tập đỉnh, **E**: tập cạnh, **n**: số đỉnh của đồ thị

- Degree centrality được dùng để xác định node nào có thể lan truyền thông tin nhanh, có khả năng gây ảnh hưởng trực tiếp đến các node xung quanh.

- Một thực thể có giá trị degree centrality cao:

  Là người hoạt động tích cực hoặc nổi tiếng nhất / Là một đầu nối quan trọng / Có một vị trí thuận lợi / Có tầm ảnh hưởng.

2. EigenVector

- Eigenvector (vector riêng) cho biết mức độ kết nối của một nút với các nút có nhiều kết nối khác.

- Cách tính Eigenvector

  Cho x là eigenvector của eigenvalue (trị riêng) lớn nhất λ của ma trận A của đồ thị vô hướng G = (V, E)

  Tìm eigenvalue λ bằng cách giải det(A – λI) = 0

  Thay λ vừa tìm được vào A – λI = B

  Giải: B[x] = [0]. Ta tìm được eigenvector x [8]

3. PageRank

Xếp hạng trang fanpage dựa theo mức độ thường xuyên của user nhấp vào một đường dẫn (tự nhiên) và tới trang fanpage.

- Công thức tính pagerank: [9]

$$PR(A) = (1 - d) + d(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)})$$

Trong đó: Chúng ta giả sử page A có các page T_1 và T_n trỏ tới nó

d là hệ số giảm xóc (damping factor) có thể đặt trong đoạn [0,1], thường được đặt là 0.85

C(T) là số lượng liên kết đi ra khỏi trang A

4. Closeness

- Closeness centrality là độ đo khoảng cách từ một đỉnh đến các đỉnh còn lại trong đồ thị.

- Công thức 1: Closeness cemtrality được tính bằng trị nghịch đảo của tổng số khoảng cách ngắn nhất từ một đỉnh đến tất cả các đỉnh còn lại của đồ thị

$$C_C(u) = \frac{1}{\sum_{v \in V} d(u,v)}$$

- Công thức 2: Closeness centrality được tính bằng bình quân của tổng số khoảng cách ngắn nhất từ một đỉnh đến tất cả các đỉnh còn lại.

$$C'_C(u) = \frac{n-1}{\sum_{v \in V} d(u,v)}$$

Trong đó: **d(u,v)** là đường đi ngắn nhất từ u đến v

- Một thực thể có giá trị closeness centrality tốt nhất(cao nhất):

Có thể truy xuất nhanh chóng đến các thực thể khác trong mạng

Có một đường đi ngắn nhất đến nhiều thực thể khác

5. Betweenness

- Betweenness centrality của một đỉnh được tính bằng tổng số các đường đi ngắn nhất ngang qua đỉnh đang xét chia cho tổng số các đường đi ngắn nhất của toàn mạng.
- Betweeness Centrality của một đỉnh u, ký hiệu là $C_B(u)$, độ đo này dùng để xem xét khả năng chi phối các quan hệ giữa các nút khác trong mạng.
- Khả năng u tham gia vào mối liên lạc hay các đường đi giữa s và t được tính như sau:

$$\delta_{st}(u) = \frac{\sigma_{st}(u)}{\sigma_{st}}$$

Trong đó: **$\sigma_{st}(u)$**: số đường đi ngắn nhất giữa *s* và *t* có chứa *u (s≠u≠ t),* **σ_{st}** : tổng số đường ngắn nhất giữa *s* và *t (s ≠ u ≠ t)*

- Công thức tính Betweenness Centrality của đỉnh u:

Cho đồ thị G = (V, E) có n đỉnh:

$$C_B(u) = \sum_{s \neq u \epsilon V} \sum_{t \neq u \in V} \delta_{st}(u)$$

- Công thức tính Betweenness Centrality theo dạng chuẩn

$$C'_B(v) = \frac{C_B(v)}{(n-1)(n-2)/2}$$

Miền giá trị của độ đo này nằm trong khoảng [0..1], node có giá trị càng lớn thì node đó sẽ có sự ảnh hưởng càng lớn đến việc phân bố cấu trúc của các cụm hay nhóm trong mạng càng lớn. Một tác nhân có vai trò trung tâm càng lớn trong mạng thì sẽ có tầm ảnh hưởng lớn trong việc kiểm soát mọi thông tin trao đổi giữa các tác nhân khác trong mạng.

➔ Các biện pháp khác nhau có thể hữu ích trong các tình huống khác nhau: xếp hạng(paper rank), phát hiện điểm quan trọng (betweeness), trung tâm vận chuyển (closeness) và các ứng dụng khác.

CHƯƠNG 3: MÔ HÌNH LAN TRUYỀN THÔNG TIN

I. Diffusion Information (Lan truyền thông tin)

1. Giới thiệu

Hầu hết MXH, các hoạt động tương tác như là: chia sẻ, đăng tin, đăng bài, biểu hiện cảm xúc, bình luận, vv… các hoạt động này gián tiếp lan truyền thông tin giữa mọi người trong MXH. Chắc chắn với khả năng truy cập LTTT, quy trình truyền lan này quy mô và tốc độ truyền lan rất nhanh. Để người có thể quản trị, điều hành, kiểm soát các thông tin này có tính hữu ích cao nhất thì cần phải nắm bắt và hiểu rõ quy trình này trên MXH. Để đạt được mục tiêu, quá trình truyền tải thông tin phải được mô tả một cách ngắn gọn dễ hiểu bằng mô hình truyền thông lan truyền thông tin (mô hình phổ biến thông tin).

2. Lan truyền là gì?

Quá trình lan truyền là quá trình mà thông tin được lan truyền từ nơi này đến nơi khác thông qua các tương tác. Đây là một lĩnh vực bao gồm các kỹ thuật từ nhiều khoa học và kỹ thuật từ các lĩnh vực khác nhau như xã hội học, dịch tễ học, và dân tộc học. Tất nhiên, mọi người đều quan tâm đến việc không bị lây nhiễm bởi một căn bệnh truyền nhiễm. Quá trình lan truyền bao gồm ba yếu tố chính như sau:

- Người gửi: Một người gửi (hoặc một nhóm người gửi) chịu trách nhiệm khởi đầu quá trình
- Người nhận: Một người nhận (hoặc một nhóm người nhận) nhận thông tin diffusion từ người gửi. Thông thường, số lượng người nhận cao hơn số lượng người gửi.
- Phương tiện: Đây là kênh thông qua đó thông tin diffusion được gửi từ người gửi đến người nhận. Đây có thể là TV, báo chí, mạng xã hội (ví dụ: một tweet trên Twitter), mối quan hệ xã hội, không khí (trong trường hợp quá trình lây lan bệnh), v.v.

3. Thành phần

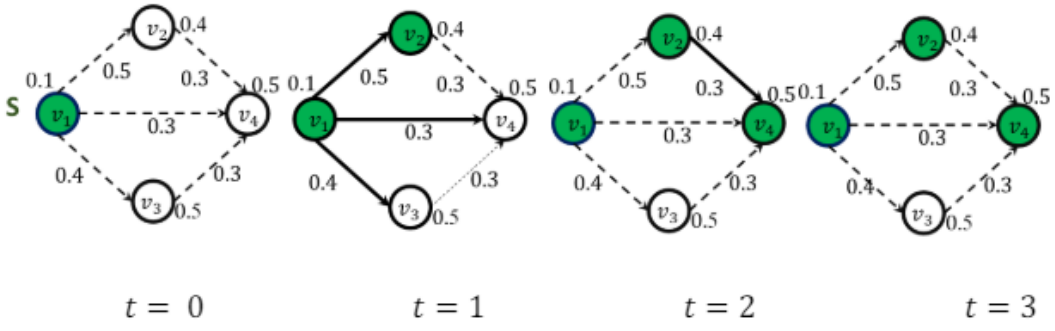
- *Trạng thái nú*t: Mỗi nút $v \in V$ tương ứng với hai trạng thái: 1) kích hoạt (active); 2) không kích hoạt (inactive). Nếu v ở trạng thái kích hoạt thì người dùng v chấp nhận thông tin truyền thông mới, ý tưởng mới được chia sẻ lan truyền từ một hoặc nhiều nút trong tập nút $N(v)$, đối với trạng thái không kích hoạt có nghĩa là chưa chấp nhận những thông tin truyền thông này, hay chưa tin tưởng về ý tưởng mới được chia sẻ đó. Trạng thái nút được thể hiện dưới nhiều mức độ, phụ thuộc vào từng MXH. Ví dụ: Chấp nhận email, chia sẻ bài viết, bình luận, biểu hiện cảm xúc, vv…
- *Quá trình thông tin được lan truyền*: Đối với lan truyền thông tin, quá trình này hoạt động tương ứng các bước thời gian rời rạc với thời gian t = 0, 1, ... Gọi $S \subseteq V$ là tập nút có trang thái kích hoạt, thì tập S gọi là tập nguồn hay tập hạt giống,

là tập nút phát tán thông tin đầu tiên hay là tập bị kích hoạt đầu tiên. Tập nút này đại diện cho những người dùng ban đầu được chọn để lan truyền ảnh hưởng hoặc đại cho những người dùng đầu tiên phát tán TTSL. Kết thúc của quá trình này là khi không còn kích hoạt thêm sau mỗi bước lan truyền.

II. Independent Cascade Model

- Mô hình Lan Truyền Độc Lập (ICM) giả định rằng các bước thời gian khuếch tán là rời rạc. Tại bất kỳ thời điểm nào, một nút có thể ở trạng thái hoạt động (tức là đã bị ảnh hưởng) hoặc không hoạt động. Một nút hoạt động có thể cố gắng kích hoạt một nút không hoạt động lân cận chỉ một lần, và một nút không thể trở lại trạng thái không hoạt động sau khi nó đã hoạt động (tức là một mô hình tiến triển)
- Quá trình lan truyền thông tin dọc theo các cạnh một cách độc lập chính là đặc trưng chính của mô hình này. Mỗi nút chưa bị kích hoạt thông tin sẽ bị kích hoạt một cách độc lập bởi từng nút lân cận đã nhiễm thông tin với một xác suất xác định. Khác với mô hình LT, Mỗi nút trên mô hình IC chỉ có một cơ hội duy nhất kích hoạt một nút khác. Mô hình này thường được dùng trong dự báo và nghiên cứu ảnh hưởng.
- Trong mô hình IC, mỗi cạnh $(u, v) \in E$ được gán một xác suất ảnh hưởng $p(u, v) \in [0, 1]$ biểu diễn mức độ ảnh hưởng của nút u với nút v. Nếu $(u, v) \notin E$, thì $p(u, v) = 0$. Mỗi nút cũng chỉ có thể nhận một trong hai trạng thái kích hoạt hoặc không kích hoạt. Gọi $\mathcal{D}^t(G, S)$ là tập các nút bị kích hoạt bởi S tại thời điểm t trên đồ thị G, quá trình lan truyền theo các bước rời rạc như sau:
- Tại thời điểm $t = 0$, tất cả các nút trong tập nguồn $S = \mathcal{D}^0(G, S)$ đều có trạng thái kích hoạt.
- Tại thời điểm $t \geq 1$, mỗi nút $u \in \mathcal{D}^{t-1}(G, S)$ có một cơ hội duy nhất kích hoạt đến nút $v \in N(u)$ với xác suất thành công là $p(u, v)$. Biến cố này có thể được thực hiện bằng cách áp dụng phép thử Bernoulli (Phép tung đồng xu độc lập) với xác suất thành công là $p(u, v)$. Nếu thành công ta thêm v và tập $\mathcal{D}^t(G, S)$ và nói rằng u kích hoạt v tại thời điểm t. Nếu nhiều nút kích hoạt v tại thời điểm t, kết quả tương tự xảy ra, v được thêm vào tập $\mathcal{D}^t(G, S)$. Một nút ở trạng thái kích hoạt, nó sẽ giữ nguyên trạng thái. Quá trình lan truyền kết thúc khi giữa hai bước không có nút nào bị kích hoạt thêm

Ví dụ



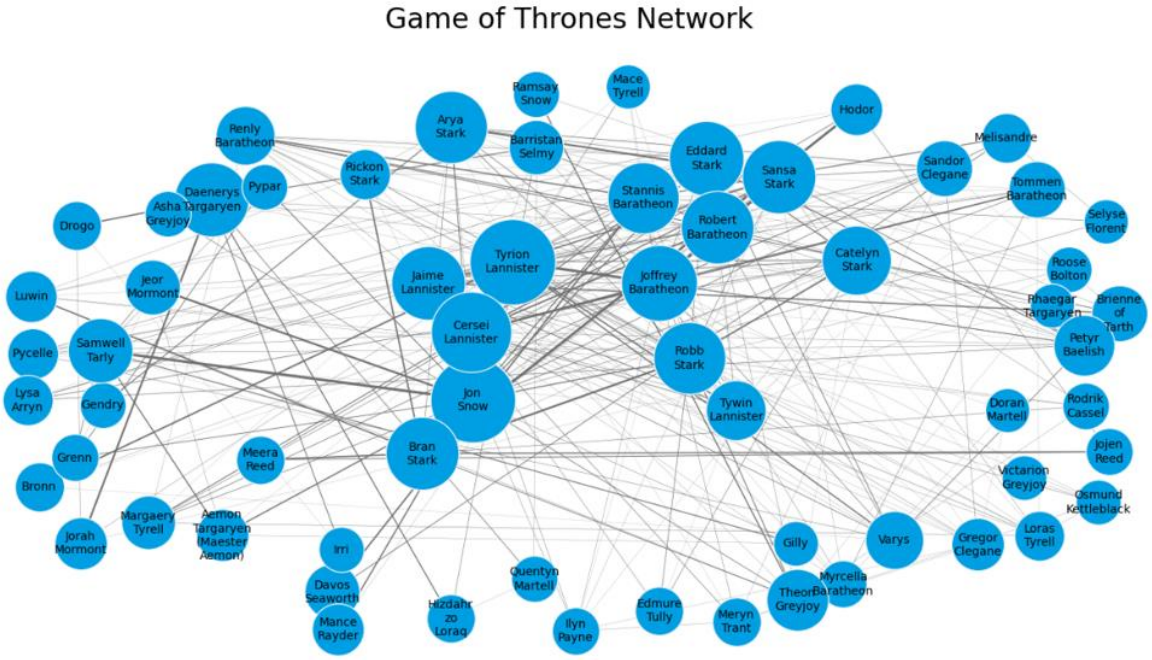
Một MXH cho bởi đồ thị $G(V, E)$, tập nút nguồn $S = \{v1\}$, các nút màu xanh là nút có trạng thái kích hoạt thông tin, các nút không màu là nút có trạng thái không kích hoạt thông tin. Cạnh nét liền thể hiện một nút đang cố gắng kích hoạt nút lân cận. Quá trình LTTT theo mô hình IC được thực hiện như sau:

- Tại thời điểm $t = 0$, nút nguồn $v1$ có trạng thái kích hoạt;
- Tại thời điểm $t = 1$, nút $v1$ kích hoạt các nút lân cận gồm $\{v2, v3, v4\}$, trong đó chỉ thành công với nút $v2$;
- Tại thời điểm $t = 2$, nút $v2$ kích hoạt thành công nút $v4$;
- Tại thời điểm$t = 3$, quá trình lan truyền kết thúc vì không có nút nào bị kích hoạt thêm.

Triển khai thuật toán bằng ngôn ngữ python:

```
def independent_cascade_of_author(G,t,infection_times):
    #doing a t->t+1 step of independent_cascade simulation
    #each infectious node infects neigbors with probabilty proportional to the weight
    max_weight = max([e[2]['weight'] for e in G.edges(data=True)])
    current_infectious = [n for n in infection_times if infection_times[n]==t]
    for n in current_infectious:
        for v in G.neighbors(n):
            if v not in infection_times:
                if G.get_edge_data(n,v)['weight'] >= np.random.random()*max_weight:
                    infection_times[v] = t+1
    return infection_times
```

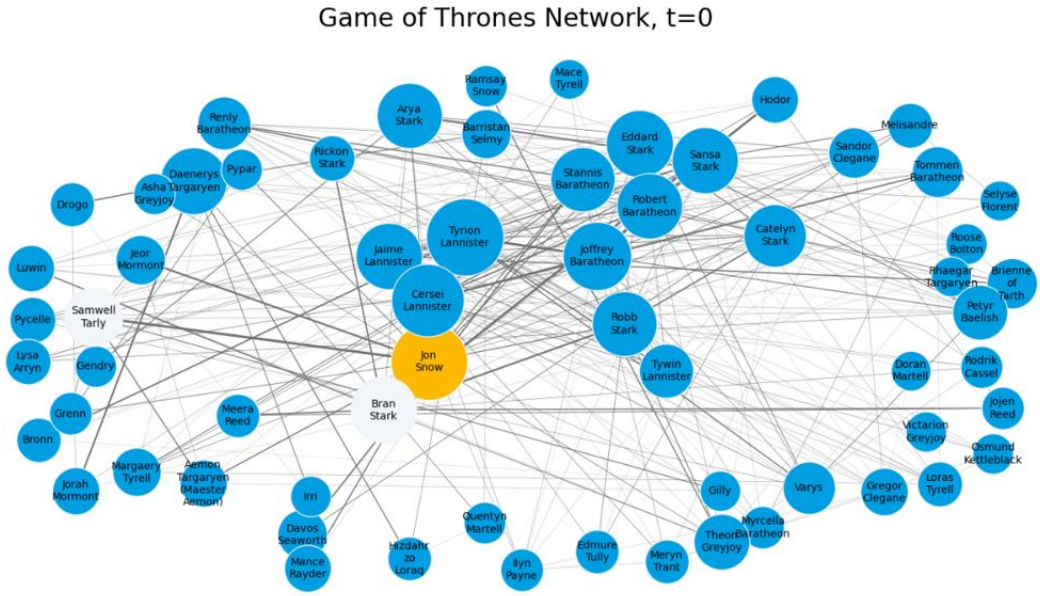
- **Ứng dụng với bộ dữ liệu:**
  Bộ dữ liệu ban đầu – Các node chưa bị nhiễm (màu xanh), đã bị nhiễm (màu trắng), bị nhiễm ở thời gian T(màu vàng)



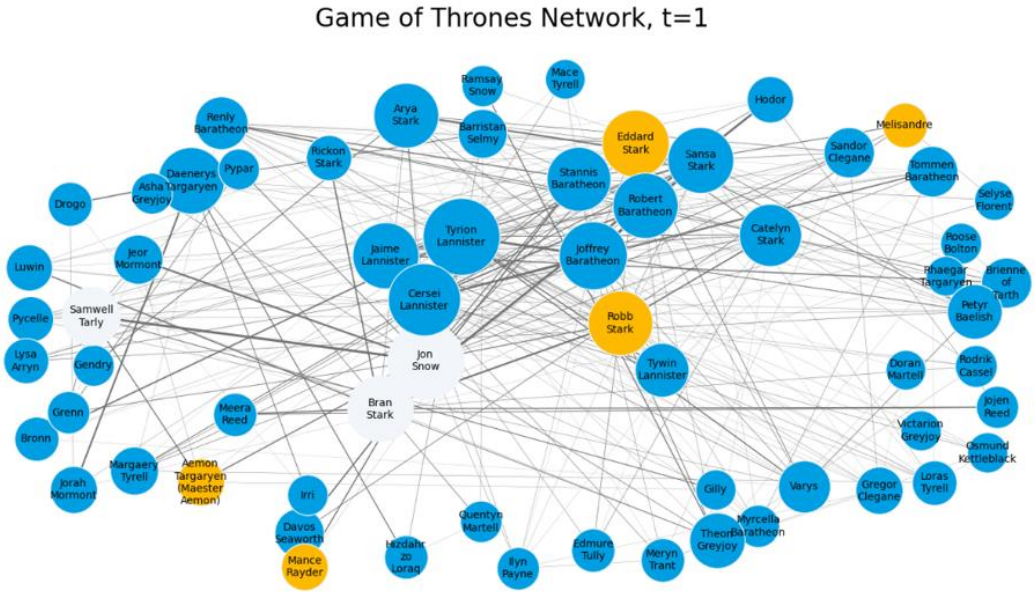
Giả sử các seed ban đầu bị nhiễm bệnh

```
infection_times = {'Bran-Stark':-1,'Samwell-Tarly':-1,'Jon-Snow':0}
```

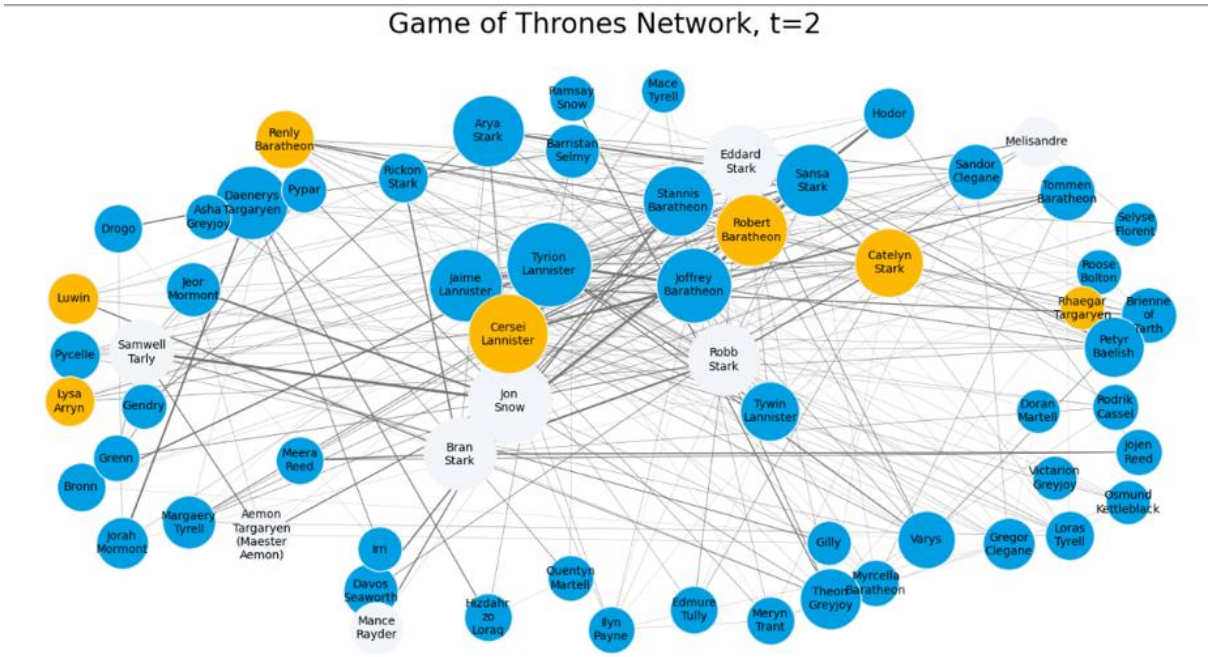
Ở thời điểm t=0



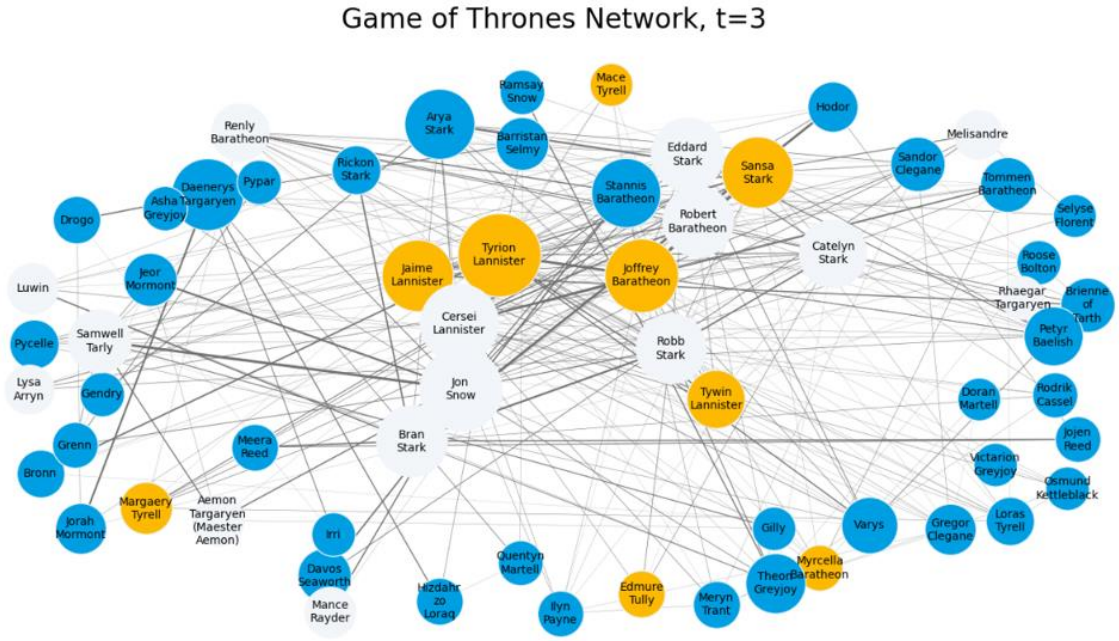
Ở thời điểm t=1



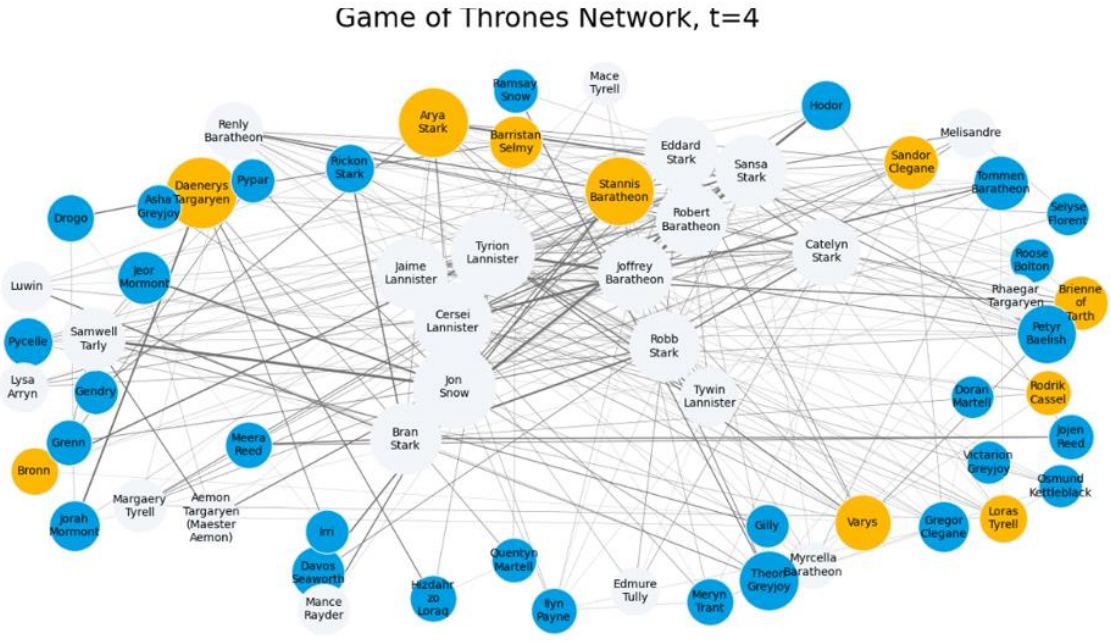
Ở thời điểm t=2



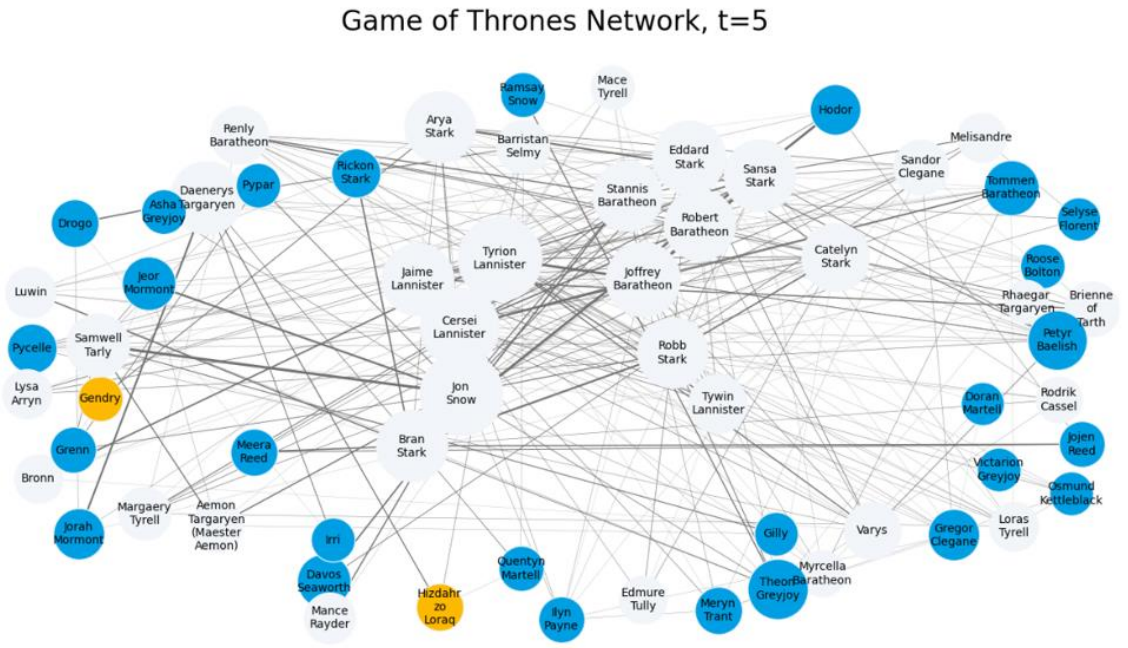
Ở thời điểm t=3



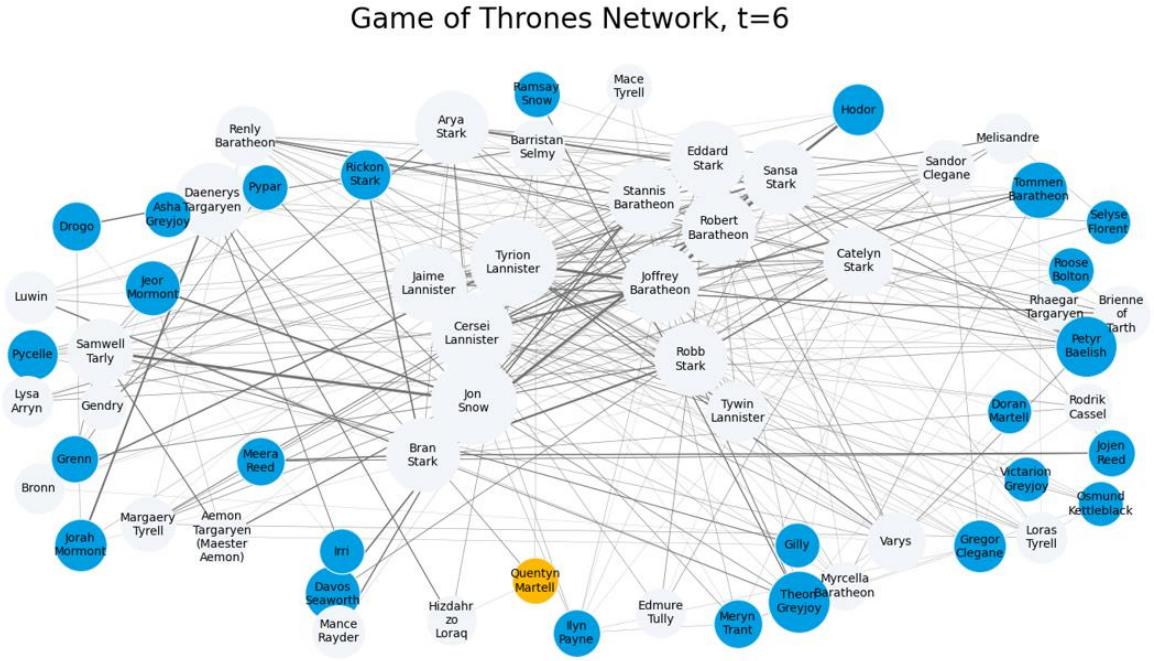
Ở thời điểm t=4



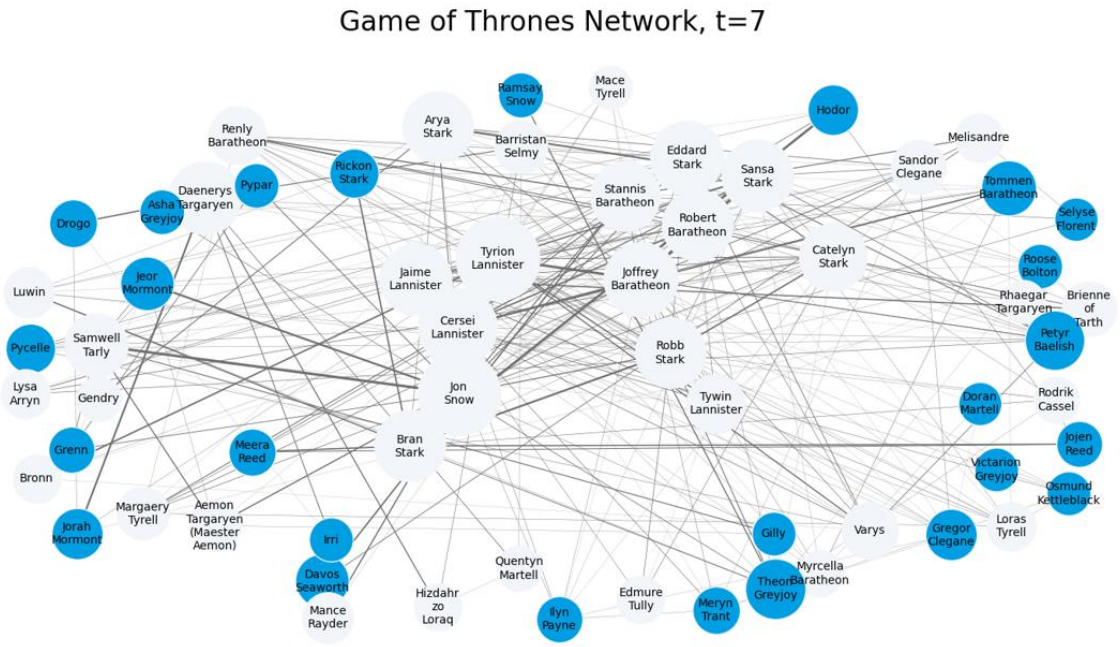
Ở thời điểm t=5



Ở thời điểm t=6



Ở thời điểm t=7

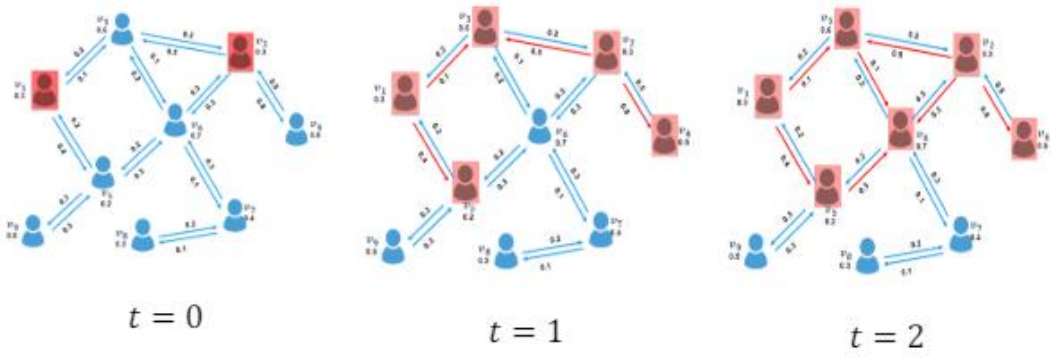


- Như vậy ở thời điểm t=7 thì đã không còn lây nhiễm thêm được nữa, đó là toàn bộ quá trình lây nhiễm của mô hình independent cascade

III. Linear Thresholds Model (LT)

- Mô hình này giả định rằng mỗi cá nhân có một ngưỡng tác động, tức là một số lượng nhất định của các cá nhân trong hàng xóm cần nhận được thông điệp hoặc ý kiến mới để chấp nhận và lan truyền tiếp. Khi số lượng những cá nhân ảnh hưởng vượt qua ngưỡng, cá nhân đó sẽ chấp nhận thông điệp và lan truyền tiếp. Mô hình ngưỡng giúp mô phỏng hiện tượng lan truyền thông điệp bùng phát hoặc lan truyền thông điệp không thành công.
- Mô hình này sử dụng ngưỡng cá nhân để xác định liệu một cá nhân có thay đổi trạng thái của mình dựa trên tỷ lệ hoặc số lượng cá nhân trong mạng lưới đã chấp nhận hành vi đó.
- Đối với mô hình LT, cách thể hiện MXH là bằng cách biểu diễn trên một đồ thị $G(V, E, w)$, mỗi cạnh $(u, v) \in E$ có trọng số $w(u, v) \in [0,1]$ thể hiện ảnh hưởng của nút u đến nút v, Nếu $(u, v) \notin E$ thì $w(u, v) = 0$, được phân bố sao cho tổng trọng số các nút u đến nút v thỏa mãn điều kiện: $\sum u \in N\ w(u, v)\mathrm{N}(v) \leq 1$.
- Mỗi nút $v \in V$ có 2 trạng thái: 1) kích hoạt (active); 2) không kích hoạt (inactive). Mỗi nút $v \in V$ có ngưỡng kích hoạt $\gamma v \in [0,1]$, nếu γv lớn thì cần nhiều nút hàng xóm kích hoạt v, nếu γv bé thì nút v dễ bị kích hoạt bởi các nút hàng xóm. Gọi $\mathcal{D}t(G, S)$ là tập các nút bị ,kích hoạt bởi S tại thời điểm t trên đồ thị G, mô tả quá trình lan truyền ứng với từng bước thời gian rời rạc như sau:

- Với thời điểm t = 0, tất cả các nút trong tập $S = \mathcal{D}^0$ (G, S) đều có trạng thái kích hoạt
- Với thời điểm $t \geq 1$, các nút v đang có trạng thái không kích hoạt, sẽ đổi trạng thái sang kích hoạt nếu các nút hàng xóm có tổng ảnh hưởng lớn hơn ngưỡng γv. Các nút có trạng thái kích hoạt sẽ tiếp tục trạng thái trong những thời điểm tiếp theo. Khi không có nút nào được kích hoạt thêm thì quá trình lan truyền này kết thúc
- Trong khi chịu ảnh hưởng của các nhân khác thì mô hình LT thể hiện hành vi "ngưỡng" của con người. Bởi vì ngưỡng kích hoạt của các cá nhân luôn thay đổi nên thường khó xác định được. Vì vậy, ngưỡng kích hoạt được chọn ngẫu nhiên trong khoảng [0,1] trong mô hình này thể hiện sự thiếu tri thức về ngưỡng ảnh hưởng thật của người dùng.
- **Ví dụ:**



Trạng thái kích hoạt và không kích hoạt tương ứng lần lượt với các nút màu da cam và màu xanh. Các cạnh màu đỏ liền kề nhau nối với nút v mô tả đang lan truyền thông tin và cố kích hoạt nút v và thành công.

Tại thời điểm t = 0, trong khoảng ngưỡng $\gamma v \in [0, 1]$ toàn bộ các nút được khởi tạo ngẫu nhiên, hai nút $v1$ và $v2$ là các nút hạt giống.

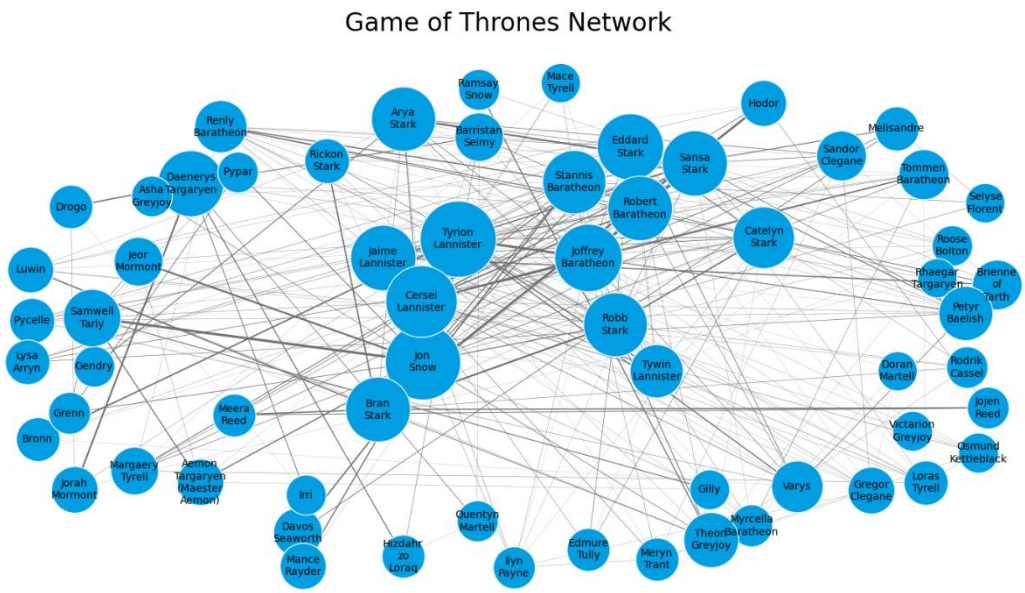
Ở thời điểm t = 1, $v1$ và $v2$ kích hoạt thành công $v3$, $v1$ cũng kích hoạt thành công $v5$ và $v2$ kích hoạt thành công $v4$; tuy nhiên $v6$ lại không bị kích hoạt vì tổng trọng số các cạnh đi đến $v6$ là 0.3, trong khi ngưỡng kích hoạt của $v6$ là 0.7.

Tại thời điểm t = 2, các nút hàng xóm đi đến $v6$ là $v1$,$v2$, $v5$ đã được kích hoạt cho nên tổng trọng số các cạnh đi đến là 0.7 đủ để kích hoạt $v6$. Tại bước t = 3, quá trình lan truyền thông tin kết thúc do không có nút nào được kích hoạt thêm

- **Triển khai thuật toán bằng ngôn ngữ python:**

```python
def linear_threshold(G, t, infection_times, thresholds):
    # Doing a t->t+1 step of linear_threshold simulation
    new_infections = []
    for n in G.nodes():
        if n not in infection_times:
            total_influence = sum(G[u][n]['weight'] for u in G.neighbors(n) if u in infection_times and infection_times[u] == t)
            if total_influence >= thresholds[n]:
                new_infections.append(n)

    for n in new_infections:
        infection_times[n] = t + 1

    return infection_times
```

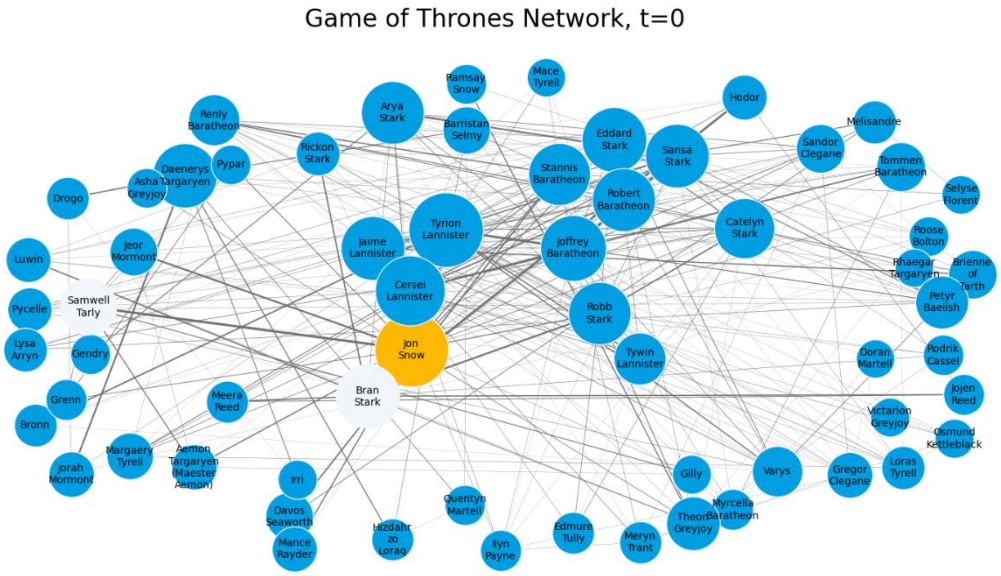
- **Ứng dụng với bộ dữ liệu:**
  Bộ dữ liệu ban đầu – Các node chưa bị nhiễm (màu xanh), đã bị nhiễm (màu trắng), bị nhiễm ở thời gian T(màu vàng)



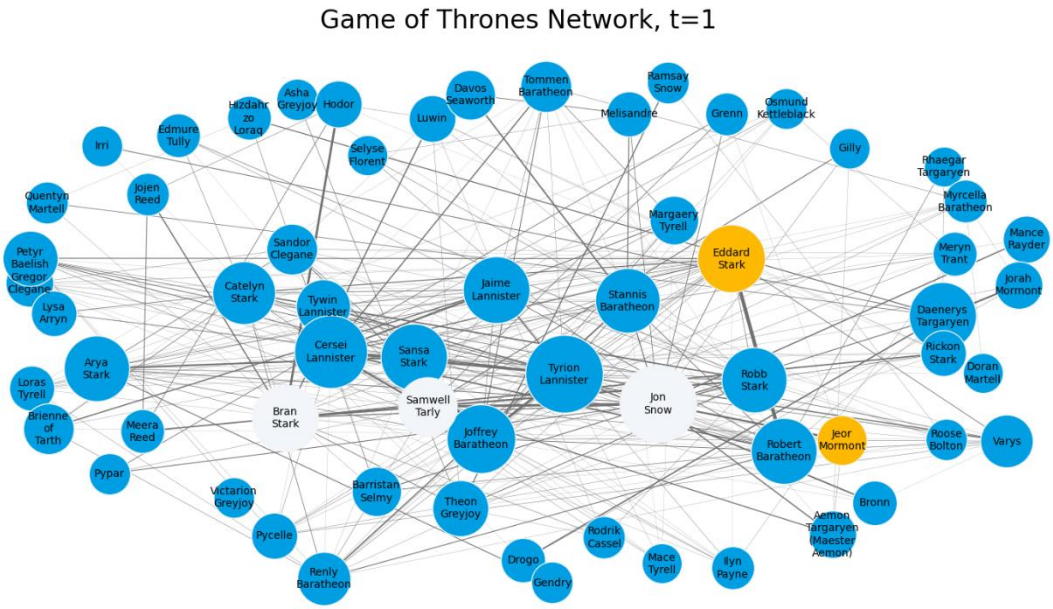
- Giả sử các seed ban đầu bị nhiễm bệnh

```python
infection_times = {'Bran-Stark':-1,'Samwell-Tarly':-1,'Jon-Snow':0}
```

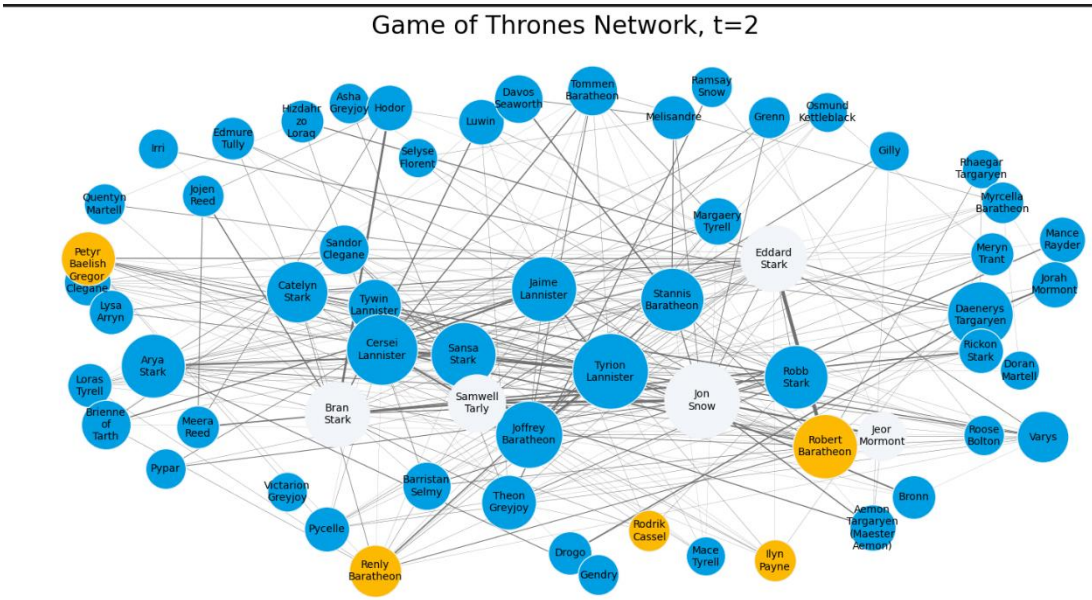
- Ở thời điểm t=0



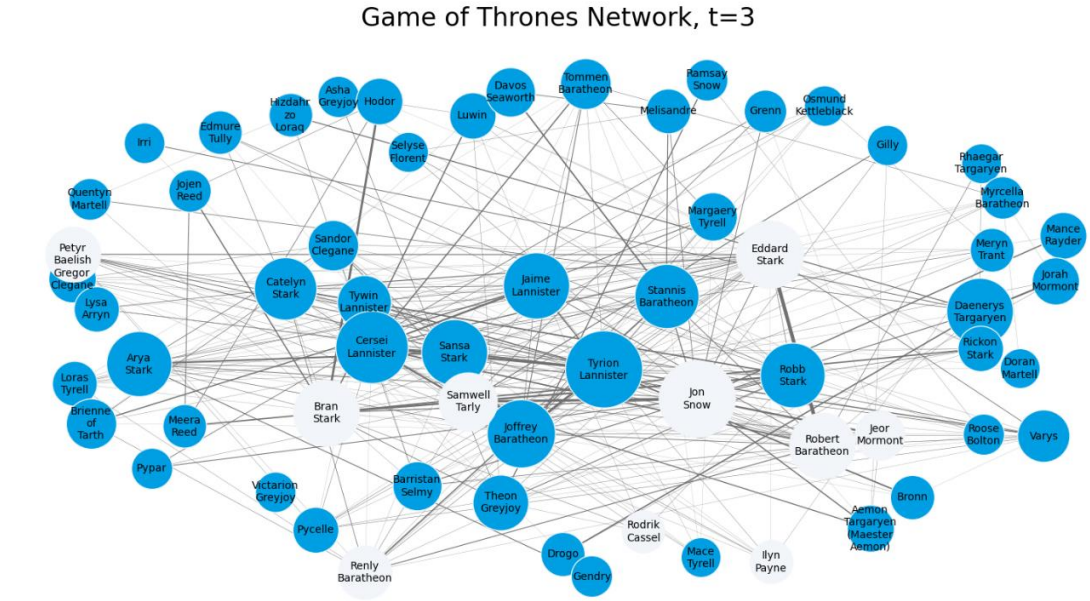
- Ở thời điểm t=1



- Ở thời điểm t=2



- Ở thời điểm t=3



➢ Như vậy ở thời điểm t=3 thì đã không còn lây nhiễm thêm được nữa, đó là toàn bộ quá trình lây nhiễm của mô hình linear threshold

CHƯƠNG 4: TỐI ƯU HÓA LAN TRUYỀN THÔNG TIN

I. Định nghĩa

- Influence Maximization là một bài toán quan trọng trong lĩnh vực mạng xã hội, nơi mục tiêu là xác định một tập hợp nhỏ các nút (người dùng) để tối đa hóa sự lan truyền của thông tin qua mạng. Cụ thể, bài toán này thường liên quan đến việc lựa chọn một số lượng giới hạn các cá nhân có ảnh hưởng nhất sao cho khi họ được cung cấp thông tin ban đầu, thông tin này sẽ lan rộng đến số lượng lớn nhất có thể trong toàn bộ mạng xã hội. Quá trình lan truyền thường được mô hình hóa thông qua các mô hình toán học như mô hình ngưỡng tuyến tính (Linear Threshold) và mô hình độc lập (Independent Cascade).

II. Ứng dụng

- Ngày nay, với sự phát triển nhanh chóng của web mạng xã hội, các ứng dụng online như Facebook, Youtube, Twitter,… mang lại nguồn thông tin phong phú, đồng thời con người có thể dễ dàng kết nối với nhiều mối quan hệ khác. Những mạng này có thể giúp cho việc tiếp thị trở nên dễ dàng, cho phép thông tin và ý tưởng có thể ảnh hưởng đến một số lượng lớn trong một thời gian ngắn. Với sự hỗ trợ của các lý thuyết đồ thị, con người có thể trích xuất được rất nhiều thông tin ngữ nghĩa quan trọng và hữu ích từ các đồ thị mạng
- Hãy xem xét ví dụ sau:

Một công ty nhỏ muốn phát triển một ứng dụng trực tuyến rất triển vọng trong một mạng xã hội trực tuyến và muốn tiếp thị thông qua chúng. Nhưng công ty đó lại có một ngân sách hạn chế, vì vậy chỉ có thể lựa chọn số lượng nhỏ người sử dụng ban đầu trong mạng để sử dụng nó (bằng cách cho họ quà tặng hoặc các khoản thanh toán). Công ty mong muốn rằng những người sử dụng ban đầu sẽ thích ứng dụng đó và bắt đầu ảnh hưởng đến bạn bè của họ trong mạng xã hội để cùng sử dụng nó, và bạn bè của họ cũng sẽ như vậy. Như vậy, nếu như trong xã hội thực ta có thể thực hiện điều đó bằng cách lan truyền miệng, còn trong mạng xã hội thì cần thông qua các ứng dụng.

Vấn đề ở đây là chọn ai làm người sử dụng ban đầu để kết quả thu được có sự ảnh hưởng đến số lượng người sử dụng lớn nhất trong mạng, tức là ta trở về vấn đề tìm kiếm các cá nhân có ảnh hưởng nhất trong mạng xã hội

Vấn đề này được gọi là tối ưu ảnh hưởng, sẽ là quan tâm của nhiều công ty cũng như các cá nhân muốn quảng bá sản phẩm, dịch vụ của họ, và ý tưởng sáng tạo của họ thông qua các cách thức tiếp thị lan truyền. Mạng xã hội trực tuyến cung cấp các giải pháp để giải quyết vấn đề này, bởi vì chúng đang kết nối một số lượng lớn người với nhau và chúng thu thập một lượng lớn thông tin về cấu trúc cũng như động lực truyền thông trên mạng xã hội. Tuy nhiên, cũng có những thách thức đặt ra khi giải quyết vấn đề này đó là : các mạng xã hội có quy mô lớn, có cấu trúc kết nối phức tạp và luôn biến đổi theo

thời gian, có nghĩa là giải pháp cho vấn đề này cần phải được rất hiệu quả và có khả năng mở rộng

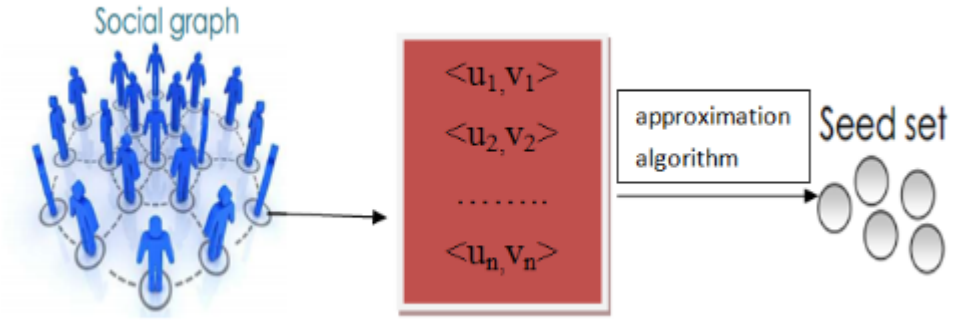
➔ Ứng dụng của Influence Maximization rất rộng rãi, bao gồm tiếp thị truyền miệng, quảng cáo, và cả trong các chiến dịch truyền thông xã hội.

III. Các thuật toán

1. Phát biểu bài toán tối ưu hóa ảnh hưởng

Bài toán tối ưu hóa ảnh hưởng của đối tượng trong mạng xã hội phát biểu như sau :

- Đầu vào: Một mạng xã hội được mô phỏng như một đồ thị vô hướng G (V,E), với đỉnh V mô hình hóa các cá nhân trong mạng và cạnh E mô hình hóa các mối quan hệ giữa các cá nhân, các cặp (đỉnh, hàng xóm của đỉnh) = (ui,vi).
- Đầu ra: Tất cả các cặp quan hệ (đỉnh, hàng xóm của đỉnh) có điểm ảnh hưởng lớn nhất.



Hình mô tả bài toán

Ví dụ: Trong một mạng có 4 người sử dụng A, B, C, D. A là bạn bè của B, B là bạn của C, C là bạn của D, A là bạn của C, A là bạn của D, B là bạn của D. Từ đó ta có thể xét ra các cặp là (đỉnh, hàng xóm của đỉnh là) là (A,B), (B,C), (C,D), (A,C), (A,D), (B,D). Những cặp đó trở thành đầu vào của thuật toán mà chúng ta lựa chọn, sau khi cài đặt thuật toán, ta thu được đầu ra là tập "nhân" có ảnh hưởng lớn nhất trong mạng.

2. Thuật toán Greedy

Thuật toán Greedy trong bài toán tối đa hóa ảnh hưởng (Influence Maximization) là một phương pháp hiệu quả để chọn ra một tập hợp các đỉnh (nodes) có khả năng lan truyền thông tin hoặc ảnh hưởng đến nhiều đỉnh khác nhất trong một mạng xã hội. Mục tiêu chính của thuật toán này là xác định k đỉnh trong đồ thị sao cho phạm vi ảnh hưởng của chúng được tối ưu hóa, dựa trên mô hình lan truyền ảnh hưởng, chẳng hạn như mô hình Independent Cascade (IC) hoặc Linear Threshold (LT).

Nguyên tắc hoạt động

Bước 1: Đánh giá Marginal Gain

Đối với mỗi đỉnh chưa nằm trong tập S, thuật toán sẽ tính toán "marginal gain" (lợi ích biên) - số lượng đỉnh mới bị ảnh hưởng nếu thêm đỉnh đó vào tập S. Việc tính toán này thường được thực hiện bằng cách mô phỏng quá trình lan truyền ảnh hưởng nhiều lần (Monte Carlo simulations) để ước tính giá trị trung bình của số đỉnh bị ảnh hưởng.

Bước 2: Chọn Đỉnh Tốt Nhất

Sau khi đánh giá tất cả các đỉnh, thuật toán sẽ chọn đỉnh có marginal gain lớn nhất và thêm nó vào tập S.

Bước 3: Cập Nhật Tập S

Tập S sẽ được cập nhật bằng cách thêm đỉnh vừa được chọn vào.

Bước 4: Lặp Lại

Quy trình trên được lặp lại cho đến khi chọn đủ k đỉnh.

Độ phức tạp thuật toán

Độ phức tạp của một lần lặp:

Trong mỗi lần lặp, thuật toán thực hiện các bước sau:

Đánh giá Marginal Gain: Đối với mỗi đỉnh chưa nằm trong tập S, tính toán marginal gain bằng cách thực hiện R lần mô phỏng. Giả sử việc thực hiện một mô phỏng có độ phức tạp là $O(m)$, do cần duyệt qua các cạnh để lan truyền ảnh hưởng.

Đối với mỗi đỉnh, việc tính toán marginal gain có độ phức tạp là O(R·m)

➔ Tổng hợp lại, độ phức tạp của một lần lặp là O(n·R·m)

Độ phức tạp của toàn bộ thuật toán

Thuật toán lặp lại quá trình trên k lần để chọn đủ k đỉnh. Do đó, độ phức tạp tổng thể của thuật toán Greedy là: O(k·n·R·m)

Ưu điểm:

+ Hiệu quả: Thuật toán Greedy cung cấp một phương pháp tiếp cận đơn giản và trực quan để giải quyết bài toán tối đa hóa ảnh hưởng.

+ Đảm bảo gần tối ưu: Mặc dù không đảm bảo kết quả tối ưu tuyệt đối, thuật toán Greedy thường cho kết quả rất gần với tối ưu và có thể được chứng minh là đạt ít nhất 63% hiệu quả của giải pháp tối ưu (theo lý thuyết của các hàm con).

Nhược điểm:

+ Chi phí tính toán cao: Việc tính toán marginal gain cho mỗi đỉnh có thể tốn kém, đặc biệt là trong các đồ thị lớn, do phải thực hiện nhiều lần mô phỏng Monte Carlo.

+ Không linh hoạt với các mô hình lan truyền phức tạp: Thuật toán Greedy có thể cần điều chỉnh để phù hợp với các mô hình lan truyền khác nhau ngoài IC và LT.

Đánh giá: Thuật toán Greedy cho bài toán tối đa hóa ảnh hưởng có độ phức tạp là O(k·n·R·m), nghĩa là nó có thể trở nên rất tốn kém về mặt tính toán đối với các đồ thị lớn. Tuy nhiên, với các cải tiến như CELF và sử dụng xử lý song song, hiệu quả của thuật toán có thể được cải thiện đáng kể. Mặc dù vậy, cần phải cân nhắc kỹ lưỡng giữa độ chính xác và chi phí tính toán khi ứng dụng thuật toán này trong thực tế.

Thực hiện bằng ngôn ngữ python

```
[54] def greedy(g,k,p=0.5,mc=500):
         """
         Input:  graph object, number of seed nodes
         Output: optimal seed set, resulting spread, time for each iteration
         """

         S, spread, timelapse, start_time = [], [], [], time.time()

         # Find k nodes with largest marginal gain
         for _ in range(k):

             # Loop over nodes that are not yet in seed set to find biggest marginal gain
             best_spread = 0
             for j in set(range(len(g.vs))) - set(S):

                 # Get the spread
                 s = IC(g,S + [j],p,mc)

                 # Update the winning node and spread so far
                 if s > best_spread:
                     best_spread, node = s, j

             # Add the selected node to the seed set
             S.append(node)

             # Add estimated spread and elapsed time
             spread.append(best_spread)
             timelapse.append(time.time() - start_time)

         return(S,spread,timelapse)
```

3. Thuật toán CELF.

Thuật toán Greedy là thuật toán đơn giản nhất, dễ thực hiện cài đặt. Hiệu suất tối ưu của thuật toán là 1-1/e (63%) [10] tương đối tốt nhưng nó có một số hạn chế là độ phức tạp lớn, mất nhiều thời gian chạy. Vì vậy, để giải quyết vấn đề này cách tối ưu thuật toán này bằng cách kết hợp thuật toán này với thuật toán CELF (Cost Effective Lazy Forward).

Thuật toán CELF (Cost-Effective Lazy Forward) được coi là một cải tiến quan trọng so với thuật toán Greedy truyền thống. CELF giúp giảm đáng kể số lần tính toán cần thiết để xác định các đỉnh có ảnh hưởng lớn nhất, từ đó tăng hiệu suất tính toán mà vẫn duy trì độ chính xác cao.

Thực hiện bằng ngôn ngữ python

```python
def celf(g,k,p=0.5,mc=500):
    """
    Input:  graph object, number of seed nodes
    Output: optimal seed set, resulting spread, time for each iteration
    """


        # --------------------
    # Find the first node with greedy algorithm
    # --------------------
    # Calculate the first iteration sorted list
    start_time = time.time()
    marg_gain = [IC(g,[node],p,mc) for node in range(g.vcount())]

    # Create the sorted list of nodes and their marginal gain
    Q = sorted(zip(range(g.vcount()),marg_gain), key=lambda x: x[1],reverse=True)

    # Select the first node and remove from candidate list
    S, spread, SPREAD = [Q[0][0]], Q[0][1], [Q[0][1]]
    Q, LOOKUPS, timelapse = Q[1:], [g.vcount()], [time.time()-start_time]

    # --------------------
    # Find the next k-1 nodes using the list-sorting procedure
    # --------------------

    for _ in range(k-1):

        check, node_lookup = False, 0

        while not check:

            # Count the number of times the spread is computed
            node_lookup += 1

            # Recalculate spread of top node
            current = Q[0][0]

            # Evaluate the spread function and store the marginal gain in the list
            Q[0] = (current,IC(g,S+[current],p,mc) - spread)

            # Re-sort the list
            Q = sorted(Q, key = lambda x: x[1], reverse = True)

            # Check if previous top node stayed on top after the sort
            check = (Q[0][0] == current)

        # Select the next node
        spread += Q[0][1]
        S.append(Q[0][0])
        SPREAD.append(spread)
        LOOKUPS.append(node_lookup)
        timelapse.append(time.time() - start_time)

        # Remove the selected node from the list
```

Nguyên tắc hoạt động của thuật toán CELF

CELF dựa trên ý tưởng tối ưu hóa quá trình tìm kiếm các đỉnh có marginal gain lớn nhất thông qua việc lưu trữ và tái sử dụng thông tin tính toán. Cụ thể, CELF thực hiện theo các bước sau:

1. Khởi tạo: Ban đầu, thuật toán tính toán marginal gain cho tất cả các đỉnh và lưu trữ các giá trị này cùng với thứ tự ưu tiên trong một hàng đợi ưu tiên (priority queue).

2. Chọn Đỉnh Tốt Nhất: Trong mỗi lần lặp, thuật toán chọn đỉnh có marginal gain lớn nhất từ hàng đợi ưu tiên.

3. Cập Nhật Marginal Gain: Khi một đỉnh được chọn, marginal gain của các đỉnh khác sẽ được cập nhật. Tuy nhiên, CELF chỉ cập nhật giá trị marginal gain cho đỉnh ở đầu hàng đợi ưu tiên và đẩy nó xuống hàng đợi nếu cần thiết. Điều này giúp giảm số lần tính toán cần thiết.

4. Lặp Lại: Quy trình trên được lặp lại cho đến khi chọn đủ k đỉnh.

Độ phức tạp thuật toán

Khởi tạo

Khởi tạo marginal gain: Ban đầu, thuật toán tính toán marginal gain cho tất cả các đỉnh. Với mỗi đỉnh, cần thực hiện R lần mô phỏng để ước tính ảnh hưởng.

➔ Độ phức tạp của bước này là O(n·R·m), vì mỗi mô phỏng có độ phức tạp O(m).

Chọn và cập nhật đỉnh

Trong mỗi lần lặp, thuật toán CELF thực hiện các bước sau:

1. Chọn đỉnh có marginal gain lớn nhất: Đỉnh này được lấy từ hàng đợi ưu tiên (priority queue). Thao tác này có độ phức tạp O(log$_{f0}$n)
2. Cập nhật marginal gain: Nếu đỉnh ở đầu hàng đợi không phải là đỉnh có marginal gain thực sự lớn nhất (do thay đổi khi các đỉnh khác được chọn), thuật toán sẽ cập nhật marginal gain của đỉnh đó và đưa nó trở lại hàng đợi nếu cần. Trong trường hợp xấu nhất, cần cập nhật tất cả n đỉnh.
3. Số lần cập nhật: Trong thực tế, số lần cập nhật thực sự ít hơn nhiều so với trường hợp xấu nhất, do thuật toán "lười biếng" (lazy) trong việc cập nhật. Thông thường, số lần cập nhật marginal gain cho mỗi đỉnh là rất nhỏ.

Tổng độ phức tạp

Kết hợp lại, độ phức tạp tổng thể của thuật toán CELF có thể được ước lượng như sau:

Khởi tạo: O(n·R·m)

Chọn và cập nhật đỉnh: Trong trường hợp xấu nhất, cần cập nhật tất cả n đỉnh k lần, do đó độ phức tạp là O(k·n·R·m).

➔ Vì vậy, độ phức tạp tổng thể của thuật toán CELF trong trường hợp xấu nhất là: O(n·R·m+k·n·R·m)=O((k+1)·n·R·m)

Ứng dụng vào bộ dữ liệu

Thay vì chọn seed lây nhiễm ban đầu là

```
infection_times = {'Bran-Stark':-1,'Samwell-Tarly':-1,'Jon-Snow':0}
```

Thì 2 thuật toán sẽ tối ưu cho ta chọn seed lây nhiễm như sau

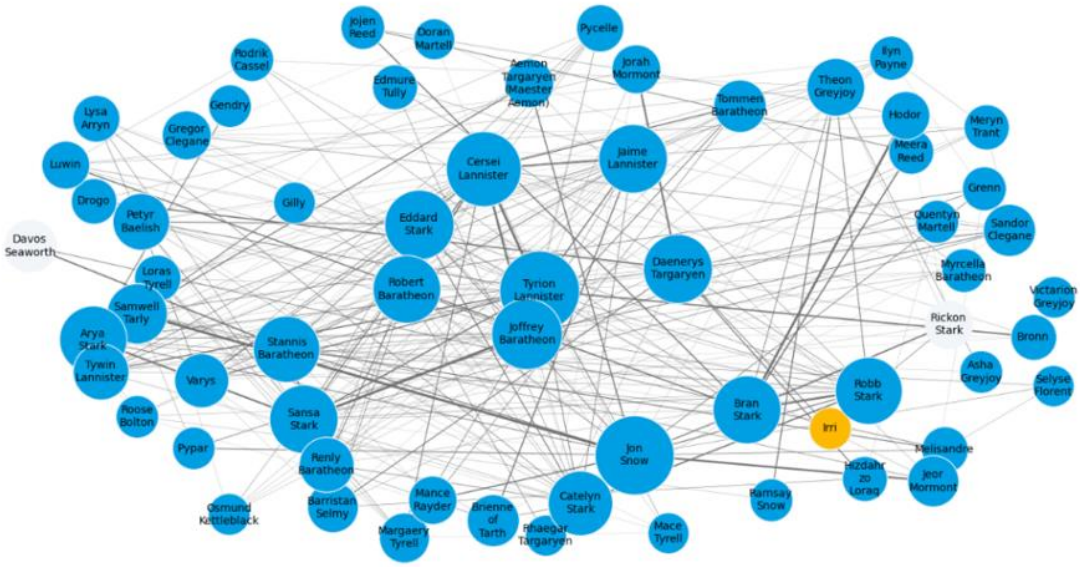
```
[55] celf_output   = celf(g,3,p = 0.5,mc = 500)
     greedy_output = greedy(g,3,p = 0.5,mc = 500)

     # Print results
     print("celf output:   " + str(celf_output[0]))
     print("greedy output: " + str(greedy_output[0]))

     celf output:   [1, 0, 13]
     greedy output: [1, 0, 13]
```

Đó là node thứ 1, 0 và 13. Sẽ gây ảnh hưởng lớn nhất đến toàn mạng xã hội

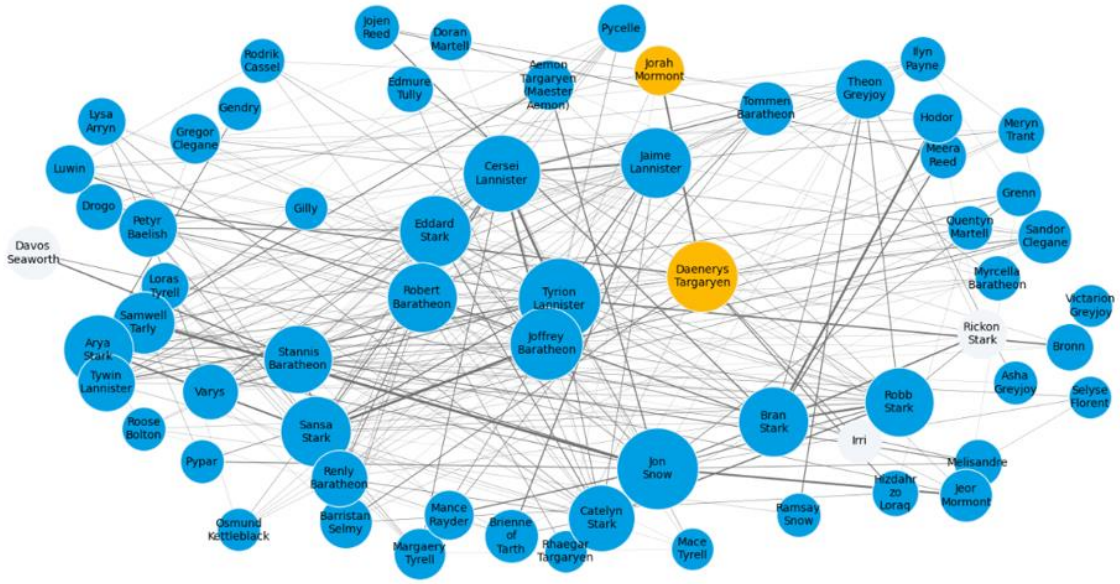
Sau đó sẽ thử chạy thuật toán với mô hình Independent Cascade ta sẽ thu được

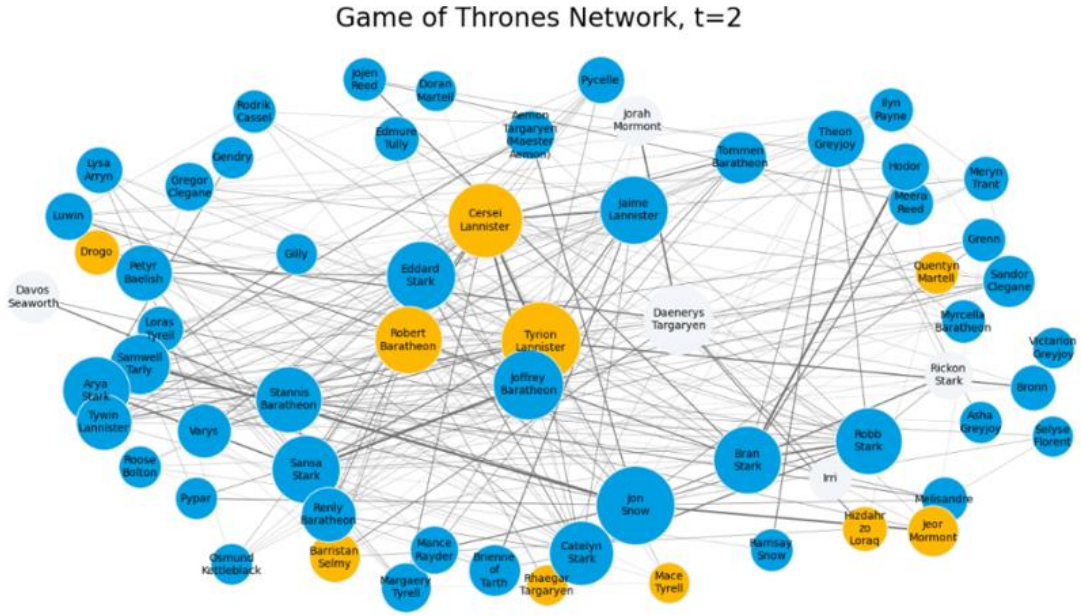
```
[60] infection_times = {'Rickon-Stark':-1,'Davos-Seaworth':-1,'Irri':0}
```

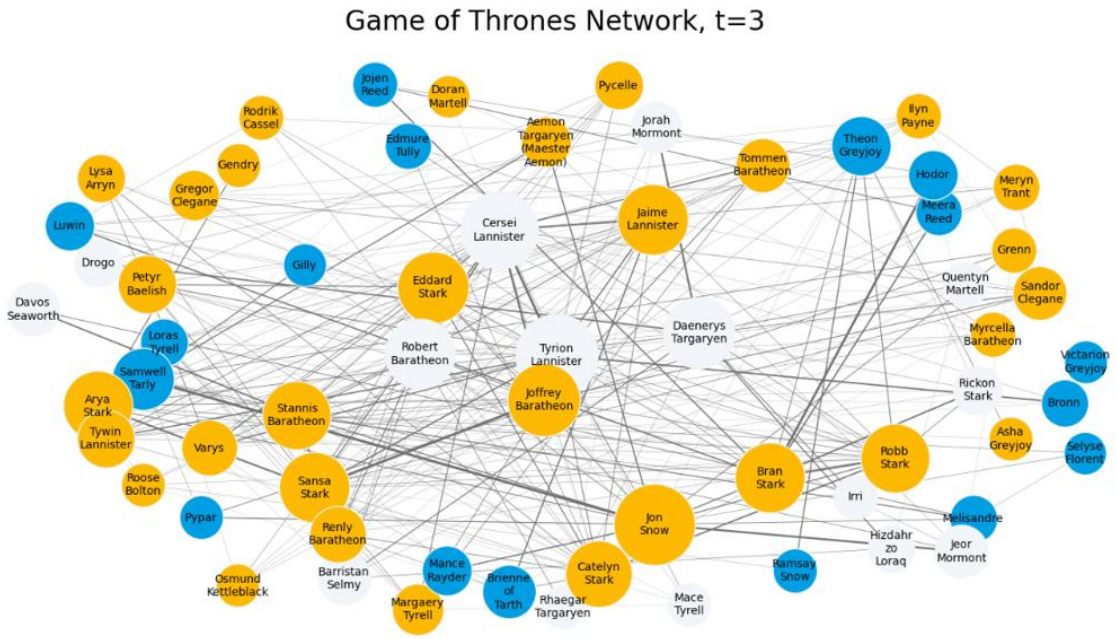
Game of Thrones Network, t=0

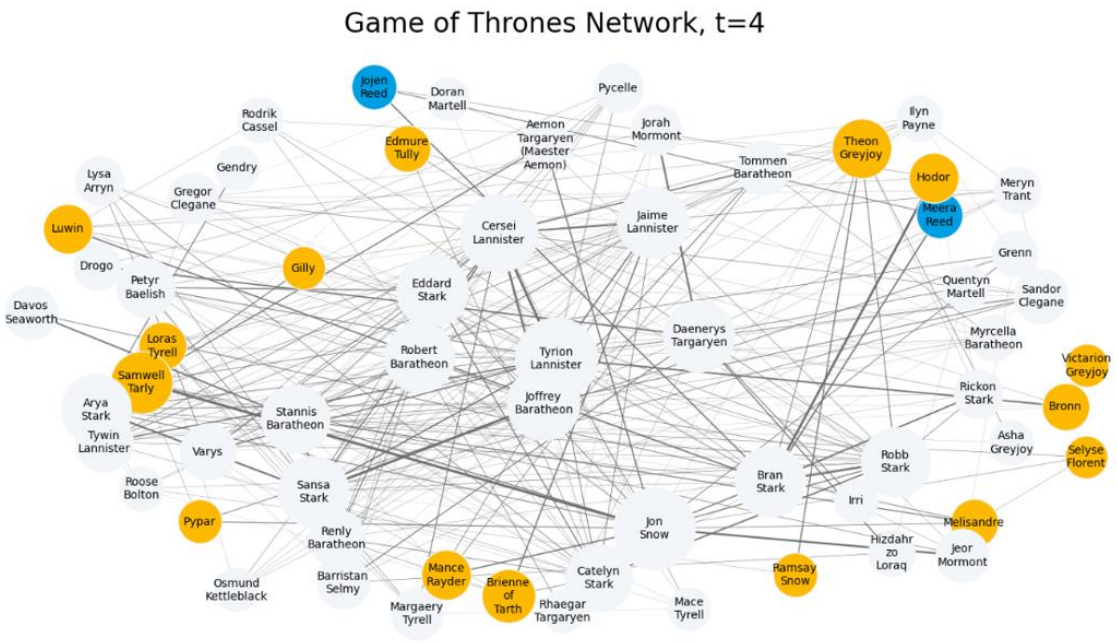


Game of Thrones Network, t=1




Game of Thrones Network, t=2
Jojen Reed
Doran Martell
Pycelle
Rodrik Cassel
Edmure Tully
Aemon Targaryen (Maester Aemon)
Jorah Mormont
Ilyn Payne
Theon Greyjoy
Tommen Baratheon
Hodor
Meryn Trant
Meera Reed
Lysa Arryn
Gendry
Gregor Clegane
Luwin
Cersei Lannister
Jaime Lannister
Drogo
Petyr Baelish
Gilly
Eddard Stark
Grenn
Quentyn Martell
Sandor Clegane
Davos Seaworth
Daenerys Targaryen
Myrcella Baratheon
Loras Tyrell
Robert Baratheon
Tyrion Lannister
Victarion Greyjoy
Samwell Tarly
Rickon Stark
Arya Stark
Joffrey Baratheon
Bronn
Stannis Baratheon
Asha Greyjoy
Tywin Lannister
Varys
Robb Stark
Selyse Florent
Bran Stark
Roose Bolton
Sansa Stark
Irri
Melisandre
Pypar
Jon Snow
Hizdahr zo Loraq
Jeor Mormont
Renly Baratheon
Ramsay Snow
Mance Rayder
Catelyn Stark
Barristan Selmy
Brienne of Tarth
Osmund Kettleblack
Rhaegar Targaryen
Margaery Tyrell
Mace Tyrell


Game of Thrones Network, t=3
Jojen Reed
Doran Martell
Pycelle
Rodrik Cassel
Edmure Tully
Aemon Targaryen (Maester Aemon)
Jorah Mormont
Ilyn Payne
Theon Greyjoy
Tommen Baratheon
Hodor
Meryn Trant
Meera Reed
Lysa Arryn
Gendry
Gregor Clegane
Luwin
Cersei Lannister
Jaime Lannister
Drogo
Petyr Baelish
Gilly
Eddard Stark
Grenn
Quentyn Martell
Sandor Clegane
Davos Seaworth
Daenerys Targaryen
Myrcella Baratheon
Loras Tyrell
Robert Baratheon
Tyrion Lannister
Victarion Greyjoy
Samwell Tarly
Rickon Stark
Arya Stark
Joffrey Baratheon
Bronn
Stannis Baratheon
Asha Greyjoy
Tywin Lannister
Varys
Robb Stark
Selyse Florent
Bran Stark
Roose Bolton
Sansa Stark
Irri
Melisandre
Pypar
Jon Snow
Hizdahr zo Loraq
Jeor Mormont
Renly Baratheon
Ramsay Snow
Mance Rayder
Catelyn Stark
Barristan Selmy
Brienne of Tarth
Osmund Kettleblack
Rhaegar Targaryen
Margaery Tyrell
Mace Tyrell

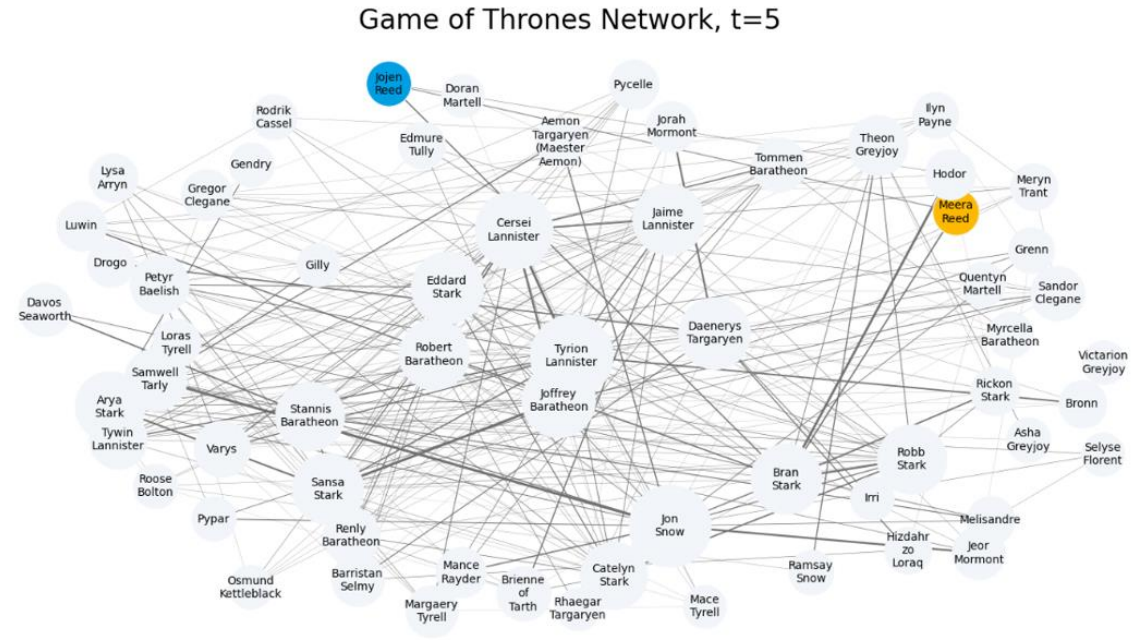


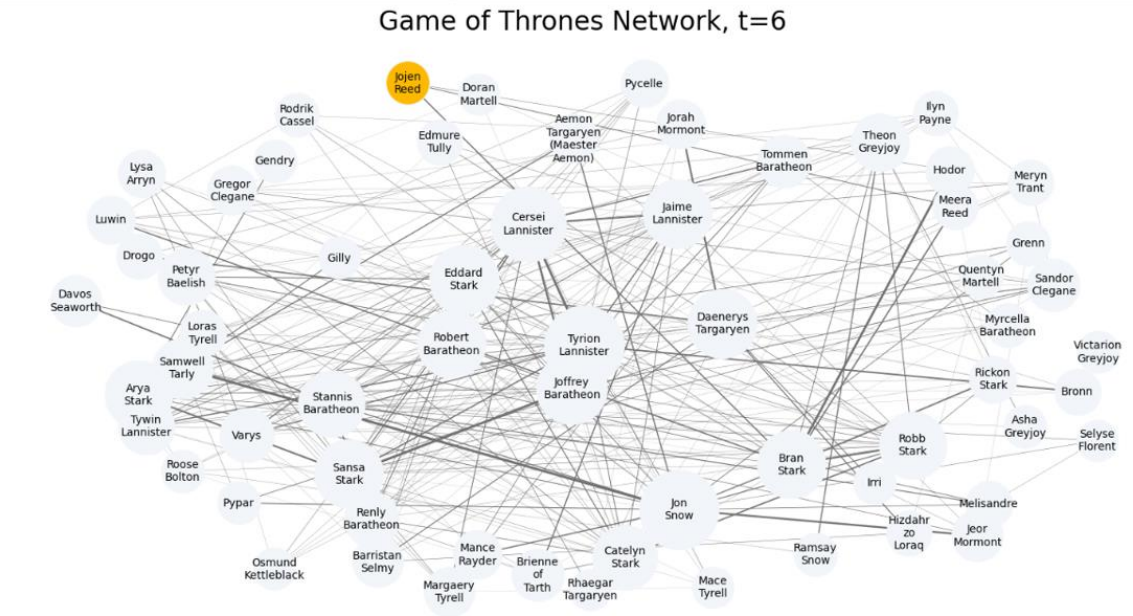
Đánh giá

Thuật toán CELF cải thiện hiệu quả so với thuật toán Greedy truyền thống nhờ việc giảm số lần tính toán marginal gain thông qua chiến lược "lười biếng" và hàng đợi ưu tiên. Mặc dù độ phức tạp trong trường hợp xấu nhất là tương đương với Greedy, nhưng trong thực tế, CELF thường nhanh hơn nhiều do số lần cập nhật marginal gain thực sự nhỏ hơn. Điều này làm cho CELF trở thành một lựa chọn hiệu quả hơn cho bài toán tối đa hóa ảnh hưởng trong các mạng xã hội lớn.

Vòng đầu tiên của CELF thì nó cũng chạy chậm như thuật toán gốc. Tuy nhiên bắt đầu từ vòng thứ 2, nếu dùng CELF cúng ta chỉ cần tìm thêm một số lượng nhỏ các đỉnh và các hàm RanCas(S) thường dừng ngay để tiếp tục với phần khác của đồ thị. Ngược lại, trong mỗi vòng của thuật toán NewGreedyIc chúng ta cần phải đi qua toàn bộ đồ thị R một lần để tạo ra các R ngẫu nhiên từ đồ thị G

Để giải quyết vấn đề này, đề xuất thuật toán MixGreedyIC, đó là vòng đầu tiên sẽ sử dụng thuật toán NewGreedyIC và các vòng tiếp theo sẽ sử dụng giải thuật tối ưu CELE để lựa chọn các điểm làm hạt nhân còn lại

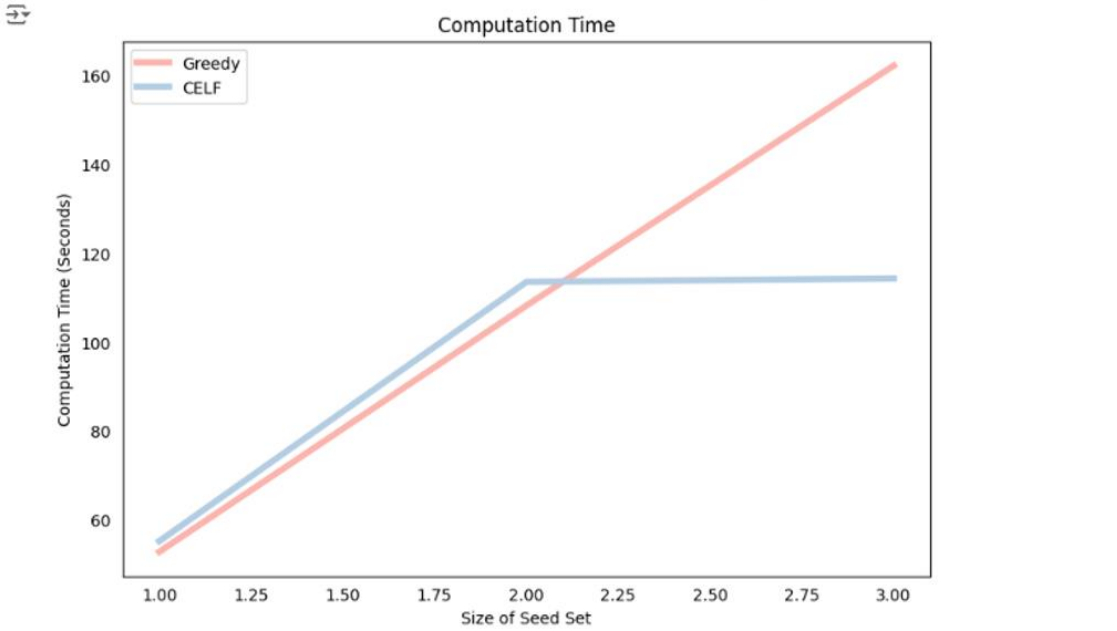




- Chỉ sau khoảng thời gian t=6 thì toàn bộ mạng xã hội đã bị ảnh hưởng

So sánh 2 thuật toán

```
plt.rcParams['figure.figsize'] = (9,6)
plt.rcParams['lines.linewidth'] = 4
plt.rcParams['xtick.bottom'] = False
plt.rcParams['ytick.left'] = False

# Plot Computation Time
plt.plot(range(1,len(greedy_output[2])+1),greedy_output[2],label="Greedy",color="#FBB4AE")
plt.plot(range(1,len(celf_output[2])+1),celf_output[2],label="CELF",color="#B3CDE3")
plt.ylabel('Computation Time (Seconds)'); plt.xlabel('Size of Seed Set')
plt.title('Computation Time'); plt.legend(loc=2);
```



Đánh giá:

Greedy Algorithm: Thời gian tính toán của Greedy Algorithm tăng tuyến tính với số lượng đỉnh khởi nguồn (seed nodes) cần tìm. Mỗi lần thêm một đỉnh, nó phải tính toán lại khả năng lan truyền của tất cả các đỉnh chưa chọn, dẫn đến thời gian tính toán dài khi số lượng đỉnh lớn.

CELF: CELF giảm thiểu số lần tính toán bằng cách lưu trữ và sử dụng lại kết quả từ các bước trước đó. Kỹ thuật "lazy evaluation" giúp CELF chỉ thực hiện tính toán khi cần thiết, do đó, thời gian tính toán giảm đáng kể so với Greedy Algorithm. Các thử nghiệm đã chứng minh rằng CELF có thể nhanh hơn Greedy Algorithm đến một bậc độ lớn, đặc biệt khi số lượng đỉnh và cạnh trong đồ thị lớn.

➔ Kết luận, nghiên cứu này đã chỉ ra rằng phương pháp CELF có ưu thế vượt trội hơn so với Greedy Algorithm về thời gian tính toán, đặc biệt khi số lượng đỉnh khởi nguồn ban đầu lớn. CELF cung cấp một giải pháp hiệu quả hơn cho việc quản lý và khai thác mạng xã hội, giúp tiết kiệm thời gian và tài nguyên trong các chiến dịch marketing, quảng cáo và truyền thông cộng đồng

CHƯƠNG 5: TỔNG KẾT

Trong đồ án này, chúng ta đã nghiên cứu lý thuyết đồ thị và ứng dụng của nó trong việc lan truyền thông tin trên mạng xã hội. Đồ thị, với các đỉnh đại diện cho người dùng và các cạnh đại diện cho mối quan hệ hoặc tương tác giữa họ, là công cụ quan trọng để mô hình hóa và phân tích mạng xã hội. Lan truyền thông tin là quá trình mà thông tin, ý tưởng hoặc nội dung lan rộng từ một người dùng đến những người dùng khác thông qua các kết nối mạng. Các mô hình như Independent Cascade Model (ICM) và Linear Threshold Model (LTM) đã được sử dụng để mô tả chi tiết quá trình này.

Bên cạnh đó, chúng ta đã xem xét các phương pháp tối ưu hóa lan truyền thông tin nhằm xác định các đỉnh khởi nguồn tối ưu. Các phương pháp này bao gồm Greedy Algorithm và Cost-Effective Lazy Forward (CELF), mỗi phương pháp có những ưu điểm riêng trong việc tối đa hóa phạm vi lan truyền thông tin. Greedy Algorithm là phương pháp tham lam, chọn các đỉnh khởi nguồn dựa trên khả năng lan truyền cao nhất trong mỗi bước, đảm bảo rằng mỗi lựa chọn đều tối ưu tại thời điểm đó. Trong khi đó, CELF cải tiến Greedy Algorithm bằng cách giảm thiểu số lượng tính toán cần thiết thông qua việc lưu trữ và sử dụng lại thông tin từ các bước trước, từ đó tăng hiệu quả tính toán đáng kể mà vẫn duy trì hiệu quả lan truyền thông tin tương đương.

Independent Cascade Model (ICM) mô tả quá trình lan truyền thông tin dựa trên xác suất. Mỗi người dùng đã nhận thông tin có một xác suất cố định để lây lan thông tin đến những người dùng kết nối với mình trong mỗi bước thời gian. Nếu người dùng không lan truyền thông tin thành công trong lần đầu tiên, họ sẽ không có cơ hội khác để làm như vậy trong những bước sau.

Linear Threshold Model (LTM) thì dựa trên ngưỡng ảnh hưởng: mỗi người dùng có một ngưỡng nhất định và họ chỉ chấp nhận thông tin khi số lượng bạn bè đã nhận được thông tin vượt qua ngưỡng này. Điều này mô phỏng cách mà người dùng bị ảnh hưởng bởi một số lượng đủ lớn những người trong mạng lưới của họ.

Kết quả của đồ án đã chứng minh rằng việc áp dụng lý thuyết đồ thị và các phương pháp tối ưu lan truyền thông tin có thể cải thiện đáng kể hiệu quả của các chiến dịch marketing, quảng cáo và truyền thông cộng đồng trên mạng xã hội. Những phương pháp này không chỉ giúp xác định các người dùng có ảnh hưởng nhất mà còn tối ưu hóa chi phí và nguồn lực cần thiết để đạt được mục tiêu lan truyền thông tin. Kết luận, nghiên cứu này cung cấp nền tảng vững chắc cho các ứng dụng thực tiễn trong việc quản lý và khai thác mạng xã hội, đồng thời mở ra hướng nghiên cứu mới về các phương pháp tối ưu hóa lan truyền thông tin trong các mạng phức tạp khác.

BẢNG PHÂN CHIA CÔNG VIỆC

Họ Và Tên	Công Việc	Hoàn Thành
Lê Bá Nhất Long 21522300	Tìm hiểu và triển khai 2 thuật toán Greedy và CELF, tìm hiểu bộ dữ liệu	100%
Nguyễn Việt Hoàng 21522095	Tìm hiểu và triển khai mô hình Independent Cascade, phân tích bài báo, tìm hiểu lí thuyết đồ thị cơ bản	100%
Trần Hoàng Phúc 21522479	Tìm hiểu và triển khai mô hình Linear Thresholds Model, làm báo cáo	100%

TÀI LIỆU THAM KHẢO

1. Influence Maximization:

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/07/kdd12-tutorial-inf-part-iii.pdf

2. Information dissemination in socially aware networks under the linear threshold model: https://www.researchgate.net/publication/224225300_Information_dissemination_in_socially_aware_networks_under_the_linear_threshold_model

3. The Independent Cascade and Linear Threshold Models\

https://www.researchgate.net/publication/300470631_The_Independent_Cascade_and_Linear_Threshold_Models

4. Influence Maximization in Independent Cascade Networks

https://amu.hal.science/hal-02373686/document

5. Influence maximization in social networks: Theories, methods and challenges

https://www.sciencedirect.com/science/article/pii/S2590005622000972

Hết