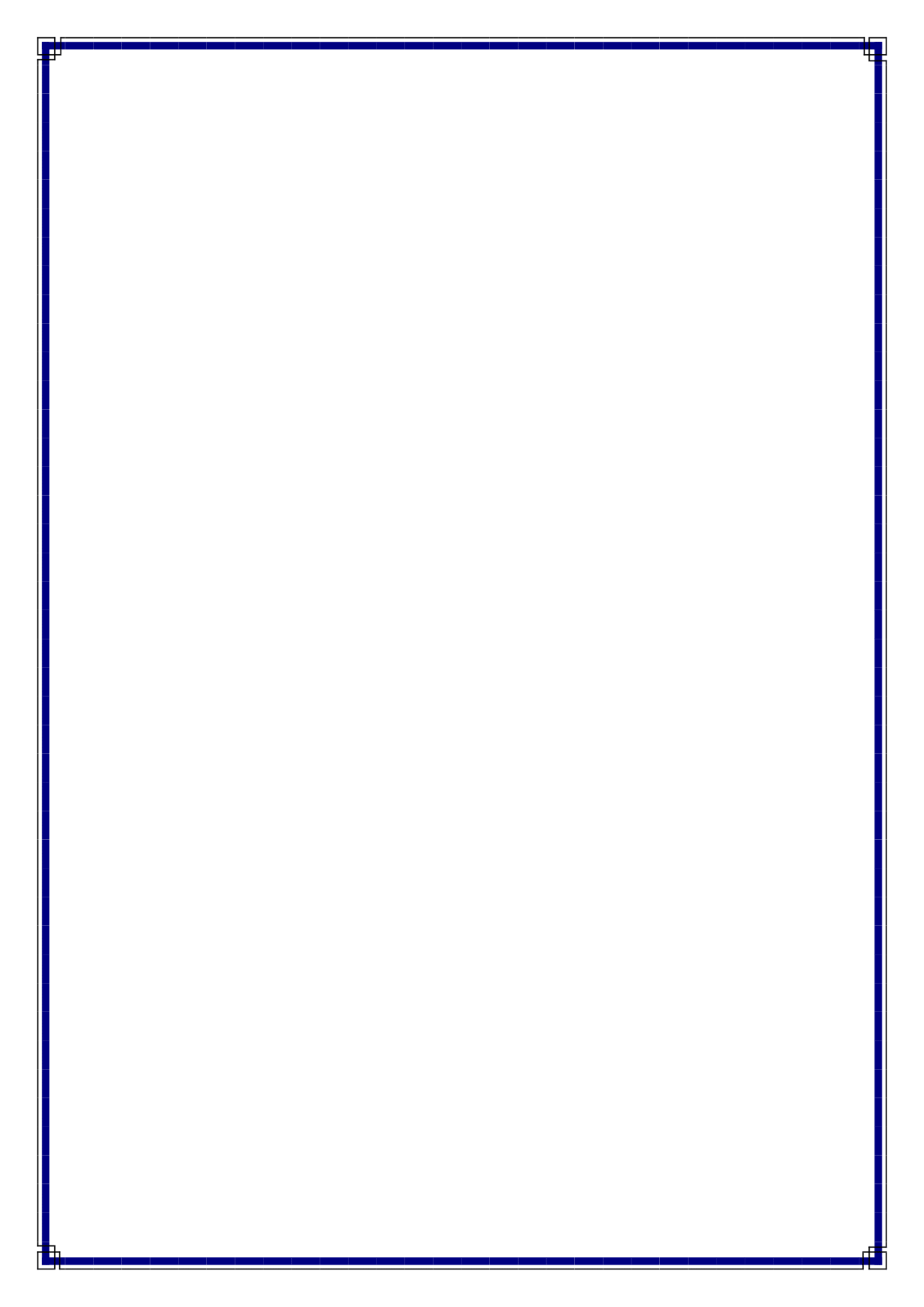
**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**





**BÁO CÁO ĐỒ ÁN**

**MÔN MẠNG XÃ HỘI**

**Đề tài: Phân tích mạng xã hội dựa trên bài báo**

**Social Network Analysis Based on Graph SAGE**

GVHD: ThS. Thái Bảo Trân

Nhóm sinh viên thực hiện:

| 1. | Nguyễn Quốc Huy | MSSV: 21522158 |
| --- | --- | --- |
| 2. | Lê Thị Kiều Lam | MSSV: 21522275 |
| 3. | Trần Phan Thanh Thảo | MSSV: 21522610 |
| 4. | Ngô Anh Tuấn | MSSV: 21521629 |

**Tp. Hồ Chí Minh, 05/2024**

# LỜI CẢM ƠN

Nhóm chúng em muốn bày tỏ lòng tri ân sâu sắc đến Cô – ThS. Thái Bảo Trân, người đã hướng dẫn chúng em trong đồ án môn Mạng xã hội. Với sự am hiểu và sự tận tình, Cô đã dành thời gian và tâm huyết để hướng dẫn chúng em có thể hoàn thành đề tài của mình. Những lời góp ý quý báu và sự chỉ bảo chân thành từ Cô đã giúp chúng em vượt qua những khó khăn, từ đó báo cáo của chúng em trở nên hoàn thiện và chất lượng hơn.

Chúng em cũng biết ơn sự hỗ trợ không ngừng từ Cô trong việc tích luỹ kiến thức cơ bản và nghiên cứu các thông tin mới. Nhờ vào sự hướng dẫn và khuyến khích của Cô, cùng với sự học hỏi từ bạn bè và tài liệu tham khảo, chúng em đã có thể áp dụng những kiến thức đó để hoàn thành báo cáo một cách tốt nhất. Chúng em biết rằng còn nhiều hạn chế và thiếu sót trong báo cáo của mình, chúng em rất trân trọng sự góp ý và chỉ bảo từ Cô để có thể hoàn thiện kỹ năng và kiến thức của mình. Chúng em tin rằng những kinh nghiệm này sẽ là nền tảng quý báu cho bài báo cáo cuối kỳ và những dự án tương lai.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn Cô vì sự hỗ trợ và sự cống hiến của mình. Chúng em cảm thấy rất may mắn được có Cô là người hướng dẫn của mình và chúc Cô luôn mạnh khỏe và thành công trong công việc giảng dạy và nghiên cứu.

Thành phố Hồ Chí Minh, tháng 05 năm 2024

Nhóm sinh viên thực hiện

# NHẬN XÉT CỦA GIẢNG VIÊN

……………………………………………………………………………………………….

……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….……………………………………………………………………………………………….

# MỤC LỤC

[**LỜI CẢM ƠN 2**](#_z4vm625y3l2p)

[**NHẬN XÉT CỦA GIẢNG VIÊN 3**](#_kr739p6h9yzi)

[**MỤC LỤC 4**](#_lkp5crtqrhiq)

[**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 5**](#_xc1vleti2zf0)

[**I. THÔNG TIN BÀI BÁO 5**](#_seqhj02c7er1)

[**II. BÀI TOÁN ĐẶT RA 5**](#_pyfsbdr31i4)

[**III. GIỚI THIỆU CHUNG 5**](#_30anj2ue9qzu)

[**IV. CÁC CÔNG NGHỆ SỬ DỤNG 6**](#_gn54ue77b0hy)

[**CHƯƠNG 2: LOGISTIC REGRESSION 10**](#_toiuznehmihr)

[**CHƯƠNG 3 : THUẬT TOÁN DEEPWALK 13**](#_w429zc5jyrv1)

[**CHƯƠNG 4: GRAPHSAGE 15**](#_xg4tj74svihi)

[**CHƯƠNG 5: PHÂN TÍCH THỰC NGHIỆM 21**](#_ie1ljmeaocgz)

[1. Hướng dẫn cài đặt: 21](#_meyt1nckfd5w)

[2. Demo thuật toán: 21](#_p7o9k2lxttmi)

[3. Thực nghiệm: 21](#_haf1ssnl8bup)

[**CHƯƠNG 6: KẾT LUẬN 23**](#_aqzfpyk426wn)

[**BẢNG PHÂN CÔNG 24**](#_8kc6biyft1k4)

[**TÀI LIỆU THAM KHẢO: 25**](#_8y3o9j8oqss5)

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## THÔNG TIN BÀI BÁO

* Bài báo được đăng ở Hội nghị Quốc tế lần thứ 12 về Thông minh Tính toán và Thiết kế (ISCID) năm 2019.
* Ngày diễn ra Hội nghị: 14-15 tháng 12 năm 2019.
* Ngày được thêm vào IEEE Xplore: 14 tháng 5 năm 2020.
* Nhà xuất bản: IEEE.
* Địa điểm diễn ra Hội nghị: Hàng Châu, Trung Quốc.
* Link bài báo: <https://ieeexplore.ieee.org/document/9092549>

## BÀI TOÁN ĐẶT RA

Được rút ra từ mạng xã hội Sina Weibo, dữ liệu được sử dụng bao gồm thông tin cá nhân của người dùng cũng như các tương tác giữa họ trên nền tảng này. Dữ liệu được chia thành hai phần chính: thông tin người dùng và thông tin các bài đăng. Thông tin người dùng bao gồm các trường như ID người dùng, tên người dùng, số người mà họ đang theo dõi và số người đang theo dõi họ, cùng với danh sách các người mà họ đang theo dõi và danh sách các người đang theo dõi họ.

Thông tin của các bài đăng bao gồm số lượng chia sẻ, bình luận và thích, cũng như ID của người tương tác. Sau khi thu thập và xử lý dữ liệu, một bảng kề của mối quan hệ giữa người dùng và một bảng thông tin người dùng được tạo ra. Mục tiêu của nghiên cứu là dự đoán nhãn của người dùng dựa trên mối quan hệ và thông tin cá nhân.

Nghiên cứu sử dụng mô hình GraphSage để phân tích mạng xã hội, so sánh với các phương pháp phân tích mạng xã hội truyền thống, và thảo luận về giá trị của một số tham số của mô hình. Các cải tiến cũng được thực hiện trong phần lấy mẫu của mô hình để cải thiện hiệu suất của nó.

## GIỚI THIỆU CHUNG

Với sự phát triển nhanh chóng của Internet, mạng xã hội đã trở thành một phần không thể thiếu trong cuộc sống của mọi người. Với sự lan rộng toàn cầu của các nền tảng mạng xã hội trực tuyến và di động, mọi người đã chứng kiến sự ảnh hưởng của phân tích mạng xã hội trong các lĩnh vực như quảng cáo, bầu cử Tổng thống và giám sát ý kiến công cộng. Và thông tin trên mạng xã hội rất đa dạng và phong phú, trong đó thông tin rác xuất hiện không tránh khỏi, gây khó chịu cho người dùng. Đối với nhãn của, việc tìm kiếm chúng sẽ dẫn đến các nền tảng xã hội tốt hơn và cuộc sống xã hội của mọi người.

Mạng neural đồ thị là một công nghệ rất phổ biến trong những năm gần đây. Đồ thị là một cấu trúc dữ liệu được sử dụng rộng rãi, trong đó các nút đại diện cho các đối tượng và các cạnh đại diện cho các mối quan hệ. Nó có ứng dụng rộng rãi trong machine learning như mạng xã hội, kỹ thuật sinh học, Atlas Kiến thức, chủ yếu cho việc phân loại nút, dự đoán liên kết và phân cụm nhiệm vụ khác. Mạng neural đồ thị (GNN) là một mô hình kết nối sử dụng công nghệ mạng neural dựa trên đồ thị để nắm bắt các mối quan hệ trong đồ thị bằng cách truyền tin nhắn giữa các nút đồ thị. Khác với một mạng neural tiêu chuẩn, một mạng neural đồ thị có thể thu thập thông tin từ các nút lân cận ở bất kỳ độ sâu nào xung quanh một nút. Ngoài ra. mạng neural còn có thể xử lý dữ liệu cấu trúc đồ thị.

## CÁC CÔNG NGHỆ SỬ DỤNG

1. Logistic Regression

Logistic regression là một phương pháp thống kê được sử dụng để ước lượng xác suất của một sự kiện xảy ra, như việc bỏ phiếu hoặc không bỏ phiếu, dựa trên một tập dữ liệu cho trước của các biến độc lập. Mô hình này, còn được gọi là mô hình logit, thường được áp dụng cho phân loại và dự đoán. Khi kết quả là xác suất, biến phụ thuộc được giới hạn trong khoảng từ 0 đến 1. Trong logistic regression, một phép biến đổi logit được áp dụng để chuyển đổi tỷ lệ thành công chia cho tỷ lệ thất bại. Điều này còn được gọi là log odds, hoặc logarit tự nhiên của tỷ lệ cơ hội. Hồi quy logistic có thể được biểu diễn bằng các công thức như sau:

Logit(pi) = 1/(1+ exp(-pi))

ln(pi/(1-pi)) = Beta\_0 + Beta\_1X\_1 + … + B\_kK\_k

Trong đó, logit(pi) là biến phụ thuộc hoặc biến phản ứng và X là biến độc lập. Tham số beta, hoặc hệ số, thường được ước lượng thông qua phương pháp ước lượng hợp lý cực đại (MLE), mà kiểm tra các giá trị beta khác nhau qua nhiều lần lặp để tối ưu hóa log odds. Khi có được các hệ số tối ưu, xác suất có điều kiện cho mỗi quan sát có thể được tính toán, đăng nhập và tổng hợp để đưa ra xác suất dự đoán. Trong phân loại nhị phân, xác suất nhỏ hơn 0.5 sẽ dự đoán là 0, trong khi xác suất lớn hơn 0 sẽ dự đoán là 1.

Có ba loại mô hình hồi quy logistic, mỗi loại được định nghĩa dựa trên phản ứng phân loại:

* Hồi quy logistic nhị phân (Binary logistic regression): Biến phụ thuộc chỉ có hai kết quả có thể.
* Hồi quy logistic đa nhóm (Multinomial logistic regression): Biến phụ thuộc có ba hoặc nhiều hơn ba kết quả có thể, mà không có thứ tự cụ thể.
* Hồi quy logistic thứ bậc (Ordinal logistic regression): Biến phụ thuộc có ba hoặc nhiều hơn ba kết quả có thứ tự cụ thể.

Logistic regression được áp dụng rộng rãi cho nhiều vấn đề dự đoán và phân loại, bao gồm:

* Phát hiện gian lận trong dữ liệu.
* Dự đoán khả năng mắc bệnh hoặc ốm đau cho một nhóm dân số cụ thể trong lĩnh vực y học.
* Dự đoán khả năng rời bỏ của khách hàng hoặc nhân viên trong tổ chức để thực hiện các biện pháp phòng ngừa hoặc giữ chân.

1. DeepWalk

DeepWalk là một phương pháp biểu diễn đồ thị không hướng dựa trên học sâu. Nó sử dụng các kỹ thuật học sâu như mạng nơ-ron hồi quy đa lớp để học các biểu diễn vector cho các đỉnh trong một đồ thị. Cụ thể, DeepWalk thực hiện việc này bằng cách lấy mẫu ngẫu nhiên các đường đi ngắn (walks) trong đồ thị, sau đó áp dụng các mô hình Word2Vec để học biểu diễn vector cho các đỉnh dựa trên các từ xuất hiện trong các walks này.

DeepWalk có thể được áp dụng trong nhiều lĩnh vực, bao gồm:

* Phân cụm đỉnh đồ thị: Các biểu diễn vector học được từ DeepWalk có thể được sử dụng để phân cụm đỉnh đồ thị thành các nhóm có tính chất tương tự.
* Dự đoán các kết nối hoặc mối quan hệ trong đồ thị: Sử dụng các biểu diễn vector học được, ta có thể dự đoán xem có mối quan hệ nào giữa các đỉnh không.
* Gợi ý các đỉnh kề cận: DeepWalk có thể được sử dụng để gợi ý các đỉnh kề cận cho một đỉnh cho trước trong đồ thị, ví dụ như gợi ý bạn bè hoặc sản phẩm tương tự.
* Phân tích cấu trúc đồ thị: DeepWalk cung cấp cách tiếp cận để hiểu cấu trúc và tính chất của đồ thị, giúp phân tích mối quan hệ và cấu trúc bên trong của nó.

Quá trình huấn luyện DeepWalk bao gồm các bước sau:

* Tạo walks: Bắt đầu từ một số đỉnh ngẫu nhiên trong đồ thị, tạo ra các walks bằng cách di chuyển ngẫu nhiên qua các cạnh trong một số bước nhất định.
* Xây dựng từ điển: Tạo ra một từ điển các đỉnh trong đồ thị và gán một chỉ số số học cho mỗi đỉnh.
* Huấn luyện mô hình Word2Vec: Sử dụng walks thu thập được để huấn luyện mô hình Word2Vec. Mục tiêu là để học ra các biểu diễn vector cho các đỉnh sao cho các đỉnh có liên quan trong đồ thị sẽ có các biểu diễn vector gần nhau trong không gian vector.
* Thu được biểu diễn vector: Sau khi huấn luyện xong, các biểu diễn vector cho các đỉnh có thể được thu được bằng cách lấy các hàng tương ứng từ ma trận trọng số ẩn của mô hình Word2Vec.

DeepWalk có thể được áp dụng cho các bài toán như phân loại đỉnh đồ thị, dự đoán kết nối giữa các đỉnh, phát hiện cấu trúc cộng đồng trong đồ thị, và nhiều ứng dụng khác trong lĩnh vực phân tích đồ thị và học sâu. Tuy nhiên, cần lưu ý rằng DeepWalk cũng có nhược điểm, bao gồm khả năng xử lý đồ thị lớn và mất mát thông tin về cấu trúc của đồ thị trong quá trình lấy mẫu walks.

1. GraphSAGE

GraphSAGE là một phương pháp học sâu được phát triển để nhúng đồ thị vào không gian vector, từ đó giải quyết các vấn đề liên quan đến đồ thị như phân loại đồ thị, dự đoán liên kết và phát hiện cộng đồng. Ý tưởng chính của GraphSAGE là sử dụng các lớp học sâu để tổng hợp thông tin từ các láng giềng của mỗi đỉnh trong đồ thị và biểu diễn đỉnh đó dưới dạng vector. Quá trình này thường được thực hiện qua nhiều lớp để tạo ra các biểu diễn phức tạp và giàu chi tiết của đồ thị.

Cụ thể, GraphSAGE bao gồm các bước sau:

* Lấy mẫu (Sampling): Lấy một tập hợp các láng giềng của mỗi đỉnh trong đồ thị. Quá trình lấy mẫu này có thể sử dụng các chiến lược ngẫu nhiên hoặc dựa trên độ xa giữa các đỉnh.
* Tổng hợp (Aggregation): Tổng hợp thông tin từ các láng giềng đã lấy mẫu để tạo ra biểu diễn vector cho mỗi đỉnh. Các phương pháp tổng hợp này có thể bao gồm tính trung bình, tổng hoặc sử dụng mạng nơ-ron để học các hàm tổng hợp phức tạp.
* Huấn luyện (Training): Sử dụng các biểu diễn đỉnh được tạo ra từ quá trình tổng hợp để huấn luyện một mô hình học sâu cho các nhiệm vụ như phân loại, dự đoán liên kết hoặc phát hiện cộng đồng.

GraphSAGE đã được áp dụng thành công trong nhiều lĩnh vực, bao gồm mạng xã hội, hóa học và lọc cộng tác. Phương pháp này cho phép hiểu cấu trúc của đồ thị và sử dụng thông tin từ láng giềng để cải thiện hiệu suất của các mô hình học sâu trên dữ liệu đồ thị. Đặc biệt, các lớp tổng hợp trong GraphSAGE được thiết kế để hiệu quả, và nó có thể được tích hợp với các mô hình học sâu khác để thực hiện các nhiệm vụ cụ thể trên đồ thị.

# 

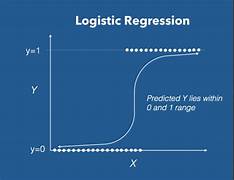
# CHƯƠNG 2: LOGISTIC REGRESSION

1. **Giới thiệu thuật toán:**

Logistic Regression được gọi là phân lớp nhị phân - Binary Classification cụ thể mô hình này dùng để dự đoán output dựa vào các giá trị input đã cho. Hầu hết output của Logistic Regression chỉ có 2 giá trị như: True/False, Yes/No, 0/1,...

* Mô hình máy học có giám sát.
* Sử dụng phương trình tuyến tính cho dự đoán.
* Sử dụng cho các bài toán phân lớp.
* Dự đoán giá trị của biến rời rạc
* Sử dụng hàm độ lỗi Cross Entropy
* Mục đích là tìm ra đường cong (đường cong sigmoid) để phân biệt các biến
* Đầu ra là các giá trị trong khoảng (0,1)

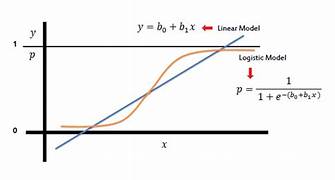
Trong bài báo này, Logistic Regression là mô hình phân loại truyền thống thường được sử dụng trong dựa trên thuật toán phân loại nút mạng xã hội truyền thống trên vòng lặp. Mô hình chỉ sử dụng các đặc điểm của các nút làm đầu vào của bộ phân loại và không sử dụng thông tin cấu trúc của dữ liệu.



1. **Hồi quy Logistic hoạt động như thế nào?**

Hàm sigmoid:

Hồi quy Logistic làm việc dựa trên nguyên tắc của hàm sigmoid – một hàm phi tuyến tự chuyển đầu vào của nó thành xác suất thuộc về một trong hai lớp nhị phân.



* Hàm sigmoid được biểu diễn như sau:

S(z)=1/(1+ )

* Tính chất:

Hàm liên tục, cho giá trị trong khoảng (0,1).

Có đạo hàm trên mọi điểm.

Hàm Sigmoid nhận đầu vào là một giá trị z bất kỳ, và trả về đầu ra là một giá trị xác suất nằm trong khoảng [0,1]. Khi áp dụng vào mô hình Hồi quy Logistic với đầu vào là ma trận dữ liệu X và trọng số w, ta có z=Xw.

Việc huấn luyện của mô hình là tìm ra bộ trọng số w sao cho đầu ra dự đoán của hàm Sigmoid gần với kết quả thực tế nhất. Để làm được điều này, ta sử dụng hàm mất mát (Loss Function) để đánh giá hiệu năng của mô hình. Mô hình càng tốt khi hàm mất mát càng nhỏ.

*\*\* Loss Function: Máy tính học thông qua hàm loss. Đây là một phương pháp đánh giá xem mô hình biểu diễn dữ liệu đã cho tốt đến đâu. Mô hình biểu diễn càng tốt thì giá trị loss càng nhỏ.*

Hàm mất mát (Loss Function) là một hàm số được sử dụng để đo lường mức độ lỗi mà mô hình của chúng ta tạo ra khi dự đoán các kết quả từ dữ liệu đầu vào. Trong bài toán Hồi quy Logistic, chúng ta sử dụng hàm mất mát Cross-Entropy (còn gọi là Log Loss) để đánh giá hiệu năng của mô hình.

*\*\*cross-entropy thể hiện sự sai khác giữa hai phân bố xác suất, thường được sử dụng trong bài toán phân loại nhiều nhãn.*

Hàm Cross-Entropy được thể hiện như sau:

L(w)=

Trong đó:

n: số lượng mẫu dữ liệu trong tập huấn luyện.

: giá trị thực tế của đầu ra thứ i.

: xác suất dự đoán thuộc lớp 1 của mô hình cho đầu vào thứ i.

Hàm Cross-Entropy đo lường khoảng cách giữa hai phân phối xác suất và . Khi mô hình dự đoán chính xác, tức là nếu =1 thì càng gần 1, và nếu = 0 thì càng gần 0, sau đó hàm mất mát sẽ tiến gần về 0.

Trong quá trình huấn luyện, chúng ta tìm cách cập nhật bộ trọng số w sao cho giá trị hàm mất mát Cross-Entropy đạt giá trị nhỏ nhất, dẫn đến một mô hình dự đoán tốt nhất.

Để tìm giá trị tối ưu cho bộ trọng số w, chúng ta có thể sử dụng kỹ thuật Gradient Descent*\*\**.

*\*\*Kỹ thuật Gradient Descent là một phương pháp tối ưu hóa được sử dụng để điều chỉnh các tham số của một mô hình sao cho giá trị của một hàm mất mát được giảm thiểu. Nó hoạt động bằng cách tính gradient của hàm mất mát tại mỗi điểm và di chuyển theo hướng âm của gradient đó với một khoảng cách xác định bởi một tham số gọi là learning rate. Quá trình này được lặp lại cho đến khi đạt được điều kiện dừng, như đạt đến số lần lặp tối đa hoặc gradient gần đủ gần với 0. Gradient Descent có thể áp dụng cho nhiều loại bài toán tối ưu hóa, bao gồm cả tối ưu hóa trong machine learning.*

Tại mỗi bước lặp, chúng ta cập nhật w theo phương tương ứng với đạo hàm của hàm mất mát L(w) theo w.

*\*\*Gradient (độ dốc) là véc tơ biểu diễn tốc độ thay đổi của hàm. Các thành phần của Gradient là đạo hàm riêng theo các biến độc lập của hàm.*

Một số công thức cần chú ý thêm khi làm việc với Hồi quy Logistic:

Xác suất thuộc lớp 1 cho mỗi ví dụ:

P(y=1|X)=S(Xw)

Gradient của hàm mất mát (đạo hàm theo w của hàm mất mát):

∇L(w) =

1. **Hạn chế của Logistic Regression:**

* Khả năng mô hình hóa phức tạp: Logistic Regression chỉ có thể mô hình hóa các quan hệ tuyến tính giữa biến đầu vào và biến mục tiêu, do đó không thể xử lý được các mối quan hệ phi tuyến tính hoặc cấu trúc phức tạp trong dữ liệu đồ thị.
* Giả định về độc lập của các biến: Logistic Regression giả định rằng các biến đầu vào độc lập với nhau, điều này không phù hợp cho dữ liệu đồ thị với các đặc tính mạng lưới phức tạp.
* Khả năng xử lý đặc trưng phiêu lưu (non-Euclidean features): Logistic Regression không thể xử lý các đặc trưng phiêu lưu như cấu trúc đồ thị hay thông tin về láng giềng của các đỉnh.

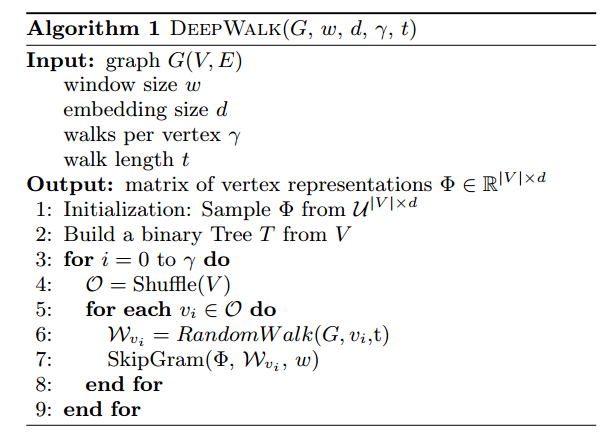
# CHƯƠNG 3: THUẬT TOÁN DEEPWALK

1. **Giới thiệu thuật toán:**

DeepWalk, một phương pháp tiếp cận mới để tìm hiểu các biểu diễn tiềm ẩn của các đỉnh trong mạng. Những biểu diễn tiềm ẩn này mã hóa các mối quan hệ xã hội trong một không gian vectơ liên tục, dễ dàng được các mô hình thống kê khai thác. Deep-Walk khái quát những tiến bộ gần đây trong mô hình hóa ngôn ngữ và học tính năng không giám sát (hoặc học sâu) từ chuỗi từ đến biểu đồ.

DeepWalk sử dụng thông tin cục bộ thu được từ các bước đi ngẫu nhiên được cắt ngắn để tìm hiểu các biểu diễn tiềm ẩn bằng cách coi các bước đi tương đương với các câu.

1. **Hoạt động thuật toán:**

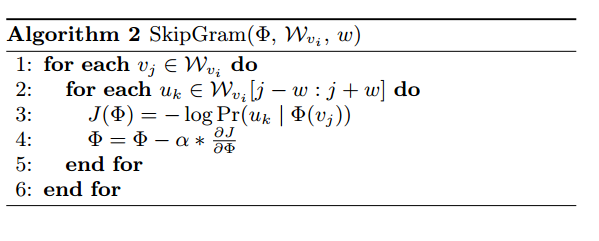


*Hình 3. Mô tả thuật toán DeepWalk*

Thuật toán bao gồm hai thành phần chính: trước tiên là một bộ tạo đi bộ ngẫu nhiên và sau đó là một quy trình cập nhật.

Bộ tạo bước đi ngẫu nhiên(Random walk) trong DeepWalk nhận một đồ thị G và lấy mẫu một đỉnh ngẫu nhiên theo phân phối đồng đều để sử dụng làm gốc cho bước đi ngẫu nhiên . Tiếp theo, một bước đi được lấy mẫu ngẫu nhiên từ các hàng xóm của đỉnh cuối cùng đã được ghé thăm cho đến khi đạt đến chiều dài tối đa t. Trong DeepWalk, không có ràng buộc cụ thể về chiều dài của bước đi, mặc dù trong một số trường hợp, chiều dài này có thể được giữ cố định. Bước đi có thể có khởi đầu lại, nhưng không có bằng chứng nào cho thấy rằng việc này mang lại lợi ích trong các thử nghiệm sơ bộ.

Các dòng 3-9 (*Hình 3)* trong thuật toán thể hiện lõi của phương pháp DeepWalk. Vòng lặp bên ngoài chỉ định số lượt, , mà mỗi đỉnh bắt đầu một bước đi ngẫu nhiên. Mỗi lượt được xem như một lần lặp qua dữ liệu và lấy mẫu một bước đi cho mỗi đỉnh trong suốt lượt này. Ở đầu mỗi lượt, một thứ tự ngẫu nhiên được tạo ra để đi qua các đỉnh, nhằm tăng tốc độ hội tụ của gradient descent ngẫu nhiên. Trong vòng lặp bên trong, tất cả các đỉnh của đồ thị được lặp qua. Đối với mỗi đỉnh , một bước đi ngẫu nhiên được tạo ra, và sau đó được sử dụng để cập nhật biểu diễn của đỉnh đó (Dòng 7). Thuật toán SkipGram được sử dụng để cập nhật các biểu diễn này theo hàm mục tiêu cụ thể.



*Hình 4. SkipGram*

Thuật toán SkipGram nhận vào các tham số là ma trận biểu diễn đỉnh Φ, một bước đi ngẫu nhiên , và một cửa sổ kích thước w.

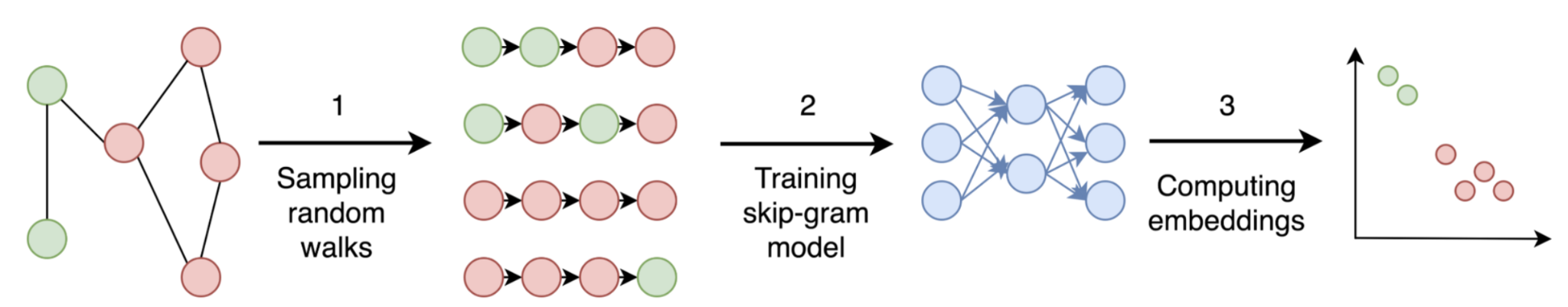
1. Duyệt qua tất cả các đỉnh trong bước đi ngẫu nhiên :

* Duyệt qua các từ hàng xóm trong cửa sổ kích thước 𝑤 xung quanh .
* Tính hàm mất mát 𝐽(Φ) dựa trên xác suất 𝑃𝑟(∣Φ()) của từ hàng xóm dựa trên biểu diễn của đỉnh .
* Cập nhật ma trận biểu diễn Φ bằng cách sử dụng gradient descent, giảm bớt 𝛼 lần độ dốc của hàm mất mát 𝐽(Φ)

1. Lặp lại quá trình cho tất cả các đỉnh trong bước đi ngẫu nhiên ​

​Thuật toán SkipGram trong DeepWalk có mục tiêu là cập nhật ma trận biểu diễn sao cho xác suất xuất hiện của các từ hàng xóm được tối ưu hóa dựa trên biểu diễn của đỉnh trong không gian vectơ. Bằng cách này, các biểu diễn của các đỉnh trong đồ thị có thể được cập nhật để phản ánh mối quan hệ với các từ hàng xóm của chúng.

SkipGram là một phương pháp được sử dụng trong mô hình ngôn ngữ để tối ưu hóa xác suất xuất hiện của các từ cùng nhau trong một cửa sổ văn bản nhất định. Thuật toán này hoạt động bằng cách duyệt qua tất cả các cặp từ liên tiếp trong một bước đi ngẫu nhiên và cố gắng tối đa hóa xác suất của các từ hàng xóm trong cửa sổ xung quanh mỗi từ. Đối với mỗi cặp từ, chúng ta ánh xạ mỗi đỉnh trong đồ thị về vector biểu diễn hiện tại của nó Φ(). Sau đó, chúng ta cố gắng tối đa hóa xác suất xuất hiện của các đỉnh láng giềng của ​ trong bước đi ngẫu nhiên. Để làm điều này, chúng ta có thể sử dụng một số lựa chọn của các bộ phân loại để học phân phối hậu nghiệm. Các biểu diễn của các đỉnh trong đồ thị được cập nhật để phản ánh mối quan hệ với các từ hàng xóm của chúng, giúp cải thiện hiệu suất của mô hình trong việc dự đoán các từ liên quan.



*Hình 6.DeepWalk*

1. **Hạn chế của DeepWalk:**

* Khả năng hiểu cấu trúc đồ thị: DeepWalk dựa vào việc nhúng đồ thị bằng cách tạo ra các biểu diễn đỉnh từ các dự đoán của mô hình dự đoán từ ngữ (language model), điều này không cho phép nắm bắt được cấu trúc toàn diện của đồ thị.
* Khả năng xử lý thông tin từ láng giềng: DeepWalk không thực hiện tổng hợp thông tin từ các láng giềng của các đỉnh, do đó có thể bỏ qua thông tin quan trọng từ mối quan hệ giữa các đỉnh trong đồ thị.
* Khả năng mô hình hóa độ phức tạp: DeepWalk không thể mô hình hóa được cấu trúc phức tạp của dữ liệu đồ thị, đặc biệt là trong các mô hình học sâu cần phải xử lý những biểu diễn có kích thước lớn.

# 

# CHƯƠNG 4: GRAPHSAGE

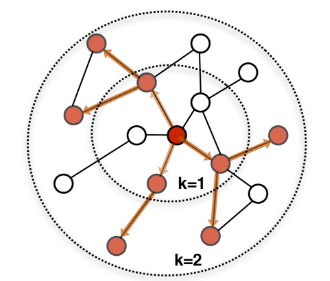
1. **Giới thiệu thuật toán:**

GraphSAGE là một khung để học biểu diễn quy nạp trên các biểu đồ lớn. GraphSAGE được sử dụng để tạo biểu diễn vectơ chiều thấp cho các nút và đặc biệt hữu ích cho các biểu đồ có thông tin thuộc tính nút phong phú.

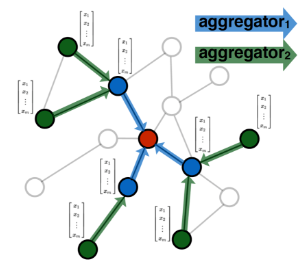
Trong bài báo, GraphSAGE được sử dụng để phân loại nút trong mạng xã hội.

1. **Hoạt động của thuật toán:**

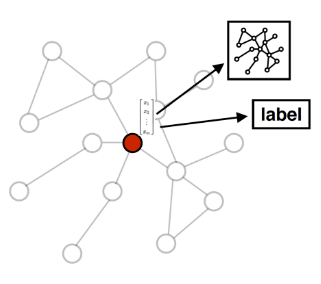
Khung làm việc graphSAGE được thể hiện lần lượt 3 ảnh 7,8,9:



*Hình 7. Vùng lân cận mẫu*

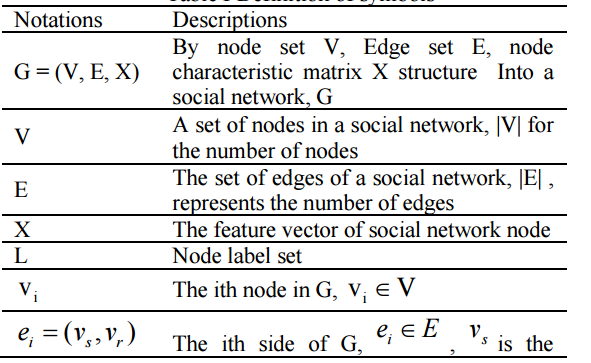
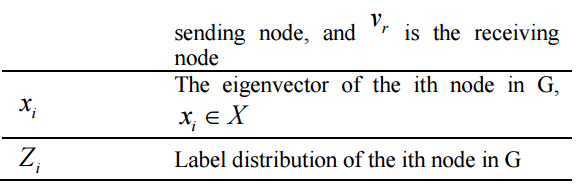


*Hình 8.Tổng hợp thông tin tính năng từ hàng xóm*



*Hình 9.Dự đoán nhãn quảng cáo ngữ cảnh biểu đồ bằng thông tin tổng hợp*

Thành lập bảng ký hiệu mô tả:



Quá trình lan truyền tin nhắn: mục tiêu là lan truyền thông tin của nút tới các nút láng giềng và miền, đồng thời khai thác thông tin tương tác giữa các nút. Đối với một mạng xã hội đã cho G=(V, E, X). Các nút kề của mỗi nút được lấy mẫu và tập hợp các tập đặc trưng bằng một hàm tập hợp.



Lúc này, chứa thông tin về các hàng xóm và thông tin đặc trưng của các nút kề cạnh.

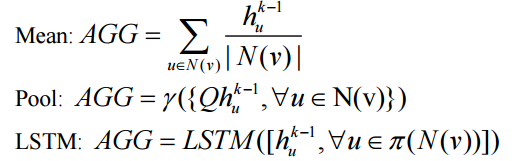
ở đây là một mẫu của các nút hàng xóm.

Trong GraphSAGE, tại mỗi bước thời điểm, các nút hàng xóm được sử dụng để xây dựng biểu diễn cho một nút trong đồ thị. Đối với mỗi nút, ta chọn một mẫu kích thước cố định hoặc biến đổi của các hàng xóm, đảm bảo hiệu quả tính toán và xử lý dữ liệu trong các trường hợp nút có số lượng hàng xóm lớn. Bởi vì các hàng xóm không được xem xét theo thứ tự, việc lấy mẫu này không cần phải quan tâm đến sự hợp lệ.

GraphSAGE cung cấp ba loại hàm tổng hợp để tạo ra biểu diễn cho mỗi nút dựa trên các biểu diễn của các hàng xóm. Các hàm tổng hợp này bao gồm:

* Hàm tổng hợp Mean: Tính trung bình của các biểu diễn của các hàng xóm.
* Hàm tổng hợp Pool: Tổng hợp thông tin từ các hàng xóm bằng cách áp dụng một hàm tổng hợp cụ thể, ví dụ như max-pooling hoặc mean-pooling.
* Hàm tổng hợp LSTM: Sử dụng một mạng nơ-ron LSTM để tổng hợp thông tin từ các hàng xóm theo thứ tự.

Các hàm tổng hợp này được sử dụng để tạo ra biểu diễn mới cho mỗi nút, dựa trên thông tin của các hàng xóm của nó.



Để khai thác thông tin tương tác giữa các nút ở hai đầu của một cạnh, chúng ta cần sử dụng thông tin của cả hai nút để biểu diễn đặc trưng của cạnh. Bởi vì thông tin của cạnh mật thiết liên quan đến thông tin của các nút kết nối ở hai đầu của cạnh đó, ta sử dụng các vectơ riêng của các nút ở hai đầu của cạnh để đại diện cho các đặc trưng của cạnh. Việc kết hợp các thông tin này có thể được thực hiện bằng cách sử dụng các Vector splicing, average pooling, maximum pooling, hoặc summation. Hàm nối vectơ có thể giữ thông tin của hai nút tốt hơn. Ở đây, chúng ta kết hợp thông tin đã tổng hợp với thông tin của nút trung tâm.



Sau khi kết hợp thông tin, chúng ta nhúng nó bằng công thức sau đây:



Trong đó:

là một hàm kích hoạt phi tuyến tính

là một ma trận trọng số.

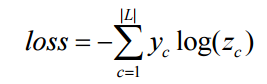
Sự kết hợp của chúng có thể được xem như một lớp mạng nơ-ron đơn. Mặc dù có thể sử dụng mạng nơ-ron nhiều lớp, nhưng việc sử dụng mạng nơ-ron đơn lớp thường mang lại kết quả tốt hơn. Mỗi nút trong đồ thị thực hiện các tính toán được mô tả trước đó, và mỗi nút thu được thông tin về các hàng xóm và cấu trúc của nó. Sau đó, chúng ta tiến hành quá trình tích hợp và nhúng, giúp mỗi nút có thể tích hợp thông tin từ xa so với nút trung tâm. Phạm vi tích hợp phụ thuộc vào số lần lặp. Sau mỗi vòng lặp tính toán, các đặc trưng kết quả cần được chuẩn hóa theo công thức:



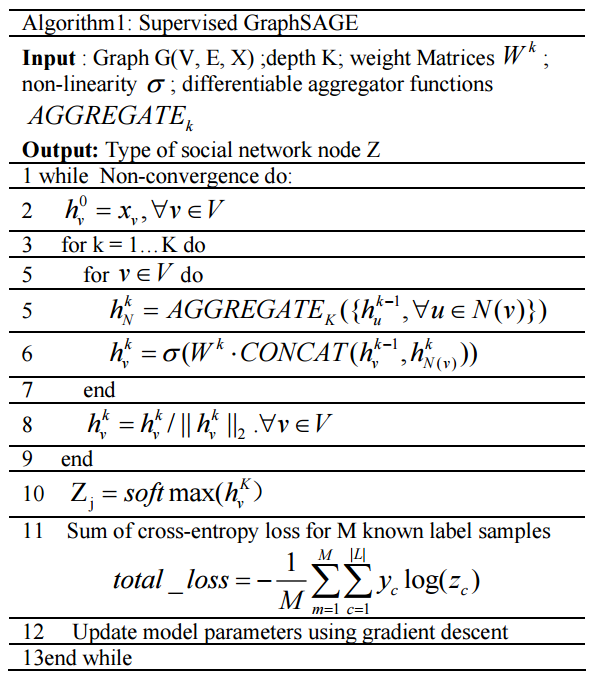
Sau phương pháp tính toán như trên, chúng ta có thể thu được một vector cho mỗi nút và một số lượng cố định của các nút kề cận. Vector này chứa thông tin về các nút kề cận và cấu trúc của chúng ở một số lượng cố định. Sau đó, vector của nút mục tiêu sẽ là vectơ cuối cùng. Xác suất của mỗi lớp có thể được tính bằng hàm Softmax.



Trong mô hình này, độ mất mát của mô hình được tính bằng phương pháp Cross entropy, và công thức như sau:



Mô tả thuật toán:



Tất cả các đỉnh v trong tập đỉnh V của đồ thị được khởi tạo ngẫu nhiên.

* Đối với mỗi cấp độ lặp K(dòng 3\_9):
  + Đối với mỗi đỉnh v trong tập V(dòng 5\_6):
    - Tính toán biểu diễn v mới bằng cách kết hợp thông tin từ các đỉnh hàng xóm của nó trong bước lặp trước, sử dụng hàm tổng hợp.
  + Tính toán biểu diễn của tất cả các đỉnh v của đồ thị sau K lần lặp.

Tính toán độ lỗi tổng cộng cho M mẫu nhãn đã biết bằng cách tính tổng các mất mát chéo-entropy giữa nhãn thực tế và dự đoán được tính từ biểu diễn của các đỉnh.

Cập nhật các tham số mô hình bằng cách sử dụng \*\*gradient descent dựa trên độ lỗi tính toán được.

*\*\* Gradient Descent là một phương pháp tối ưu hóa được sử dụng để điều chỉnh các tham số của một mô hình sao cho giá trị của một hàm mất mát được giảm thiểu. Nó hoạt động bằng cách tính gradient của hàm mất mát tại mỗi điểm và di chuyển theo hướng âm của gradient đó với một khoảng cách xác định bởi một tham số gọi là learning rate. Quá trình này được lặp lại cho đến khi đạt được điều kiện dừng, như đạt đến số lần lặp tối đa hoặc gradient gần đủ gần với 0. Gradient Descent có thể áp dụng cho nhiều loại bài toán tối ưu hóa, bao gồm cả tối ưu hóa trong machine learning.*

Tiếp tục lặp lại cho đến khi tiêu chí hội tụ được đáp ứng.

1. **Ưu điểm của GraphSAGE:**

GraphSAGE có nhiều ưu điểm so với các phương pháp khác trong việc xử lý dữ liệu đồ thị phức tạp.

* Khả năng tổng hợp thông tin từ láng giềng: GraphSAGE có khả năng tổng hợp thông tin từ các láng giềng của mỗi đỉnh trong đồ thị. Điều này cho phép mô hình hóa các mối quan hệ phức tạp giữa các đỉnh trong đồ thị và tạo ra các biểu diễn đỉnh có tính khái quát và giàu chi tiết.
* Xử lý các đặc trưng đồ thị: GraphSAGE có thể xử lý được các đặc trưng đồ thị như cấu trúc đồ thị và thông tin về láng giềng của các đỉnh. Điều này cho phép mô hình hóa toàn diện các mối quan hệ trong dữ liệu đồ thị.
* Tính linh hoạt và linh động: GraphSAGE có thể được điều chỉnh và tinh chỉnh để phù hợp với nhiều loại dữ liệu đồ thị khác nhau. Các phương pháp lấy mẫu, tổng hợp và mô hình hóa có thể được điều chỉnh để đáp ứng các yêu cầu cụ thể của từng bài toán.
* Hiệu suất cao: GraphSAGE thường mang lại hiệu suất cao trên các tập dữ liệu đồ thị phức tạp. Các biểu diễn đỉnh sinh ra từ GraphSAGE thường có khả năng biểu diễn chính xác và toàn diện về cấu trúc và mối quan hệ trong đồ thị.
* Áp dụng rộng rãi: GraphSAGE đã được chứng minh là hiệu quả trong nhiều lĩnh vực, bao gồm mạng xã hội, hóa học, lọc cộng tác và nhiều lĩnh vực khác. Điều này chỉ ra tính đa dạng và linh hoạt của GraphSAGE trong việc giải quyết nhiều loại vấn đề trên dữ liệu đồ thị.

Tóm lại, GraphSAGE là một phương pháp mạnh mẽ và linh hoạt trong việc xử lý dữ liệu đồ thị phức tạp, với khả năng tổng hợp thông tin từ láng giềng và xử lý các đặc trưng đồ thị một cách hiệu quả.

# CHƯƠNG 5: PHÂN TÍCH THỰC NGHIỆM

## Hướng dẫn cài đặt

Cài đặt các thư viện:

Mở cmd( run as administrator) và chạy câu lệnh sau:

pip install torch torch-geometric scikit-learn node2vec

## Giới thiệu dữ liệu và link demo thuật toán

Dữ liệu dùng trong thuật toán:

Bộ dữ liệu Cora bao gồm 2708 ấn phẩm khoa học được phân loại thành một trong bảy lớp. Mạng trích dẫn bao gồm 5429 liên kết. Mỗi ấn phẩm trong tập dữ liệu được mô tả bằng vectơ từ có giá trị 0/1 cho biết sự vắng mặt/hiện diện của từ tương ứng trong từ điển. Từ điển bao gồm 1433 từ độc đáo.

Tập dữ liệu này bao gồm các bài báo khoa học như các nút, và các cạnh giữa các nút đại diện cho các mối quan hệ trích dẫn giữa các bài báo. Mỗi nút có các đặc trưng (features) và một nhãn (label) tương ứng với chủ đề của bài báo.

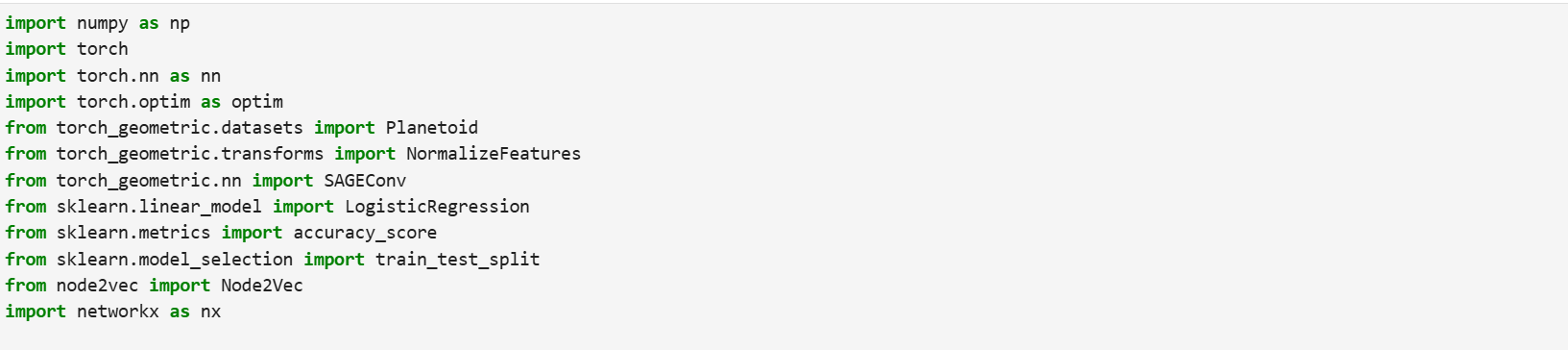
Link data: <https://paperswithcode.com/dataset/cora>

Link code và demo thuật toán Logistic Regression, DeepWalk, GraphSAGE: <https://drive.google.com/drive/folders/14hClI6R2y0lpvPT2TUhkFhDy2qJiXiHk?usp=sharing>

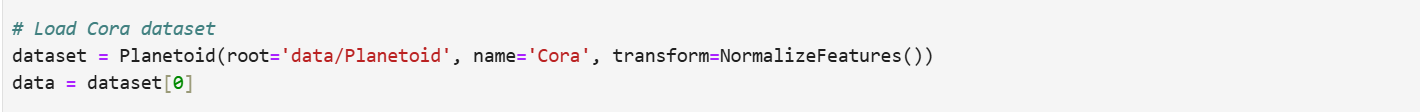
## Thực nghiệm

Trong thực nghiệm này, chúng tôi dùng 3 thuật toán Logistic Regression, DeepWalk và GraphSAGE để giải cùng một bài toán phân loại nút trên Cora Dataset, nút ở đây sẽ là các bài báo phân loại theo 7 chủ đề. Mục tiêu: xác định thuật toán nào sẽ đạt độ chính xác cao nhất với bài toán phân loại.

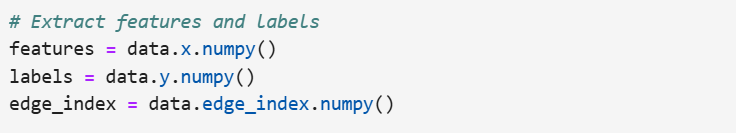
Import các thư viện:



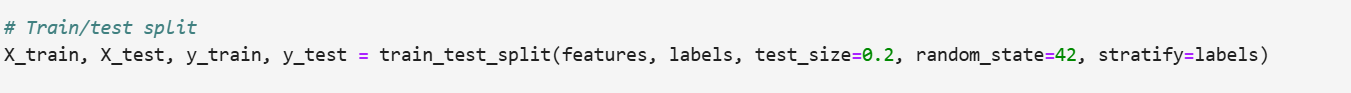
Load data:



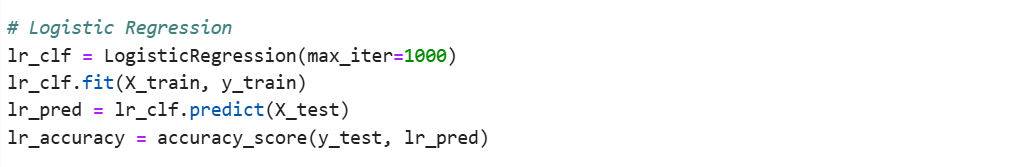
Trích xuất đặc trưng và nhãn:



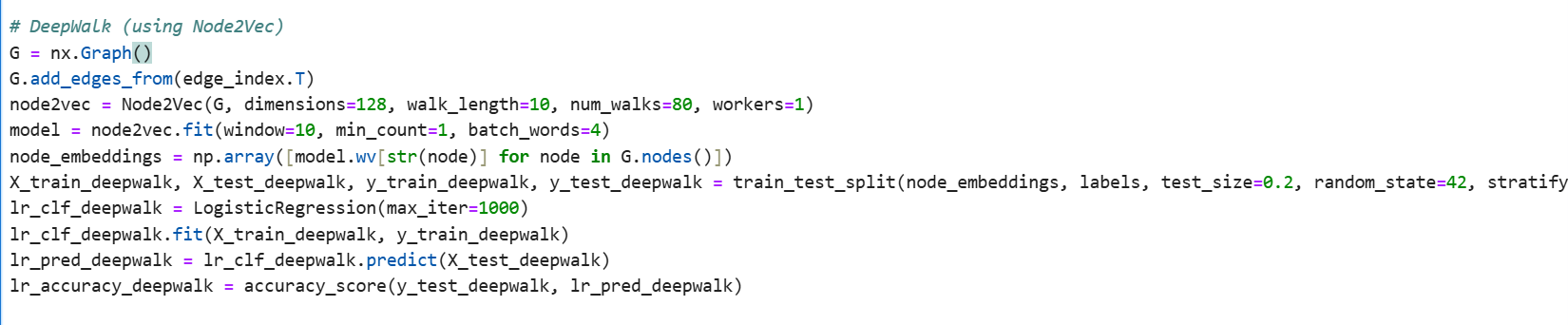
Chia dữ liệu huấn luyện và kiểm tra:



Logistic Regression:



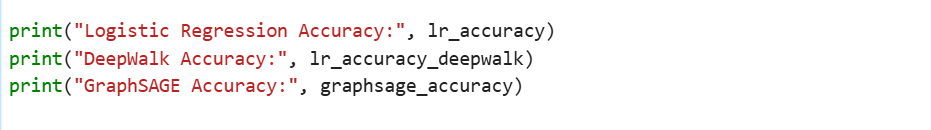
DeepWalk:



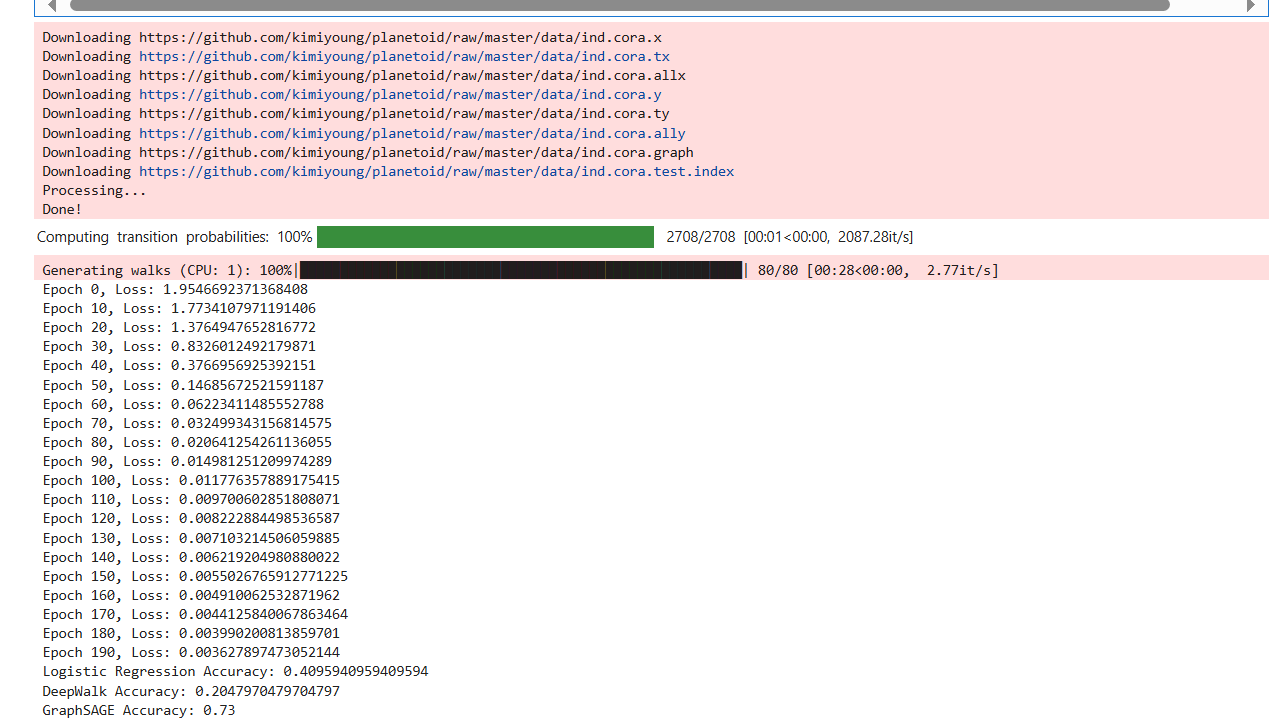
GraphSAGE:



In ra kết quả:

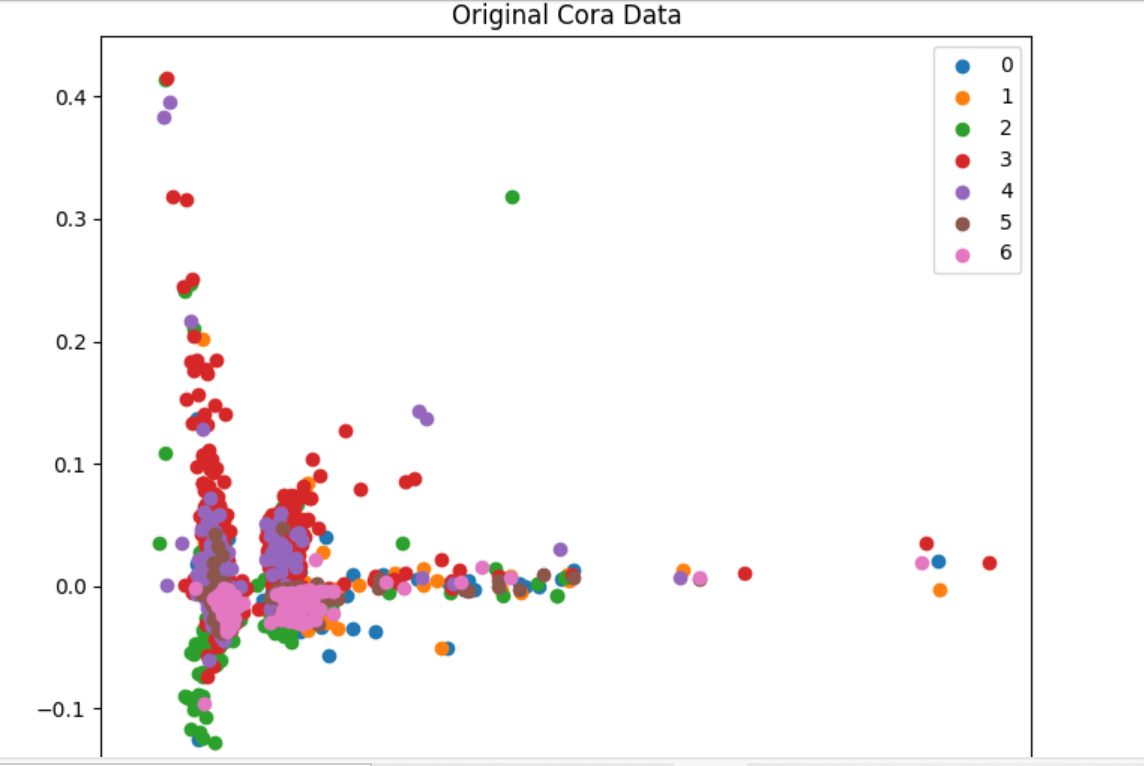


Kết quả sau khi chạy 3 thuật toán:

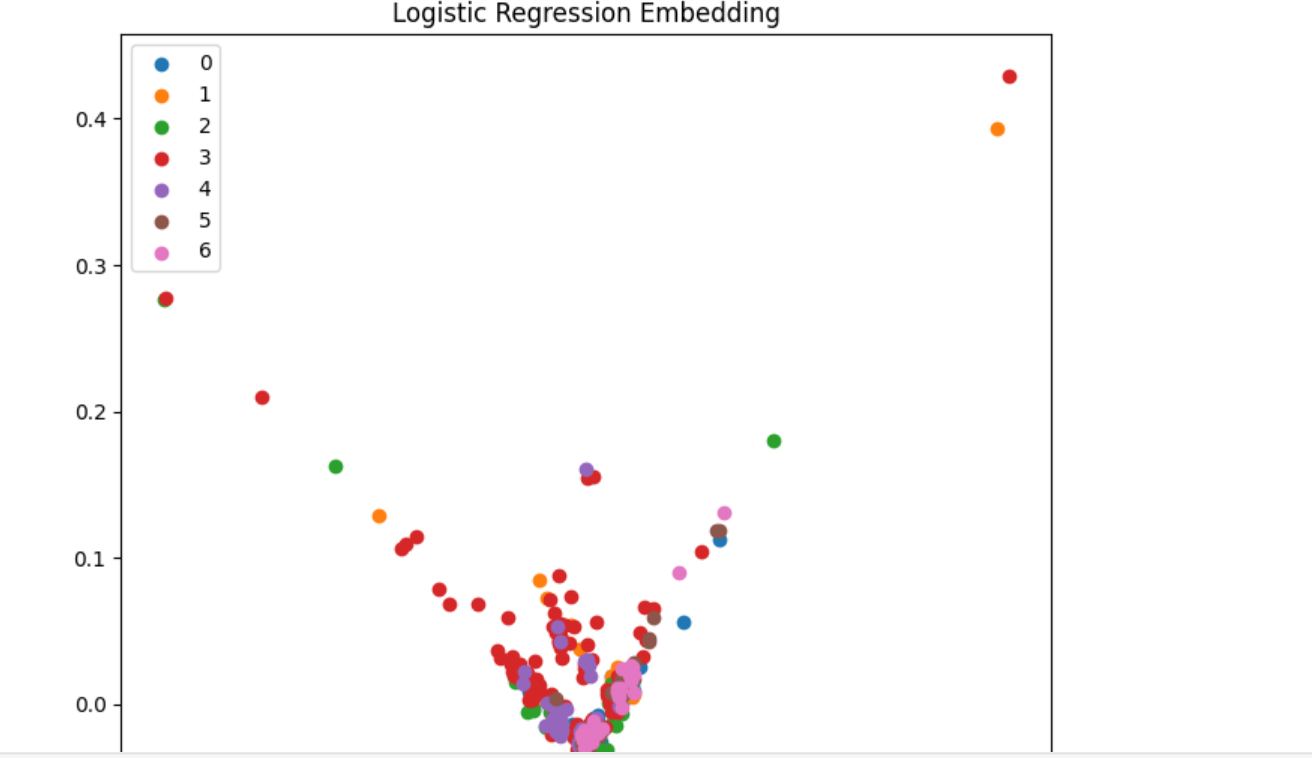


Minh họa bằng hình ảnh:

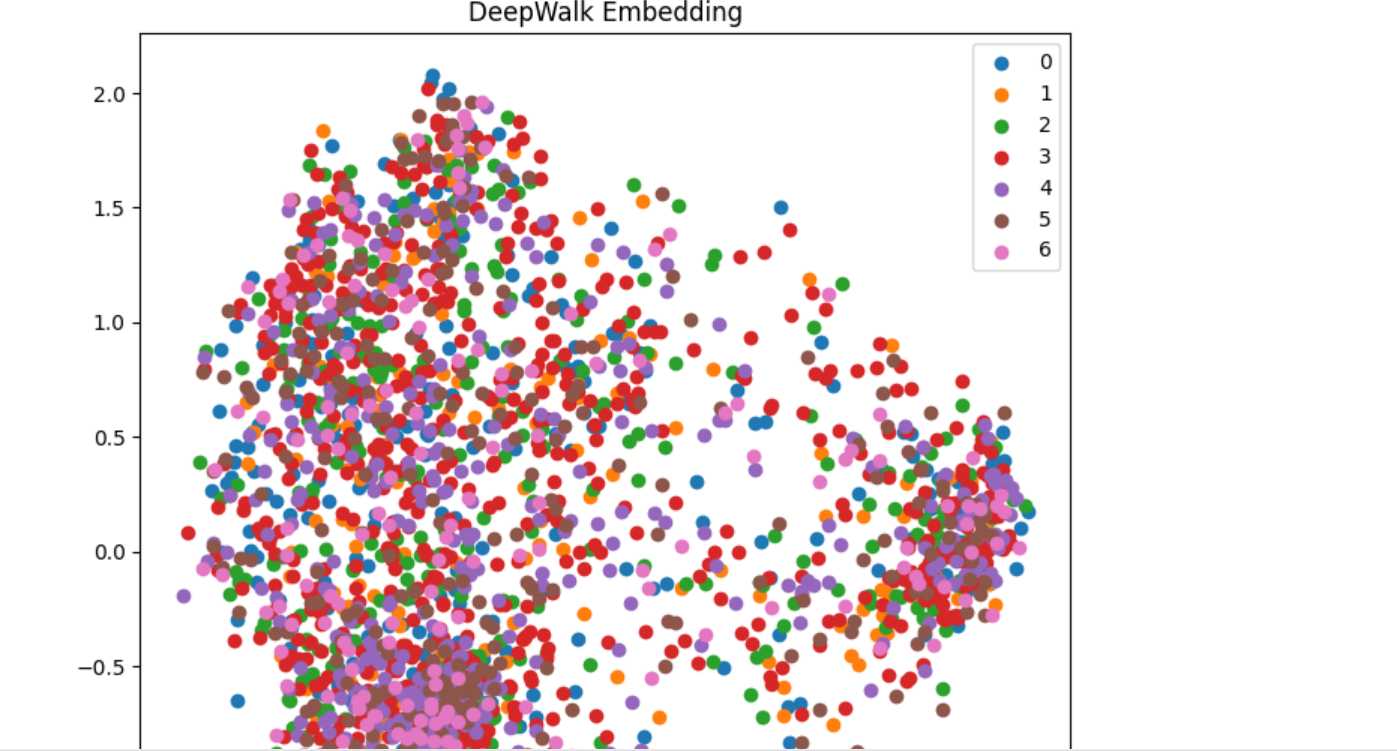
Dataset ban đầu: ban đầu chưa thấy được sự phân loại của các bài báo.



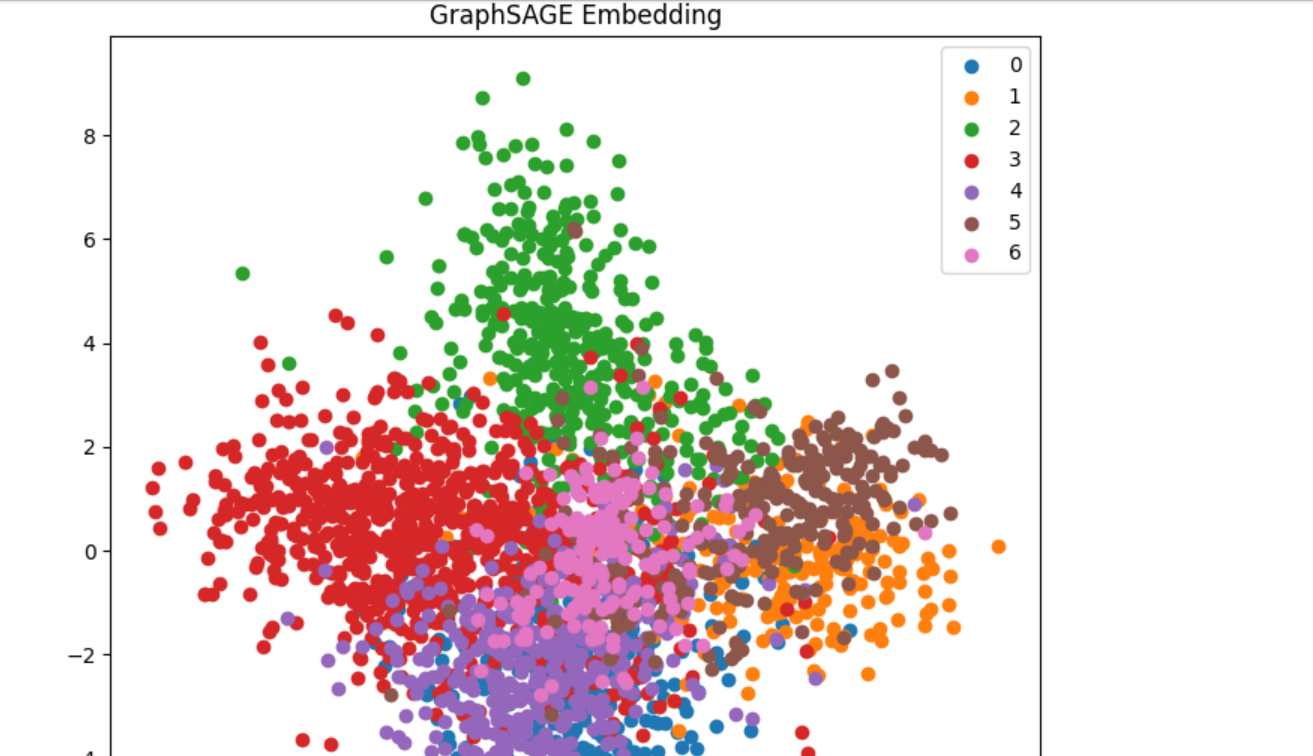
Khi áp dụng Logistic Regression thấy nút( các phài báo) có được phân loại nhưng không đáng kể.



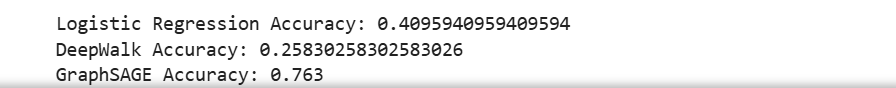
Với DeepWalk, hầu như không thể thấy được cái nút đã được phân loại theo từng chủ đề.



Với GraphSAGE, các nút hầu như được phân loại theo đúng chủ đề của nó.



Kết quả cuối cùng thấy được GraphSAGE mang giá trị accuracy lớn nhất, tiếp đền là Logistic Regression và cuối cùng là DeepWalk. Từ đó có thể nói rằng, GraphSAGE mang lại sự chính xác cao hơn DeepWalk và Logistic Regression trong cùng một bài toán phân loại nút.



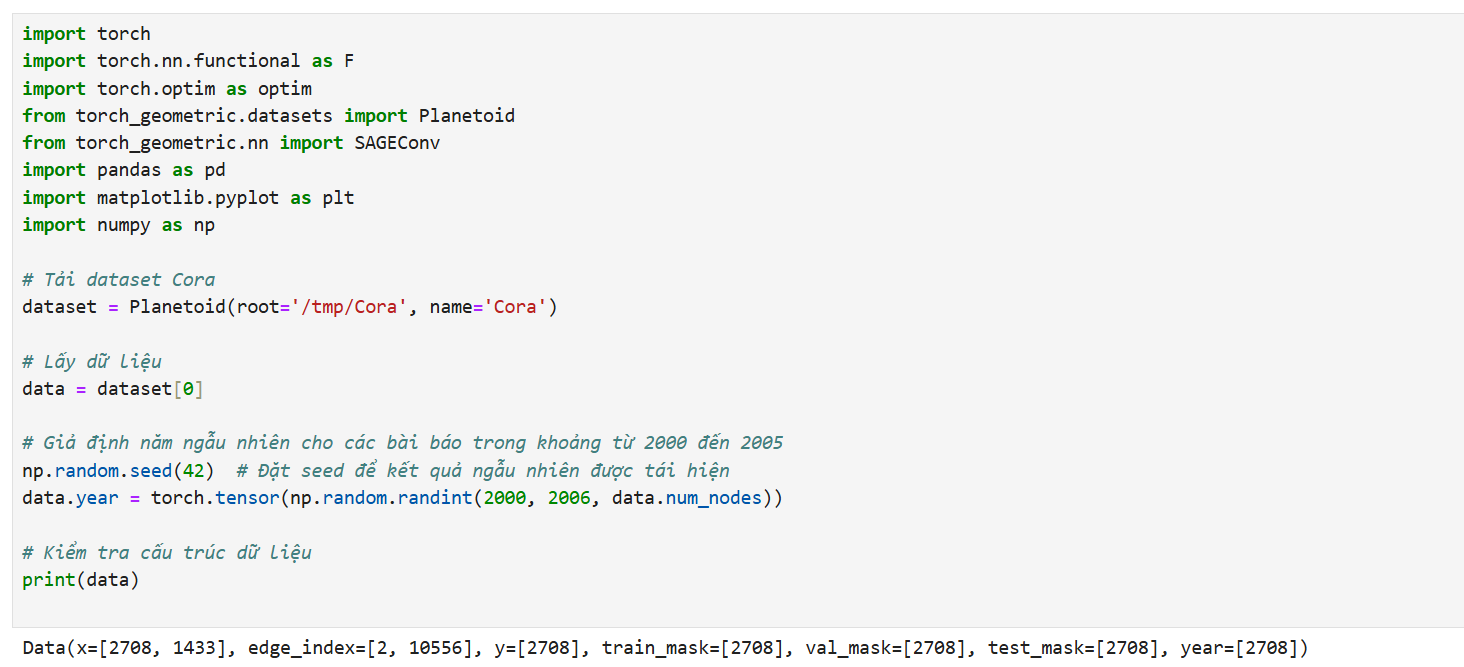
***➤ Mở rộng cho phần thực nghiệm của thuật toán:***

Bài toán: Sử dụng Cora dataset để phân loại các bài báo thành các chủ đề khác nhau theo từng năm bằng cách sử dụng thuật toán GraphSAGE và thư viện PyTorch Geometric. Tính tỷ lệ của từng chủ đề trong mỗi năm và biểu diễn kết quả.

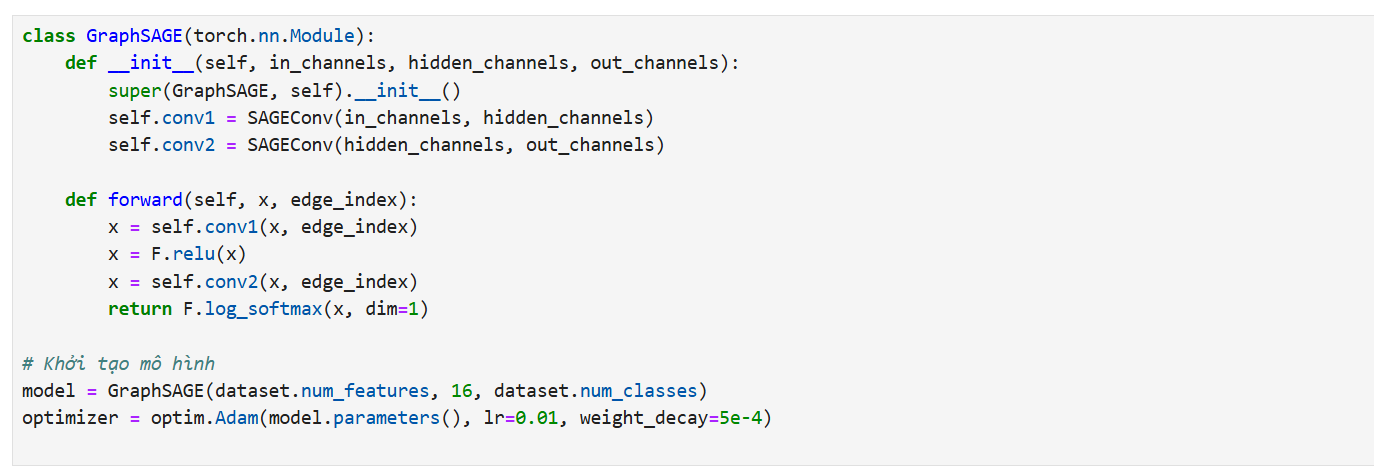
Mục tiêu: Phân loại nút theo từng chủ đề theo từng năm để có sự đánh giá nhận xét sự thay đổi theo từng năm( giả định từ năm 2000 đến 2005).

Các bước thực hiện:

Import các thư viện và chuẩn bị dữ liệu



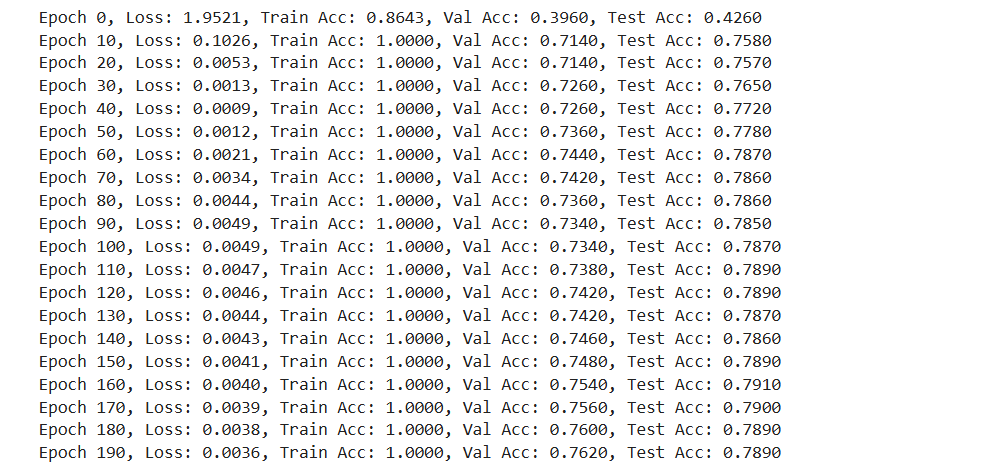
Xây dựng mô hình graphSAGE



Tiến hành huấn luyện mô hình và phân loại nút



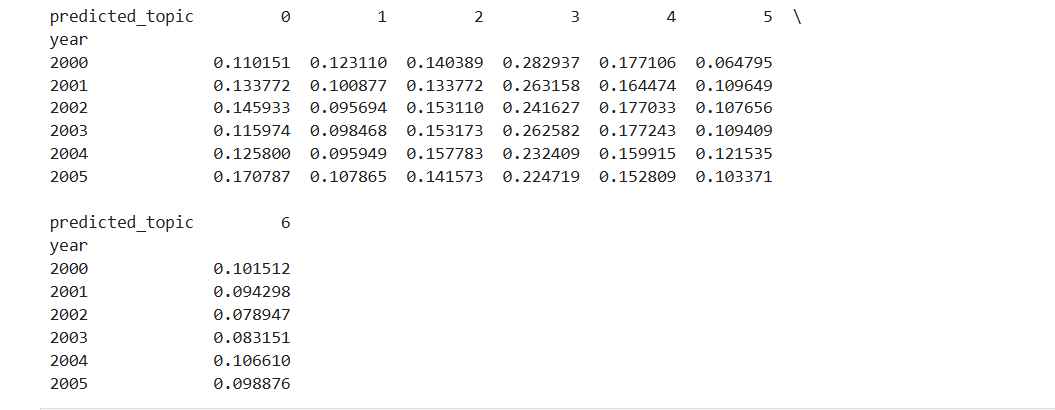
Kết quả sau quá trình này sẽ là:



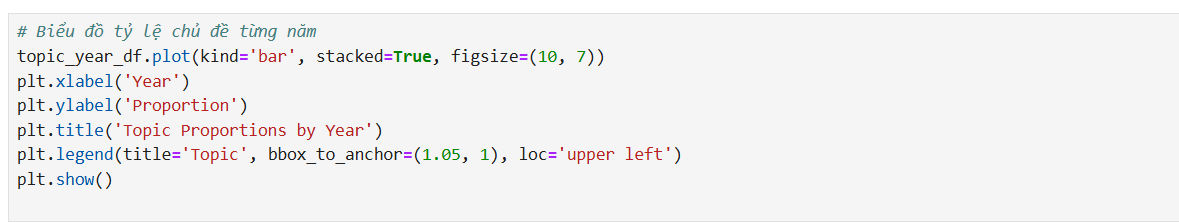
Phân tích kết quả và tính tỷ lệ của từng chủ đề theo từng năm



Kết quả:



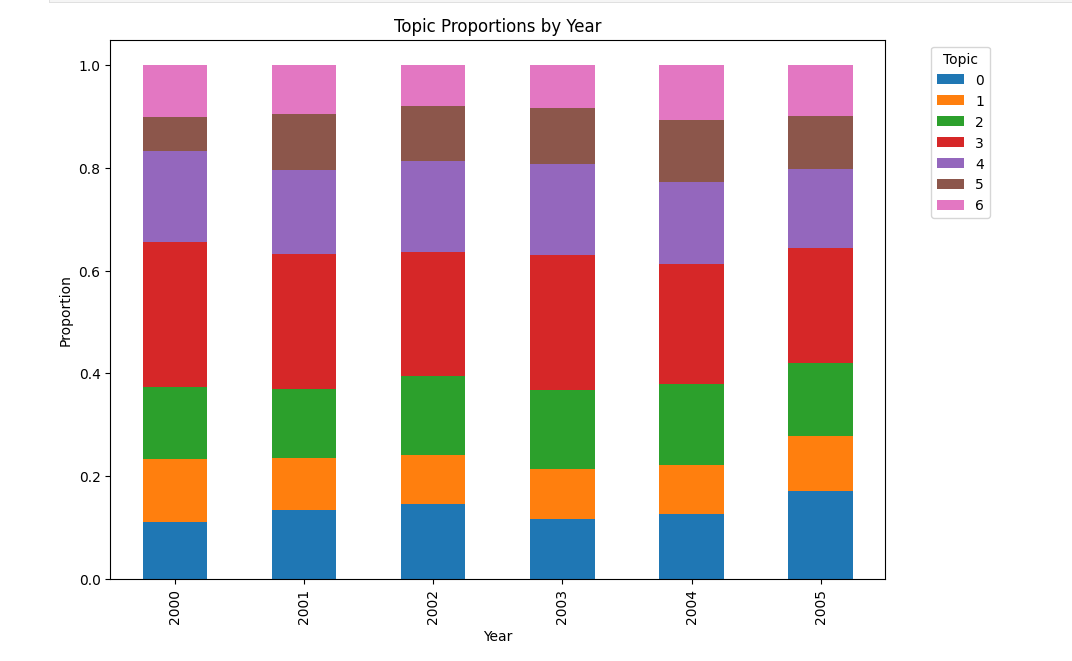
Sau đó biểu diễn ở dạng đồ thị để mang đến cái nhìn trực quan hơn.



Kết quả cuối cùng ta thấy được, tỉ lệ các loại bài báo theo từng chủ đề có sự thay đổi tăng giảm theo từng năm. Theo từng năm, có số lượng bài báo của từng chủ đề sẽ có sự thay đổi tỉ lệ.

Ví dụ: chẳng hạn như từ năm 2000 đến năm 2001 thì tỉ lệ của những bài báo có chủ đề 0 có xu hướng tăng, từ đó ta có thể nhận định chủ đề 0 có xu hướng được quan tâm ở năm 2001 hơn so với năm 2000.

Trong các năm ta thấy các bài báo ở chủ đề 3 luôn nhận được sự quan tâm khá nhiều của các chuyên gia nghiên cứu. Tuy nhiên càng về sau, tỉ lệ có sự sụt giảm và dần được chia sẻ cho các chủ đề còn lại.



# CHƯƠNG 6: KẾT LUẬN

Việc phân loại các nút mạng xã hội có giá trị ứng dụng quan trọng trong đời sống thực.Việc phân loại nút hỗ trợ rất nhiều trong quá trình nghiên cứu cũng như ở đa dạng các lĩnh vực.

Phân loại ứng dụng các thuật toán truyền thống dựa trên thuật toán lặp và truyền nhãn dựa trên bước đi ngẫu nhiên không xem xét đầy đủ thông tin lẫn nhau giữa các nút. Với các thuật toán đồ thị trích xuất các tính năng nút cấp cao hơn theo thông tin tương tác với các nút lân cận và các đặc điểm thuộc tính riêng của nút và cuối cùng phân loại các nút.

Đồ án được xây dựng với mục đích tìm hiểu những thuật toán mới trong việc phân loại nút mạng xã hội: Logistic Regression, mở rộng ở các thuật toán deep learning đòi hỏi độ khó cao như DeepWalk, GraphSAGE .Sau đó sử dụng mô hình đồ thị để phân tích mạng xã hội và so sánh với phương pháp phân tích mạng xã hội truyền thống.

Qua quá trình thực nghiệm, nhóm đã nhận được kết quả tương tự như bài báo đã đưa ra thuật toán graphSAGE mang lại kết quả vượt trội hơn so với các thuật toán như DeepWalk hay là Logistic Regression.

Sau khi thực hiện đồ án, nhóm đã tích lũy nhiều kiến thức về các thuật toán mạng xã hội, là tiền đề để phát triển sau này.

# 

# BẢNG PHÂN CÔNG

| MSSV | Họ và tên | Công việc thực hiện | Đánh giá |
| --- | --- | --- | --- |
| 21522158 | Nguyễn Quốc Huy | Thực hiện thuật toán DeepWalk,thuyết trình, demo DeepWalk | Hoàn thành |
| 21521629 | Ngô Anh Tuấn | Thực hiện thuật toán Logistic Regression, thuyết trình, demo Logistic Regression | Hoàn thành |
| 21522610 | Trần Phan Thanh Thảo | Tổng hợp báo cáo, slide thuyết trình, update đồ án. | Hoàn thành |
| 21522275 | Lê Thị Kiều Lam | Thực hiện thuật toán GraphSAGE, thuyết trình và demo, thực hiện update đồ án. | Hoàn thành |

# 

# TÀI LIỆU THAM KHẢO:

[1]DeepWalk\_Link truy cập:<https://paperswithcode.com/method/deepwalk>

[2]Logistic Regression\_Link truy cập: <https://aws.amazon.com/vi/what-is/logistic-regression/>

[3]Trí tuệ nhân tạo-Logistic Regression\_Link truy cập:<https://trituenhantao.io/machine-learning-co-ban/bai-6-logistic-regression-hoi-quy-logistic/>

[4]GraphSAGE: Inductive Representation Learning on Large Graphs\_Link truy cập: <https://snap.stanford.edu/graphsage/>

[5]GraphSAGE\_Link truy cập:<https://paperswithcode.com/method/graphsage>

[6] Github tham khảo\_Link truy cập:<https://github.com/dsgiitr/graph_nets>

**Link drive các sản phẩm:** <https://drive.google.com/drive/folders/14hClI6R2y0lpvPT2TUhkFhDy2qJiXiHk?usp=sharing>