

MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding

Giảng viên hướng dẫn: TS. Đỗ Thị Thanh Hà

Nguyễn Mạnh Linh

Nguyễn Đức Thịnh

Tóm tắt nội dung—Một lượng lớn các đồ thị hay mạng trong thực tế vốn dĩ không đồng nhất, có nhiều loại nút và nhiều loại quan hệ. Biểu diễn đồ thị không đồng nhất là việc biểu diễn từ cấu trúc lớn và nhiều thông tin của đồ thị về biểu diễn nút trong không gian thấp chiều. Các mô hình đã tồn tại tường định nghĩa metapaths trong một đồ thị không đồng nhất để ghi lại các quan hệ và định hướng lựa chọn các nút lân cận. Tuy nhiên các mô hình này bỏ qua đặc trưng của từng nút mà tìm hiểu ngay lập tức các nút trên metapath hoặc chỉ xem xét một metapath. Để khắc phục ba giới hạn này, tác giả đề xuất một mô hình mới là *Metapath Aggregated Graph Neural Network* (MAGNN) để nâng cao hiệu năng cuối cùng. Đặc biệt, MAGNN sử dụng ba thành phần chính, biến đổi nội dung của nút thành các thuộc tính đóng gói của nút đầu vào, tổng hợp intra-metapath để kết hợp các nút ngữ nghĩa trung gian và tổng hợp inter-metapath để kết hợp thông tin từ nhiều metapaths. Các thí nghiệm được thực hiện trên ba bộ dữ liệu đồ thị không đồng nhất trong thực tế để phân loại nút, phân cụm nút và dự đoán liên kết chỉ ra rằng MAGNN đạt được kết quả dự đoán chính xác hơn so với các mô hình state-of-the-art hiện tại.

I. GIỚI THIỆU

Nhiều bộ dữ liệu thực tế được biểu diễn dưới dạng cấu trúc dữ liệu đồ thị, trong đó các đối tượng và quan hệ giữa chúng được biểu diễn bằng các nút và cạnh. Ví dụ như mạng xã hội [14, 29], hệ thống vật lý [2, 10], mạng giao thông [18, 34], mạng trích dẫn [1, 14, 16], hệ thống gợi ý [26, 35], đồ thị tri thức [3, 24],... Bản chất non-Euclidean của đồ thị khiến chúng khó có thể mô hình hóa bằng các mô hình học máy truyền thống. Với tập lân cận của mỗi nút, không hề có thứ tự hoặc giới hạn về kích thước. Tuy nhiên, hầu hết các mô hình thống kê ngày nay lại được xây dựng dựa trên giả định rằng các thông tin đầu vào có thứ tự và kích thước cố định trong không gian Euclid. Do đó, việc biểu diễn các nút bằng các vector thấp chiều trong không gian Euclid sẽ thuận tiện hơn nhiều cho việc sử dụng làm thông tin đầu vào của các mô hình học máy.

Hiện nay, đã có nhiều kỹ thuật biểu diễn đồ thị khác nhau được đề xuất để biểu diễn cấu trúc dữ liệu đồ thị. Chẳng hạn như mô hình LINE [25] tạo ra biểu diễn nút dựa vào độ tương đồng bậc nhất và bậc hai giữa các nút. Các phương pháp dựa trên bước ngẫu nhiên (Random-walk) bao gồm DeepWalk [21], node2vec [13] và TADW [32] sử dụng một chuỗi các nút được sinh ra bởi các bước ngẫu nhiên để đưa vào một mô hình skip-gram [19] để học cách biểu diễn nút. Với sự phát

triển nhanh chóng của deep learning, một số nghiên cứu đề xuất sử dụng mạng neuron đồ thị (Graph neural networks - GNNs), đây là mô hình học cách biểu diễn đồ thị bằng việc sử dụng các lớp neuron được thiết kế một cách đặc biệt. GNNs dựa trên phương pháp phổ (spectral-based), bao gồm ChebNet [8] và GCN [16] biểu diễn các toán tử tích chập đồ thị trong miền Fourier của một đồ thị đầy đủ. Các mô hình GNN dựa trên không gian (spatial-based), bao gồm GraphSAGE [14], GAT [28] và các biến thể khác [17, 34, 45], giải quyết các vấn đề xung quanh khả năng mở rộng và khả năng tổng quát hóa của các mô hình dựa trên phương pháp phổ bằng cách biểu diễn các phép toán tích chập đồ thị trực tiếp trên miền đồ thị. Ngoài ra, càng ngày càng có nhiều nghiên cứu khác nhau về đề tài này, cho thấy đây là một lĩnh vực đầy hứa hẹn.

Mặc dù GNNs đã đạt được những kết quả tốt nhất trong nhiều bài toán, hầu hết các mô hình dựa trên GNN giả định rằng đầu vào là đồ thị đồng nhất với chỉ một loại nút và một loại cạnh. Tuy nhiên, trên thực tế hầu hết đồ thị lại có thể bao gồm nhiều loại nút và loại cạnh tương ứng với các thuộc tính trong các không gian thuộc tính khác nhau. Ví dụ, một mạng đồng tác giả chứa ít nhất hai loại nút là các tác giả và bài báo. Các thuộc tính của tác giả có thể bao gồm nơi làm việc, trích dẫn và lĩnh vực nghiên cứu. Thuộc tính của bài báo bao gồm từ khóa, nơi xuất bản, năm phát hành,... Tác giả gọi loại đồ thị này là *mạng thông tin không đồng nhất* (HINs) hoặc *đồ thị không đồng nhất*. Sự không đồng nhất trong cả cấu trúc biểu diễn và nội dung của nút khiến GNN gặp khó khăn trong việc mã hóa thông tin vào không gian vector thấp chiều.

Hầu hết các phương pháp biểu diễn đồ thị không đồng nhất hiện có dựa trên ý tưởng về metapaths. Một *metapath* là một chuỗi có thứ tự của các loại nút và loại cạnh được xác định trên lược đồ mạng, mô tả mối quan hệ tổng hợp giữa các loại nút liên quan. Ví dụ một mạng với tác giả, bài báo và địa điểm, *Tác giả-Bài báo-Tác giả* (APA) và *Tác giả-Bài báo-Địa điểm-Bài báo-Tác giả* (APVPA) là các metapaths mô tả mối quan hệ khác nhau giữa các tác giả. Metapath APA tương ứng với hai đồng tác giả, trong khi APVPA tương ứng với hai tác giả có các bài báo trong cùng một tạp chí. Do đó, ta có thể xem metapath là độ tương đồng bậc cao giữa hai nút. Do các mô hình GNNs truyền thống xử lý tất cả các nút như nhau, vì vậy các mô hình này không thể mô hình hóa hết được toàn bộ cấu trúc phức tạp và lượng thông tin phong phú trong các đồ thị không đồng nhất.

Mặc dù các phương pháp biểu diễn dựa trên metapath này

hoạt động tốt hơn phương pháp biểu diễn mạng truyền thống trên các bài toán khác nhau, chẳng hạn như phân loại nút và dự đoán liên kết, chúng vẫn tồn tại ít nhất một trong những hạn chế sau. (1) Mô hình không sử dụng đặc trưng về nội dung nút nên nó khó có thể hiệu quả với các đồ thị không đồng nhất với các nút có nội dung nhiều đặc trưng (ví dụ metapath2vec [9], ESIM [22], HIN2vec [11], HERec [23]). (2) Mô hình loại bỏ tất cả các nút trung gian trên metapath, chỉ xem xét 2 nút đầu và cuối dẫn đến việc đánh mất thông tin của đồ thị (ví dụ HERec [23] and HAN [31]). (3) Mô hình dựa trên một metapath duy nhất để biểu diễn đồ thị không đồng nhất. Do đó, mô hình yêu cầu một quá trình chọn metapath thủ công và mất đi các đặc điểm của thông tin từ các metapaths khác dẫn đến hiệu năng không tối ưu (ví dụ metapath2vec [9]).

Để giải quyết các hạn chế này, tác giả đề xuất một mạng neuron tổng hợp metapath (*Metapath Aggregated Graph Neural Network - MAGNN*) mới cho việc biểu diễn đồ thị không đồng nhất. MAGNN giải quyết tất cả các vấn đề được mô tả ở trên bằng cách áp dụng chuyển đổi nội dung nút, tổng hợp intra-metapath và tổng hợp inter-metapath để tạo ra các biểu diễn nút. Cụ thể, MAGNN trước tiên áp dụng biến đổi tuyến tính theo loại cụ thể để chiếu các thuộc tính nút không đồng nhất với số chiều có thể không bằng nhau cho các loại nút khác nhau, vào trong cùng một không gian latent. Tiếp theo, MAGNN áp dụng tổng hợp intra-metapath với cơ chế chú ý [28] cho mọi metapath. Trong quá trình tổng hợp intra-metapath này, mỗi nút đích trích xuất và kết hợp thông tin từ các cấu hình metapath kết nối với nút lân cận dựa trên metapath của nó. Bằng cách này, MAGNN nắm bắt được thông tin về cấu trúc và ngữ nghĩa của các đồ thị không đồng nhất từ cả các nút lân cận và bối cảnh metapath giữa chúng. Sau khi tổng hợp intra-metapath, MAGNN tiếp tục tiến hành tổng hợp inter-metapath bằng cách sử dụng cơ chế chú ý để hợp nhất các latent vectors thu được từ nhiều metapaths vào các biểu diễn nút cuối cùng. Bằng cách tích hợp nhiều metapaths, mô hình của tác giả có thể học ngữ nghĩa toàn diện ẩn giấu trong đồ thị không đồng nhất.

Tóm lại, bài báo này có một số đóng góp chính:

- (1) Tác giả đề xuất một mạng neuron đồ thị mới dựa trên tổng hợp metapath để biểu diễn đồ thị không đồng nhất.
- (2) Thiết kế một số hàm mã hóa tiềm năng để trích xuất thông tin từ các cấu hình metapaths, bao gồm trường hợp dựa trên ý tưởng về phép xoay quan hệ trong không gian phức [24].
- (3) Tác giả tiến hành nhiều thí nghiệm trên các tập dữ liệu IMDB và DBLP để phân loại và phân cụm nút cũng như dùng tập Last.fm để đánh giá khả năng dự đoán liên kết và hiệu suất của mô hình. Thí nghiệm trên tất cả các tập dữ liệu này và các bài toán chỉ ra rằng MAGNN tạo ra các biểu diễn nút luôn tốt hơn so các mô hình tham chiếu tiên tiến nhất hiện tại (SOTA).

II. SƠ LƯỢC

Trong phần này, tác giả đưa ra các định nghĩa chuẩn của một số thuật ngữ quan trọng liên quan đến đồ thị không thuần nhất. Minh họa trong hình 1. Bên cạnh đó bảng 1 tóm tắt các

Bảng 1
KÍ HIỆU ĐƯỢC DÙNG TRONG BÁO CÁO

Kí hiệu	Định nghĩa
\mathbb{R}^n	Không gian vector Euclid n chiều
$a, \mathbf{a}, \mathbf{A}$	Số, vector, ma trận
\mathbf{A}^T	Ma trận/vector chuyển vị
\mathcal{V}	Tập đỉnh của đồ thị
\mathcal{E}	Tập cạnh của đồ thị
\mathcal{G}	Đồ thị $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
ν	Một nút $\nu \in \mathcal{V}$
P	Một metapath
$P_{\nu, u}$	Một cấu hình metapath kết nối nút ν và nút u
\mathcal{N}_{ν}	Tập lân cận của nút ν
\mathcal{N}_{ν}^P	Tập lân cận dựa trên metapath P của nút ν
x_{ν}	Đặc trưng (nội dung) thô của nút ν
h_{ν}	Trạng thái ẩn (nhúng) của nút ν
\mathbf{W}	Ma trận trọng số
α, β	Trọng số chú ý chuẩn hóa
$\sigma(\cdot)$	Hàm kích hoạt
\odot	Phép nhân phần tử
$ \cdot $	Lực lượng của tập hợp
\parallel	Nối vector

kí hiệu được sử dụng nhiều trong báo cáo để thuận tiện cho việc tra cứu nhanh.

Định nghĩa II.1. Đồ thị không đồng nhất. Một đồ thị không đồng nhất được định nghĩa là một đồ thị $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ với ánh xạ của loại nút $\phi : \mathcal{V} \rightarrow \mathcal{A}$ và ánh xạ của loại cạnh $\psi : \mathcal{E} \rightarrow \mathcal{R}$. \mathcal{A} và \mathcal{R} lần lượt là các tập loại nút và loại cạnh với $|\mathcal{A}| + |\mathcal{R}| > 2$.

Định nghĩa II.2. Metapath. Một metapath P được định nghĩa là một đường đi lặp thành từ $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$ (viết tắt là $A_1 A_2 \dots A_{l+1}$) mô tả một quan hệ tổng hợp $R = R_1 \circ R_2 \circ \dots \circ R_l$ giữa các loại nút A_1 và A_{l+1} , trong đó \circ là toán tử tổng hợp trên các quan hệ.

Định nghĩa II.3. Cấu hình metapath. Cho một metapath P của một đồ thị không đồng nhất, một cấu hình metapath p của P được định nghĩa là một dãy các nút trong đồ thị theo lược đồ được xác định bởi P .

Định nghĩa II.4. Lân cận dựa trên metapath. Cho một metapath P của một đồ thị không đồng nhất, các lân cận dựa trên metapath \mathcal{N}_{ν}^P của một nút ν được định nghĩa là tập hợp các nút liên kết với nút ν qua các cấu hình metapath của P . Một lân cận được kết nối bởi hai cấu hình metapath khác nhau được đánh giá như hai nút khác nhau trong \mathcal{N}_{ν}^P . Lưu ý rằng \mathcal{N}_{ν}^P bao gồm chính nút ν nếu P đối xứng.

Ví dụ, xem xét tập dữ liệu UATA trong hình 1, nghệ sĩ *Queen* là một lân cận dựa trên metapath của người dùng *Bob*. Hai nút này được kết nối thông qua cấu hình metapath *Bob-Beatles-Rock-Queen*. Hơn nữa, chúng ta có thể tham chiếu tới *Beatles* và *Rock* như là các nút trung gian trên cấu hình metapath này.

Định nghĩa II.5. Đồ thị dựa trên metapath. Cho một metapath P của một đồ thị không đồng nhất \mathcal{G} , đồ thị dựa trên metapath \mathcal{G}^P là một đồ thị được xây dựng bởi tất cả các cặp lân cận dựa trên metapath P trong đồ thị \mathcal{G} . Lưu ý rằng

\mathcal{G}^P là đồng nhất nếu P đối xứng.

Định nghĩa II.6. Biểu diễn đồ thị không đồng nhất. Cho một đồ thị không đồng nhất $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ với các ma trận thuộc tính nút $\mathbf{X}_{A_i} \in \mathbb{R}^{|\mathcal{V}_{A_i}| \times d_{A_i}}$ của các loại nút $A_i \in \mathcal{A}$, biểu diễn đồ thị không đồng nhất là việc học các biểu diễn nút d chiều $\mathbf{h}_v \in \mathbb{R}^d$ với mọi $v \in \mathcal{V}$ với $d \ll |\mathcal{V}|$ có thể ghi lại thông tin cấu trúc và ngữ nghĩa liên quan đến \mathcal{G} .

III. NGHIÊN CỨU LIÊN QUAN

Phần này sẽ tập trung tổng kết về các phương pháp học cách biểu diễn đồ thị liên quan tới mô hình mà ta đang xét đến trong phạm vi báo cáo này. Trong đó gồm hai tiểu mục: phần đầu sẽ tóm tắt các công trình nghiên cứu về GNN để biểu diễn đồ thị, phần tiếp theo sẽ giới thiệu các phương pháp biểu diễn đồ thị cho các đồ thị không đồng nhất.

A. Mạng neuron đồ thị

Mục tiêu của một GNN là học cách biểu diễn vector thấp chiều \mathbf{h}_v cho mọi nút v , để từ đó có thể sử dụng cho nhiều bài toán tiếp theo sau đó, ví dụ như phân loại nút, phân cụm nút và dự đoán liên kết. Mục tiêu này là hoàn toàn hợp lý vì mỗi nút có thể được xác định một cách tự nhiên bởi các thuộc tính đặc trưng và khu vực lân cận của nó. Dựa trên ý tưởng này và quá trình xử lý tín hiệu đồ thị, các mô hình GNNs dựa trên phương pháp phổ (spectral-based) được phát triển để thực hiện tích chập đồ thị trong miền Fourier của một đồ thị. ChebNet [8] sử dụng các đa thức Chebusev để lọc các tín hiệu đồ thị (đặc trưng nút) trong miền Fourier của đồ thị. Một mô hình khác thuộc loại này cũng thường xuyên được nhắc đến chính là GCN [16], mô hình này ràng buộc và đơn giản hóa các tham số của ChebNet để giảm bớt vấn đề gây ra bởi hiện tượng overfitting và cải thiện hiệu năng tổng thể của mô hình. Tuy nhiên, GNNs dựa trên phổ có khả năng mở rộng và tổng quát hóa kém vì chúng yêu cầu sử dụng toàn bộ đồ thị làm đầu vào cho tất cả các lớp, đồng thời các bộ lọc (filters) cần được học của mô hình này lại phụ thuộc vào cơ sở riêng của Laplacian của đồ thị, liên quan chặt chẽ đến cấu trúc đồ thị cụ thể.

GNNs dựa trên không gian (spatial-based) đã được đề xuất để khắc phục những hạn chế kể trên. Các mô hình GNNs kiểu này xác định các tích chập một cách trực tiếp trong miền đồ thị bằng cách tổng hợp thông tin từ các lân cận của mỗi nút, giống như các toán tử tích chập trong mạng tích chập xử lý dữ liệu hình ảnh. GraphSAGE [14], một framework GNN dựa trên spatial được tạo ra dựa trên ý tưởng chung về các hàm tổng hợp để tạo các biểu diễn nút hiệu quả. Các hàm tổng hợp sẽ lấy mẫu, trích xuất, và biến đổi một lân cận của nút mục tiêu và từ đó thúc đẩy việc huấn luyện song song và tổng quát hóa cho các nút hoặc phần đồ thị chưa được quan sát (gán nhãn). Rất nhiều biến thể của GNN dựa trên không gian được đề xuất dựa trên ý tưởng này. Dựa trên ý tưởng từ Transformer [27], GAT [28] kết hợp cơ chế chú ý vào hàm tổng hợp để giải thích cho mức độ ảnh hưởng tương đối của một nút đích tới các lân cận. Trong khi đó, mô hình GGNN [17] thì lại thêm một thành phần lặp có kiểm soát (GRU) [7] vào hàm tổng hợp

bằng cách xử lý thông tin vùng lân cận được tổng hợp làm đầu vào cho GRU của bước thời gian hiện tại. GaAN [34] kết hợp GRU với cơ chế chú ý nhiều đầu để thỏa mãn không thời gian đồ thị. STAR-GCN [35] dự dụng nhiều bộ mã hóa-giải mã GCN để tăng hiệu suất dự đoán xếp hạng.

Tất cả các GNN được đề cập ở trên đều được xây dựng cho các đồ thị đồng nhất hoặc được thiết kế cho một số đồ thị với cấu trúc đặc biệt như là trong các hệ thống gợi ý người dùng - sản phẩm. Do hầu hết các mô hình GNNs hiện tại đều chỉ có thể hoạt động được trên các nút có thuộc tính đặc trưng trong cùng không gian biểu diễn chung, các mô hình này không thể hoạt động tương tự với các đồ thị không đồng nhất với các đặc trưng nút nằm trên các không gian khác nhau.

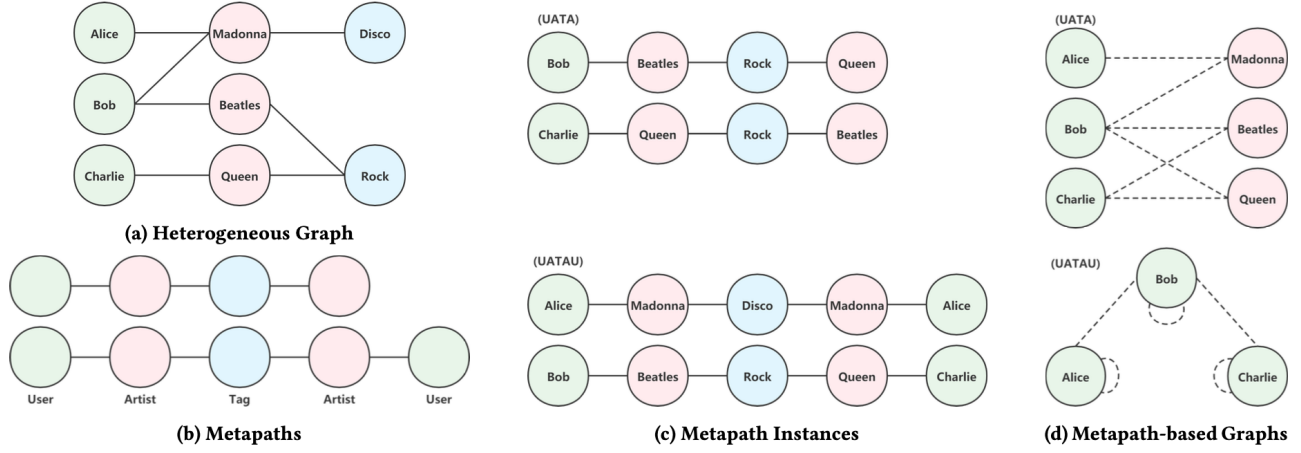
B. Biểu diễn đồ thị không đồng nhất

Biểu diễn đồ thị không đồng nhất nhằm mục đích chiếu các nút từ một đồ thị không đồng nhất vào một không gian vector thấp chiều. Bài toán đây thách thức này đã thu hút được sự quan tâm từ rất nhiều nghiên cứu. Ví dụ, metapath2vec [9] sinh các bước ngẫu nhiên được định hướng bởi một meta-path đơn, các bước ngẫu nhiên này sau đó được lấy làm đầu vào cho mô hình skip-gram [19] để sinh ra các biểu diễn nút. Với metapaths được định nghĩa trước, ESIM [22] sinh các biểu diễn nút bằng cách học từ các cấu hình metapath đã được lấy mẫu và gán nhãn negative và positive. HIN2vec [11] thực hiện nhiều tác vụ huấn luyện để học các biểu diễn của nút và metapaths của một đồ thị không đồng nhất. Cho một metapath, HERec [23] chuyển đổi một đồ thị không đồng nhất thành một đồ thị đồng nhất dựa trên các lân cận metapath-based và áp dụng mô hình DeepWalk để học biểu diễn nút của các loại mục tiêu. Giống như HERec, HAN [31] chuyển đổi một đồ thị không đồng nhất thành nhiều đồ thị đồng nhất dựa trên metapath theo cách tương tự nhưng sử dụng một kiến trúc mạng đồ thị có chú ý để tổng hợp thông tin từ các lân cận và thúc đẩy cơ chế chú ý để kết hợp nhiều metapaths. Một mô hình khác là PME [6] học cách biểu diễn nút bằng cách chiếu chúng vào các không gian quan hệ tương ứng và tối ưu hóa mức độ tương đồng giữa các nút được chiếu.

Tuy nhiên, tất cả các phương pháp biểu diễn đồ thị không đồng nhất được giới thiệu ở trên có những hạn chế là bỏ qua các đặc trưng về nội dung của nút, loại bỏ tất cả các nút trung gian dọc theo metapath hoặc chỉ sử dụng một metapath duy nhất. Mặc dù chúng có thể được cải thiện dựa trên hiệu năng của các phương pháp biểu diễn đồ thị không đồng nhất cho một số bộ dữ liệu đồ thị không đồng nhất, ta vẫn có thể làm tốt hơn bằng cách khai thác toàn diện hơn các thông tin trong đồ thị không đồng nhất.

IV. PHƯƠNG PHÁP

Trong phần này, tác giả mô tả một mạng neuron đồ thị tổng hợp metapath mới (MAGNN) để biểu diễn đồ thị không đồng nhất. MAGNN được xây dựng bởi 03 thành phần chính: biến đổi nội dung nút, tổng hợp intra-metapath và tổng hợp inter-metapath. Hình 2 minh họa việc tạo biểu diễn của một nút. Các quá trình lan truyền tiến được giới thiệu trong thuật toán 1.



Hình 1. Minh họa các thuật ngữ được định nghĩa trong Phần 2. (a) Một ví dụ về đồ thị không đồng nhất với ba loại nút (người dùng, nghệ sĩ, thể). (b) metapath Người dùng-Nghệ sĩ-Thể-Nghệ sĩ (UATA) và metapath Người dùng-Nghệ sĩ-Thể-Nghệ sĩ-Người dùng (UATAU). (c) Ví dụ các cấu hình metapath của UATA, UATAU. (d) Đồ thị dựa trên metapath cho UATA và UATAU.

A. Biến đổi nội dung nút

Với một đồ thị không đồng nhất liên kết với các thuộc tính nút, các loại nút khác nhau có thể có số chiều của các vector đặc trưng không bằng nhau. Kể cả chúng có số chiều bằng nhau thì chúng cũng nằm trên các không gian đặc trưng khác nhau. Ví dụ các bag-of-words vectors n_1 chiều của đoạn văn bản và các vectors biểu đồ cường độ n_2 chiều của hình ảnh không thể trực tiếp hoạt động cùng nhau kể cả $n_1 = n_2$. Các vectors đặc trưng với các chiều khác nhau là một khó khăn khi ta xử lý chúng trong một framework thống nhất. Do đó, ta cần chiếu các loại thuộc tính đặc trưng khác nhau của nút vào cùng một không gian vector latent trước.

Trước khi đưa các vectors nút vào MAGNN, ta áp dụng phép biến đổi tuyến tính cho từng loại nút bằng cách chiếu các vector đặc trưng vào cùng một không gian latent. Với một nút $\nu \in \mathcal{V}_A$ của loại $A \in \mathcal{A}$, ta có

$$\mathbf{h}'_{\nu} = \mathbf{W}_A \cdot \mathbf{x}_{\nu}^A \quad (1)$$

trong đó $\mathbf{x}_{\nu} \in \mathbb{R}^{d_A}$ là vector đặc trưng gốc và $\mathbf{h}'_{\nu} \in \mathbb{R}^{d'}$ là vector latent hình chiếu của nút ν . $\mathbf{W}_A \in \mathbb{R}^{d' \times d_A}$ là ma trận trọng số của các nút loại A.

Biến đổi nội dung nút giúp giải quyết tính không đồng nhất của một đồ thị gây ra bởi sự khác nhau trong các đặc trưng nội dung của nút. Sau khi áp dụng các biến đổi này, hình chiếu của tất cả các đặc trưng của nút đều có cùng chiều, tạo điều kiện thuận lợi cho quá trình tổng hợp - thành phần tiếp theo của mô hình MAGNN.

B. Tổng hợp intra-metapath

Cho một metapath P , lớp tổng hợp intra-metapath học thông tin cấu trúc và ngữ nghĩa được biểu diễn trong (i) các nút mục tiêu, (ii) các lân cận dựa trên metapath và (iii) ngữ cảnh ở giữa, bằng cách mã hóa cấu hình metapath của P . Gọi $P(\nu, u)$ là một cấu hình metapath kết nối nút mục tiêu ν và lân cận dựa trên metapath $u \in \mathcal{N}_{\nu}^P$, tác giả định nghĩa thêm nút trung

gian của $P(\nu, u)$ như sau $\{m^{P(\nu, u)}\} = P(\nu, u) \setminus \{\nu, u\}$. Tổng hợp intra-metapath sử dụng một bộ mã hóa cấu hình metapath để biến đổi tất cả các đặc trưng nút dọc theo một cấu hình metapath thành một vector duy nhất,

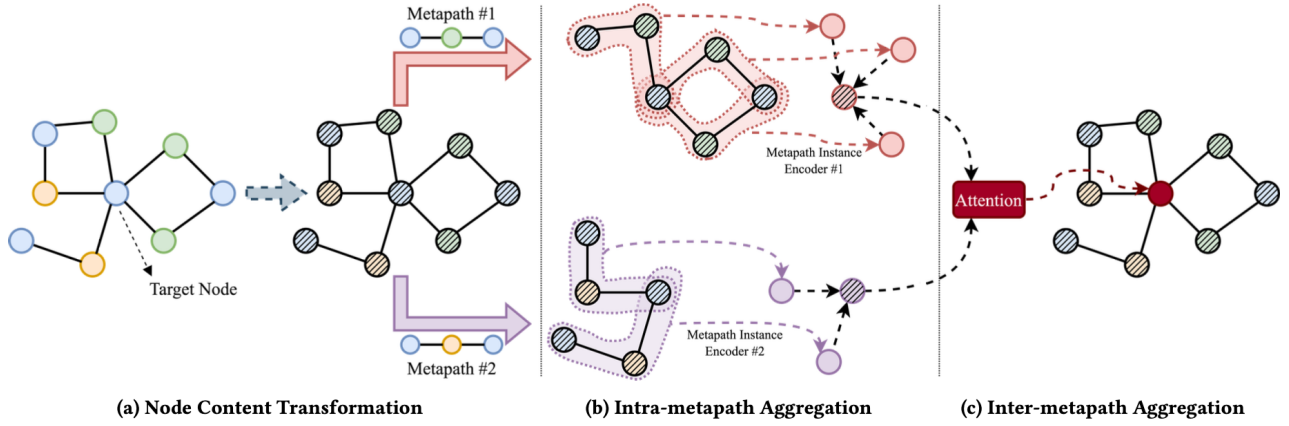
$$\mathbf{h}_{P(\nu, u)} = f_{\theta}(P(\nu, u)) = f_{\theta}(\mathbf{h}'_{\nu}, \mathbf{h}'_u, \{\mathbf{h}'_t, \forall t \in \{m^{P(\nu, u)}\}\}) \quad (2)$$

trong đó, $\mathbf{h}_{P(\nu, u)} \in \mathbb{R}^{d'}$ có số chiều là d' . Để đơn giản, ta dùng $P(\nu, u)$ để biểu diễn một cấu hình đơn, mặc dù có thể có nhiều cấu hình kết nối 2 nút. Phần sau sẽ giới thiệu một vài lựa chọn của bộ mã hóa cấu hình metapath tốt.

Sau khi mã hóa các cấu hình metapath thành dạng vector, ta sử dụng một lớp chú ý đồ thị [28] để tính tổng có nhân trọng số (weighted sum) các cấu hình metapath của P liên quan đến nút đích ν . Ý tưởng chính là các cấu hình metapath khác nhau sẽ đóng góp vào biểu diễn nút mục tiêu với mức độ (trọng số) khác nhau. Chúng ta có thể mô hình hóa điều này bằng cách học một trọng số chuẩn hóa thể hiện mức độ quan trọng $\alpha_{\nu u}^P$ tương ứng với mỗi cấu hình metapath và sau đó tính tổng có nhân trọng số của tất cả các cấu hình:

$$\begin{aligned} e_{\nu u}^P &= \text{LeakyReLU}(a_P^T \cdot [\mathbf{h}'_{\nu} \parallel \mathbf{h}_{P(\nu, u)}]), \\ \alpha_{\nu u}^P &= \frac{\exp(e_{\nu u}^P)}{\sum_{s \in \mathcal{N}_{\nu}^P} \exp(e_{\nu s}^P)}, \\ \mathbf{h}_{\nu}^P &= \sigma \left(\sum_{u \in \mathcal{N}_{\nu}^P} \alpha_{\nu u}^P \cdot \mathbf{h}_{P(\nu, u)} \right). \end{aligned} \quad (3)$$

Trong đó, $a_P \in \mathbb{R}^{2d'}$ là vector chú ý được tham số hóa cho metapath P và \parallel kí hiệu cho toán tử nối vector. $e_{\nu u}^P$ chỉ độ quan trọng của cấu hình metapath $P(\nu, u)$ đến nút ν , nút sau đó được chuẩn hóa theo các lựa chọn $u \in \mathcal{N}_{\nu}^P$ sử dụng hàm softmax. Do trọng số chuẩn hóa $\alpha_{\nu u}^P$ được lấy cho tất cả $u \in \mathcal{N}_{\nu}^P$, chúng được sử dụng để tính toán một tổ hợp có trọng số của các biểu diễn của các cấu hình metapath cho nút ν . Cuối cùng, kết quả đầu ra được đưa qua hàm kích hoạt $\sigma(\cdot)$.



Hình 2. Kiến trúc tổng thể của MAGNN

Cơ chế chú ý cũng có thể được mở rộng thành nhiều nhánh, điều này giúp ổn định quá trình học và giảm thiểu phương sai lớn từ tính không đồng nhất của đồ thị. Nghĩa là, chúng ta thực hiện K cơ chế chú ý độc lập và sau đó tổng hợp kết quả đầu ra của chúng lại, kết quả thu được trong biểu thức sau:

$$\mathbf{h}_\nu^P = \parallel_{k=1}^K \sigma \left(\sum_{u \in \mathcal{N}_\nu^P} [\alpha_{\nu u}^P]_k \cdot \mathbf{h}_{P(\nu, u)} \right) \quad (4)$$

trong đó $[\alpha_{\nu u}^P]_k$ là trọng số chuẩn hóa của cấu hình metapath $P(\nu, u)$ đến nút ν tại nhánh chú ý thứ k .

Tóm lại, với các vector đặc trưng $\mathbf{h}'_u \in \mathbb{R}^{d'} \forall u \in \mathcal{V}$ và tập các metapaths $\mathcal{P}_A = P_1, P_2, \dots, P_M$ bắt đầu hoặc kết thúc với loại nút $A \in \mathcal{A}$, tổng hợp intra-metapath của MAGNN sinh M biểu diễn metapath-specific vector của nút đích $\nu \in \mathcal{V}_A$, kí hiệu là $\{\mathbf{h}_\nu^{P_1}, \mathbf{h}_\nu^{P_2}, \dots, \mathbf{h}_\nu^{P_M}\}$. Mỗi $\mathbf{h}_\nu^{P_1} \in \mathbb{R}^{d'}$ (giả sử $K = 1$) có thể hiểu là tổng hợp của các cấu hình P_i – metapath của nút ν , thể hiện một khía cạnh của thông tin chứa trong nút ν .

C. Tổng hợp inter-metapath

Sau khi tổng hợp dữ liệu nút và cạnh với mỗi metapath, chúng ta cần kết hợp thông tin của tất cả các metapath sử dụng một lớp tổng hợp inter-metapath. Bây giờ với một loại nút A , ta có $|\mathcal{V}_A|$ tập các latent vectors: $\{\mathbf{h}_\nu^{P_1}, \mathbf{h}_\nu^{P_2}, \dots, \mathbf{h}_\nu^{P_M}\}$ với $\nu \in \mathcal{V}_A$, với M là số metapaths cho loại A . Một cách tiếp cận tổng hợp inter-metapath trực tiếp là lấy trung bình theo từng thành phần (element-wise) của các vectors nút này. Ta mở rộng cách tiếp cận này bằng cách khai thác cơ chế chú ý để gán các trọng số khác nhau cho các metapaths khác nhau. Phép toán này là hoàn toàn hợp lí vì các metapaths có đóng góp không giống nhau trong một đồ thị không đồng nhất.

Đầu tiên, ta cộng mỗi metapath $P_i \in \mathcal{P}_A$ bằng trung bình các vectors nút metapath-specific đã được biến đổi cho tất cả các nút $\nu \in \mathcal{V}_A$,

$$\mathbf{s}_{P_i} = \frac{1}{|\mathcal{V}_A|} \sum_{\nu \in \mathcal{V}_A} \tanh(\mathbf{M}_A \cdot \mathbf{h}_\nu^{P_i} + \mathbf{b}_A) \quad (5)$$

trong đó, $\mathbf{M}_A \in \mathbb{R}^{d_m \times d'}$ và $\mathbf{b}_A \in \mathbb{R}^{d_m}$ là các tham số học.

Sau đó ta sử dụng cơ chế chú ý để hợp nhất các vectors nút metapath-specific của ν như sau:

$$\begin{aligned} e_{P_i} &= \mathbf{q}_A^T \cdot \mathbf{s}_{P_i}, \\ \beta_{P_i} &= \frac{\exp(e_{P_i})}{\sum_{P \in \mathcal{P}_A} \exp(e_P)}, \\ \mathbf{h}_\nu^{P_A} &= \sum_{P \in \mathcal{P}_A} \beta_P \cdot \mathbf{h}_\nu^P \end{aligned} \quad (6)$$

trong đó, $\mathbf{q}_A \in \mathbb{R}^{d_m}$ là vector chú ý tham số hóa cho loại nút A . β_{P_i} có thể hiểu là mức độ đóng góp tương đối của metapath P_i cho các nút loại A . Do β_{P_i} được tính toán cho mỗi $P_i \in \mathcal{P}_A$, ta có thể tính tổng có trọng số tất cả các vectors nút metapath-specific của ν .

Cuối cùng, MAGNN sử dụng một biến đổi tuyến tính bổ sung với một hàm phi tuyến để chiếu các biểu diễn nút vào không gian vector với số chiều đầu ra mong muốn:

$$\mathbf{h}_\nu = \sigma(\mathbf{W}_0 \cdot \mathbf{h}_\nu^{P_A}) \quad (7)$$

trong đó, $\sigma(\cdot)$ là một hàm kích hoạt và $\mathbf{W}_0 \in \mathbb{R}^{d_o \times d'}$ là một ma trận trọng số. Phép chiếu này là một tác vụ cụ thể. Nó có thể giải thích dưới dạng là một phép phân loại tuyến tính cho việc phân loại nút hoặc được coi là hình chiếu vào không gian với các độ đo mức độ tương đồng của các nút để từ đó dự đoán liên kết.

D. Biểu diễn (mã hóa) cấu hình metapath

Để mã hóa cấu hình metapath trong phần IV.B, ta xem xét ba hàm mã hóa khả dĩ sau:

- Mã hóa trung bình. Hàm này tính toán trung bình theo từng thành phần (element-wise) của nút dọc theo cấu hình metapath $P(v, u)$:

$$\mathbf{h}_{P(v, u)} = \text{MEAN}(\{\mathbf{h}'_t, \forall t \in P(v, u)\}). \quad (8)$$

- Mã hóa tuyến tính. Hàm này là một phiên bản mở rộng của mã hóa trung bình, trong đó việc mã hóa được thực hiện dựa trên các biến đổi tuyến tính:

$$\mathbf{h}_{P(v, u)} = \mathbf{W}_P \cdot \text{MEAN}(\{\mathbf{h}'_t, \forall t \in P(v, u)\}). \quad (9)$$

- Mã hóa dựa trên phép quay có quan hệ. Ta xem xét mã hóa một cấu hình metapath dựa trên phép quay có quan hệ trong không gian phức, một phương pháp được đề xuất bởi RotatE [24] để biểu diễn đồ thị tri thức. Hàm mã hóa trung bình và tuyến tính được giới thiệu ở trên coi các cấu hình metapath như một tập hợp, và vì thế chúng bỏ qua thông tin được biểu diễn trong kiến trúc chuỗi của metapath. Trong khi đó, phép quay có quan hệ lại cho phép ta mô hình hóa kiểu tri thức đó. Biết $P(v, u) = (t_0, t_1, \dots, t_n)$ with $t_0 = u$ và $t_n = v$, let R_i là mối quan hệ giữa nút t_{i-1} và nút t_i , gọi \mathbf{r}_i là vector mối quan hệ của R_i , khi đó mã hóa dựa trên phép quay có quan hệ được thể hiện bởi công thức.

$$\begin{aligned} \mathbf{o}_0 &= \mathbf{h}'_{t_0} = \mathbf{h}'_u, \\ \mathbf{o}_i &= \mathbf{h}'_{t_i} + \mathbf{o}_{i-1} \odot \mathbf{r}_i, \\ \mathbf{h}_{P(v,u)} &= \frac{\mathbf{o}_n}{n+1}, \end{aligned} \quad (10)$$

trong đó \mathbf{h}'_{t_i} và \mathbf{r}_i đều là các vector phức, \odot là tích vô hướng. Ta có thể dễ dàng phân tích một vector thực có số chiều là d' thành một vector phức có số chiều là $d'/2$ bằng cách coi nửa đầu tiên của vector là phần thực và nửa sau là phần ảo.

E. Huấn luyện mô hình

Sau khi thực hiện các biến đổi thành phần đã giới thiệu ở các phần trước, ta sẽ thu được biểu diễn cuối cùng của các nút để phục vụ cho các bài toán khác tiếp theo. Tùy theo đặc điểm khác nhau của các bài toán và việc nhân của các nút có sẵn hay không mà chúng ta có thể huấn luyện mô hình MAGNN theo 2 mẫu hình (paradigm) chính là học bán giám sát (semi-supervised learning) và học không giám sát (unsupervised learning).

Đối với trường hợp học bán giám sát, với thông tin có được từ một phần nhỏ các nút được gán nhãn, ta có thể học các tham số của mô hình để tối thiểu hóa hàm cross-entropy bằng phương pháp lan truyền ngược (backpropagation) hoặc gradient descent. Và từ đó học được cách biểu diễn nút cho đồ thị không đồng nhất sao cho bảo toàn được nhiều thông tin quan trọng nhất có thể. Hàm tổn thất cross-entropy cho trường hợp học bán giám sát được cho bởi công thức sau:

$$\mathcal{L} = - \sum_{v \in \mathcal{V}_L} \sum_{c=1}^C \mathbf{y}_v[c] \cdot \log \mathbf{h}_v[c] \quad (11)$$

trong đó \mathcal{V}_L là tập hợp các nút có nhãn, C là số lượng phân lớp, \mathbf{y}_v là vector one-hot thể hiện nhãn của nút v , và \mathbf{h}_v là vector xác suất dự báo của nút v .

Đối với trường hợp học không giám sát thì ta không có bất kỳ nhãn của nút nào cả, khi đó ta có thể học các tham số của mô hình bằng cách tối thiểu hóa hàm tổn thất sau đây [20]:

$$\mathcal{L} = - \sum_{(u,v) \in \Omega} \log \sigma(\mathbf{h}_u^\top \cdot \mathbf{h}_v) - \sum_{(u',v') \in \Omega^-} \log \sigma(-\mathbf{h}_{u'}^\top \cdot \mathbf{h}_{v'}), \quad (12)$$

trong đó $\sigma(\cdot)$ là hàm sigmoid, Ω là tập hợp các cặp nút quan sát được (positive), Ω^- là tập hợp các cặp nút âm tính được lấy mẫu từ tất cả các cặp nút không quan sát được (phần bù của Ω).

Algorithm 1: MAGNN forward propagation.

Input: The heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

node types $\mathcal{A} = \{A_1, A_2, \dots, A_{|\mathcal{A}|}\}$,

metapaths $\mathcal{P} = \{P_1, P_2, \dots, P_{|\mathcal{P}|}\}$,

node features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$,

the number of attention heads K ,

the number of layers L

Output: The node embeddings $\{\mathbf{z}_v, \forall v \in \mathcal{V}\}$

```

1 for node type  $A \in \mathcal{A}$  do
2   | Node content transformation  $\mathbf{h}_v^0 \leftarrow \mathbf{W}_A \cdot \mathbf{x}_v, \forall v \in \mathcal{V}_A$ ;
3 end
4 for  $l = 1 \dots L$  do
5   for node type  $A \in \mathcal{A}$  do
6     for metapath  $P \in \mathcal{P}_A$  do
7       for  $v \in \mathcal{V}_A$  do
8         Calculate  $\mathbf{h}_{P(v,u)}^l$  for all  $u \in \mathcal{N}_v^P$  using the
9         metapath instance encoder function;
10        Combine extracted metapath instances
11         $[\mathbf{h}_v^P]^l \leftarrow \big\| \sigma \left( \sum_{u \in \mathcal{N}_v^P} [\alpha_{vu}^P] \cdot \mathbf{h}_{P(v,u)}^l \right)$ ;
12      end
13    end
14    Calculate the weight  $\beta_P$  for each metapath  $P \in \mathcal{P}_A$ ;
15    Fuse the embeddings from different metapaths
16     $[\mathbf{h}_v^{\mathcal{P}_A}]^l \leftarrow \sum_{P \in \mathcal{P}_A} \beta_P \cdot [\mathbf{h}_v^P]^l, \forall v \in \mathcal{V}_A$ ;
17  end
18  Layer output projection
19   $\mathbf{h}_v^l = \sigma \left( \mathbf{W}_o^l \cdot [\mathbf{h}_v^{\mathcal{P}_A}]^l \right), \forall v \in \mathcal{V}_A, \forall A \in \mathcal{A}$ ;
20 end
21  $\mathbf{z}_v \leftarrow \mathbf{h}_v^L, \forall v \in \mathcal{V}$ ;

```

V. THỰC NGHIỆM

Trong phần này, tác giả trình bày các thực nghiệm để chứng minh tính hiệu quả của MAGNN đối với việc biểu diễn đồ thị không đồng nhất. Các thực nghiệm nhằm giải quyết các câu hỏi nghiên cứu sau:

- RQ1. MAGNN có hiệu quả như thế nào trong việc phân loại các nút?
- RQ2. MAGNN có hiệu quả như thế nào trong việc phân cụm các nút?
- RQ3. MAGNN có hiệu quả như thế nào trong việc dự đoán các liên kết hợp lý giữa các cặp nút?
- RQ4. Ảnh hưởng riêng biệt của 03 thành phần chính của MAGNN đã được mô tả trong các phần trước đó là gì?
- RQ5. Làm cách nào để ta có thể xác định được tính đại diện của những phương pháp biểu diễn đồ thị khác nhau?

A. Tập dữ liệu

Chúng tôi lựa chọn 03 tập dữ liệu đồ thị không đồng nhất phổ biến nhất hiện nay từ các lĩnh vực khác nhau để đánh giá hiệu năng của MAGNN khi so sánh với các phương pháp tham chiếu là các phương pháp tối ưu nhất hiện nay (baselines). Cụ thể, tập dữ liệu IMDB và DBLP được sử dụng trong thực

nghiệm liên quan đến việc phân loại nút và phân cụm nút. Tập dữ liệu Last.fm được sử dụng cho thực nghiệm về khả năng dự báo mối quan hệ. Những giá trị thống kê cơ bản của 03 tập dữ liệu được tóm tắt trong Bảng 2, và lược đồ mạng được thể hiện trong Hình 3. Chúng tôi sử dụng vector one-hot cho các nút không có thuộc tính như là các thuộc tính đầu vào giả (dummy) của chúng.

- **IMDb**¹ là cơ sở dữ liệu trực tuyến về các bộ phim và chương trình truyền hình, bao gồm các thông tin như dàn diễn viên, đội ngũ sản xuất và tóm tắt cốt truyện. Chúng tôi sử dụng một tập mẫu được lấy ra từ IMDb, thông qua quá trình tiền xử lý dữ liệu thu được 4278 bộ phim, 2081 đạo diễn và 5257 diễn viên. Phim được gán nhãn là một trong ba loại (Hành động, Hài kịch và Chính kịch) dựa trên thông tin thể loại của chúng. Mỗi bộ phim cũng được mô tả bằng một bag-of-words đại diện cho các từ khóa cốt truyện của chúng. Đối với các mô hình học bán giám sát, các nút phim được chia thành các tập huấn luyện, xác thực và kiểm tra với kích thước lần lượt là 400 (9,35%), 400 (9,35%) và 3478(81,30%) nút.
- **DBLP**² là một trang web tổng hợp danh mục tài liệu về khoa học máy tính. Chúng tôi sử dụng một tập mẫu được lấy ra từ DBLP [12, 15], sau khi tiền xử lý dữ liệu thu được thông tin về 4057 tác giả, 14328 bài báo, 7723 thuật ngữ và 20 nơi xuất bản. Các tác giả được chia thành bốn lĩnh vực nghiên cứu (Cơ sở dữ liệu, Khai thác dữ liệu, Trí tuệ nhân tạo và Truy xuất thông tin). Mỗi tác giả được mô tả bằng một bag-of-words đại diện cho các từ khóa trong bài báo của họ. Đối với các mô hình học bán giám sát, các nút tác giả được chia thành các tập huấn luyện, xác thực và kiểm tra với kích thước lần lượt là 400(9,86%) 400 (9,86%) và 3257 (80,28%) nút.
- **Last.fm**³ là trang web âm nhạc theo dõi thông tin hành vi nghe nhạc của người dùng từ nhiều nguồn khác nhau. Chúng tôi sử dụng bộ dữ liệu do HetRec 2011 phát hành [4], sau khi tiền xử lý dữ liệu thu được thông tin về 1892 người dùng, 17632 nghệ sĩ và 1088 thể nghệ sĩ. Tập dữ liệu này được sử dụng cho tác vụ dự đoán liên kết giữa các nút, trong đó tập dữ liệu không chứa bất cứ thông tin nào liên quan đến nhãn hay các đặc điểm đặc trưng của đối tượng. Đối với các mô hình học bán giám sát, các cặp người dùng - nghệ sĩ được chia thành các tập huấn luyện, xác thực và kiểm tra với kích thước lần lượt là 64984(70%), 9283(10%) và 18567(20%) cặp.

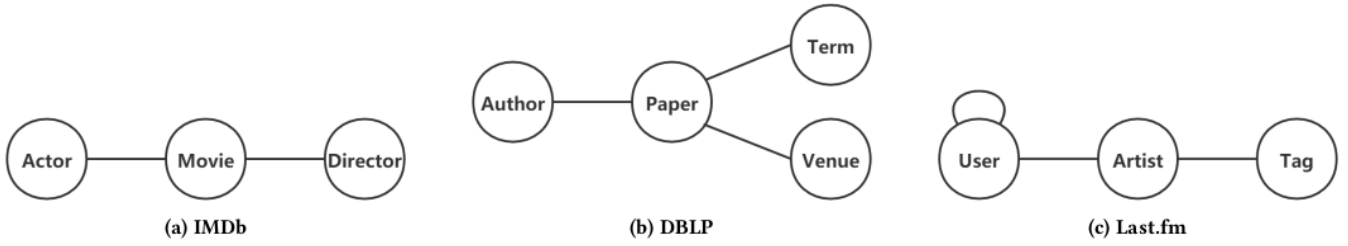
B. Các mô hình tham chiếu

Chúng tôi so sánh MAGNN với nhiều loại mô hình biểu diễn đồ thị khác nhau, bao gồm các mô hình biểu diễn đồ thị đồng nhất truyền thống (trái ngược với GNN), mô hình biểu diễn đồ thị không đồng nhất truyền thống, GNN cho đồ thị đồng nhất và GNN cho đồ thị không đồng nhất. Ta gọi chúng lần lượt là mô hình đồng nhất truyền thống, mô hình không đồng nhất truyền thống, GNN đồng nhất và GNN không đồng nhất. Danh sách các mô hình tham chiếu cơ sở được thể hiện dưới đây.

Bảng II
THỐNG KÊ MÔ TẢ CÁC TẬP DỮ LIỆU

Dữ liệu	Nút	Cạnh	Metapath
IMDb	# phim (M): 4278 # đạo diễn (D): 2081 # diễn viên (A): 5257	# M-D: 4278 # M-A: 12828	MDM MAM DMD DMAMD AMA AMDMA
DBLP	# tác giả (A): 4057 # bài báo (P): 14328 # thuật ngữ (T): 7723 # nơi xuất bản (V): 20	# A-P: 19645 # P-T: 85810 # P-V: 14328	APA APTPA APVPA
Last.fm	# người dùng (U): 1892 # nghệ sĩ (A): 17632 # thể (T): 1088	# U-U: 12,717 # U-A: 92,834 # A-T: 23,253	UU UUA UATAU AUA AUUA ATA

- **LINE** [25] là một mô hình đồng nhất truyền thống khai thác mức độ tương đồng bậc nhất và bậc hai giữa các nút. Chúng tôi áp dụng mô hình này cho các đồ thị không đồng nhất bằng cách bỏ qua tính không đồng nhất của cấu trúc đồ thị và loại bỏ tất cả các thuộc tính liên quan đến nội dung nút. Trong các thử nghiệm của tác giả, chúng tôi sử dụng biến thể LINE (sử dụng mức độ tương đồng bậc hai).
- **node2vec** [13] là một mô hình đồng nhất truyền thống và có thể coi là phiên bản tổng quát của DeepWalk [21]. Chúng tôi áp dụng mô hình này cho các đồ thị không đồng nhất theo cách tương tự như LINE.
- **ESim** [22] là một mô hình không đồng nhất truyền thống học cách biểu diễn nút từ các cấu hình metapath đã được lấy mẫu. ESim yêu cầu xác định trước các trọng số cho mỗi metapath. Ở đây, chúng tôi chỉ định các trọng số bằng nhau cho tất cả các metapath vì việc tìm kiếm các trọng số tối ưu của các metapath là rất khó và không mang lại mức tăng hiệu suất đáng kể so với việc đặt các trọng số mặc định bằng nhau theo các thử nghiệm của nhóm tác giả.
- **metapath2vec** [9] là một mô hình không đồng nhất truyền thống tạo ra các biểu diễn nút bằng cách cung cấp các random walks được định hướng bởi metapath cho một mô hình skip-gram. Mô hình này dựa trên một metapath do người dùng chỉ định, vì vậy chúng tôi thử nghiệm trên tất cả các metapath riêng biệt và tổng kết metapath có kết quả tốt nhất. Chúng tôi sử dụng biến thể metapath2vec++ trong các thử nghiệm của mình.
- **HERec** [23] là một mô hình không đồng nhất truyền thống học cách biểu diễn nút bằng cách áp dụng DeepWalk cho các đồ thị đồng nhất dựa trên metapath được chuyển đổi từ đồ thị không đồng nhất ban đầu. Mô hình này đi kèm với một thuật toán kết hợp biểu diễn được thiết kế để dự đoán xếp hạng, có thể được điều chỉnh để dự đoán liên kết. Để phân loại/phân cụm nút, chúng tôi chọn và báo cáo kết quả cho metapath có hiệu suất tốt nhất.



Hình 3. Các lược đồ mạng của ba bộ dữ liệu đồ thị không đồng nhất được sử dụng trong bài viết này.

- **GCN** [16] là một mô hình GNN đồng nhất. Mô hình này thực hiện các phép toán tích chập trong miền Fourier của đồ thị. Ở đây, chúng tôi kiểm tra hiệu suất của GCN trên các đồ thị đồng nhất dựa trên metapath và báo cáo kết quả cho metapath tốt nhất.
- **GAT** [28] là một GNN đồng nhất. Mô hình này thực hiện các thao tác tích chập trong miền không gian đồ thị với cơ chế kết hợp có chú ý. Tương tự, ở đây chúng tôi kiểm tra GAT trên các đồ thị đồng nhất dựa trên metapath và báo cáo kết quả cho metapath tốt nhất.
- **GATNE** [5] là một GNN không đồng nhất. Mô hình này tạo ra biểu diễn của nút từ biểu diễn cơ sở và biểu diễn cạnh, tập trung vào nhiệm vụ dự đoán liên kết. Ở đây chúng tôi báo cáo kết quả từ biến thể GATNE có hiệu quả tốt nhất.
- **HAN** [31] là một GNN không đồng nhất. Mô hình này học cách biểu diễn nút với metapath cụ thể từ các đồ thị đồng nhất khác nhau dựa trên metapath và tận dụng cơ chế chú ý để kết hợp chúng thành một biểu diễn vector cho mỗi nút.

Đối với các mô hình truyền thống, bao gồm LINE, node2vec, ESIm, metapath2vec và HERec, tác giả thiết lập window size là 5, walk length là 100, số bước đi trên mỗi nút là 40 và số lượng mẫu âm tính là 5 (nếu có). Đối với các mô hình GNN (bao gồm GCN, GAT, HAN và MAGNN), tác giả đặt tỷ lệ dropout là 0,5; tác giả sử dụng tập dữ liệu huấn luyện, xác thực và kiểm tra có kích thước bằng nhau; tác giả sử dụng phương pháp Adam cho các nhiệm vụ tối ưu hóa với learning rate được thiết lập là 0,005 và tham số phân rã (L2) được đặt là 0,001; tác giả huấn luyện mô hình GNN trong 100 epochs và cho phép dừng sớm với ngưỡng patience là 30. Để phân loại nút và phân cụm nút, GNN được huấn luyện theo kiểu bán giám sát với một phần nhỏ các nút được gắn nhãn. Đối với GAT, HAN và MAGNN, tác giả thiết lập số lượng head chú ý là 8. Đối với HAN và MAGNN, tác giả thiết lập số chiều (thứ nguyên) của vector chú ý trong tập hợp giữa các metapath là 128. Để việc so sánh được công bằng, tác giả đặt số chiều (thứ nguyên) biểu diễn của tất cả các mô hình được đề cập ở trên đến 64.

C. Phân loại nút (RQ1)

tác giả tiến hành thử nghiệm trên tập dữ liệu IMDb và DBLP để so sánh hiệu quả của các mô hình khác nhau đối

với tác vụ phân loại nút. tác giả sử dụng thông tin biểu diễn của các nút được gắn nhãn (phim trong IMDb và tác giả trong DBLP) được tạo ra bởi mỗi mô hình làm đầu vào cho thuật toán phân loại SVM với các tỷ lệ tập huấn luyện khác nhau. Để việc so sánh được công bằng, chỉ các nút trong tập kiểm tra (testing set) được đưa vào mô hình SVM, bởi vì các mô hình bán giám sát đã được học thông tin từ các nút trong tập huấn luyện (training set) và xác thực (validation set), như được thể hiện trong Phương trình (11). Do đó, tỉ lệ tập huấn luyện và kiểm tra của mô hình SVM ở đây chỉ liên quan đến tập dữ liệu kiểm tra (nghĩa là 3478 nút cho IMDb và 3257 nút cho DBLP). Việc phân chia tập train và test cho mô hình SVM cũng được thực hiện tương tự nhau giữa các mô hình biểu diễn đồ thị. Các chiến lược tương tự cũng được áp dụng cho các thử nghiệm liên quan đến bài toán phân cụm nút và dự đoán liên kết. tác giả tổng kết giá trị Macro-F1 và Micro-F1 trung bình của 10 lần chạy của từng mô hình biểu diễn trong Bảng 3.

Như được thể hiện trong bảng, MAGNN luôn hiệu quả hơn so với các mô hình tham chiếu khác trên các tỷ lệ tập dữ liệu huấn luyện khác nhau. Trên tập dữ liệu IMDb, thật thú vị khi thấy rằng node2vec hoạt động tốt hơn các mô hình không đồng nhất truyền thống. Điều đó cho thấy rằng, các mô hình GNN nói chung và đặc biệt là các mô hình GNN không đồng nhất nói riêng có thể giúp thu được các kết quả tốt hơn, chứng tỏ rằng kiến trúc GNN, sử dụng một cách thận trọng các đặc trưng của nút không đồng nhất, giúp cải thiện hiệu quả của việc biểu diễn đồ thị. Mức tăng hiệu suất mà MAGNN mang lại so với mô hình tham chiếu tốt nhất (HAN) là khoảng 4-7%, điều này cho thấy rằng các cấu hình metapath chứa đựng nhiều thông tin phong phú hơn so với các lân cận dựa metapath. Trên tập dữ liệu DBLP, việc phân loại nút là một nhiệm vụ không quá quan trọng, điều này có thể thấy được thông qua việc tất cả các mô hình đều có thể đạt được các chỉ số tốt cho bài toán này. Mặc dù vậy, MAGNN vẫn vượt trội so với các mô hình tham chiếu tốt nhất 1 – 2%.

D. Phân cụm nút (RQ2)

tác giả tiến hành thử nghiệm trên tập dữ liệu IMDb và DBLP để so sánh hiệu quả của các mô hình khác nhau trong tác vụ phân cụm nút. tác giả sử dụng thông tin biểu diễn của các nút được gắn nhãn (phim trong IMDb và tác giả trong DBLP) được tạo bởi từng mô hình học máy làm thông tin đầu vào cho thuật toán phân cụm K-Means. Số cụm trong K-Means

Bảng III
KẾT QUẢ THỰC NGHIỆM

Tập dữ liệu	Độ đo	Train %	Không giám sát					Bán giám sát			
			LINE	node2vec	ESim	metapath2vec	HERec	GCN	GAT	HAN	MAGNN
IMDb	Macro-F1	20%	44.04	49.00	48.37	46.05	45.61	52.73	53.64	56.19	59.35
		40%	45.45	50.63	50.09	47.57	46.80	53.67	55.50	56.15	60.27
		60%	47.09	51.65	51.45	48.17	46.84	54.24	56.46	57.29	60.66
		80%	47.49	51.49	51.37	49.99	47.73	54.77	57.43	58.51	61.44
	Micro-F1	20%	45.21	49.94	49.32	47.22	46.23	52.80	53.64	56.32	59.60
		40%	46.92	51.77	51.21	48.17	47.89	53.76	55.56	57.32	60.50
		60%	48.35	52.79	52.53	49.87	48.19	54.23	56.47	58.42	60.88
		80%	48.98	52.72	52.54	50.50	49.11	54.63	57.40	59.24	61.53
DBLP	Macro-F1	20%	87.16	86.70	90.68	88.47	90.82	88.00	91.05	91.69	93.13
		40%	88.85	88.07	91.61	89.91	91.44	89.00	91.24	91.96	93.23
		60%	88.93	88.69	91.84	90.50	92.08	89.43	91.42	92.14	93.57
		80%	89.51	88.93	92.27	90.86	92.25	89.98	91.73	92.50	94.10
	Micro-F1	20%	87.68	87.21	91.21	89.02	91.49	88.51	91.61	92.33	93.61
		40%	89.25	88.51	92.05	90.36	92.05	89.22	91.77	92.57	93.68
		60%	89.34	89.09	92.28	90.94	92.66	89.57	91.97	92.72	93.99
		80%	89.96	89.37	92.68	91.31	92.78	90.33	92.24	93.23	94.47

Bảng IV
KẾT QUẢ THỰC NGHIỆM (%) TRÊN TẬP DỮ LIỆU IMDB VÀ DBLP ĐỐI VỚI TÁC VỤ PHÂN CỤM NÚT

Tập dữ liệu	Độ đo	Không giám sát				Bán giám sát				
		LINE	node2vec	ESim	metapath2vec	HERec	GCN	GAT	HAN	MAGNN
IMDb	NMI	1.13	5.22	1.07	0.89	0.39	7.46	7.84	10.79	15.58
	ARI	1.20	6.02	1.01	0.22	0.11	7.69	8.87	11.11	16.74
DBLP	NMI	71.02	77.01	68.33	74.18	69.03	73.45	70.73	77.49	80.81
	ARI	76.52	81.37	72.22	78.11	72.45	77.50	76.04	82.95	85.54

Bảng V
KẾT QUẢ THỰC NGHIỆM (%) TRÊN TẬP DỮ LIỆU LAST.FM ĐỐI VỚI TÁC VỤ DỰ ĐOÁN LIÊN KẾT

Dataset	Metrics	LINE	node2vec	ESim	metapath2vec	HERec	GCN	GAT	GATNE	HAN	MAGNN
Last.fm	AUC	85.76	67.14	82.00	92.20	91.52	90.97	92.36	89.21	93.40	98.91
	AP	88.07	64.11	82.19	90.11	89.47	91.65	91.55	88.86	92.44	98.93

được chọn là số lớp cho mỗi tập dữ liệu, tức là 3 cụm cho tập dữ liệu IMDb và 4 cụm cho tập dữ liệu DBLP. tác giả sử dụng NMI (normalized mutual information - thông tin chung đã chuẩn hóa) và chỉ số ARI làm độ đo chính. Do kết quả phân cụm của thuật toán K-Means phụ thuộc rất nhiều vào quá trình khởi tạo tâm cụm nên tác giả lặp lại thuật toán K-Means 10 lần cho mỗi lần chạy mô hình biểu diễn đồ thị và mỗi mô hình biểu diễn đồ thị được đánh giá trong 10 lần chạy. C kết quả trung bình thu được từ các lần chạy được thể hiện trong Bảng 4.

Từ Bảng 4, chúng ta có thể thấy rằng MAGNN luôn vượt trội so với tất cả các mô hình tham chiếu khác trong việc phân cụm nút. Lưu ý rằng tất cả các mô hình có hiệu suất trên tập dữ liệu IMDb kém hơn nhiều so với trên tập dữ liệu DBLP.

Điều này có thể do nhãn bẩn (dirty labels) của các bộ phim trong tập dữ liệu IMDb: mỗi nút bộ phim trong bộ dữ liệu IMDb gốc có nhiều thể loại và tác giả chỉ chọn loại đầu tiên làm nhãn lớp của chúng. Chúng ta có thể thấy rằng các mô hình không đồng nhất truyền thống không có nhiều ưu điểm so với các mô hình đồng nhất truyền thống trong việc phân cụm nút. Node2vec được cho là sẽ hiệu quả rõ rệt trong tác vụ phân cụm nút bởi vì đây là một cách tiếp cận dựa trên random walk, vì vậy nó buộc các nút ở gần nhau trên đồ thị cũng phải ở gần nhau trong không gian biểu diễn [33] và từ đó có thể mã hóa được thông tin vị trí của nút. Đặc điểm này vô cùng thuận lợi cho thuật toán K-Means vì nó phải phân cụm các nút dựa trên khoảng cách Euclidean giữa các điểm biểu diễn. Mặc dù vậy, các GNN nhận biết tính không đồng nhất (tức là

HAN và MAGNN) vẫn dẫn đầu về hiệu quả trong việc phân cụm nút trên cả hai bộ dữ liệu.

E. Dự đoán liên kết (RQ3)

tác giả cũng tiến hành thử nghiệm trên tập dữ liệu Last.fm để đánh giá hiệu quả của MAGNN và các mô hình tham chiếu khác đối với tác vụ dự đoán liên kết. Đối với các mô hình GNN, tác giả coi cặp người dùng-nghệ sĩ được kết nối là các cặp nút positive và coi tất cả các liên kết người dùng-nghệ sĩ không được kết nối là các cặp nút negative. tác giả thêm cùng một số cặp nút negative được lấy mẫu ngẫu nhiên vào tập validation và tập testing. Trong quá trình huấn luyện mô hình GNN, các cặp nút negative cũng được lấy mẫu theo phân phối đều. Các mô hình GNN sau đó được tối ưu hóa bằng cách tối thiểu hóa hàm mục tiêu đã mô tả trong Công thức (12). Với mỗi biểu diễn của nút người dùng \mathbf{h}_u và nút nghệ sĩ \mathbf{h}_a do mô hình tạo ra, ta tính xác suất để u và v liên kết với nhau như sau:

$$p_{ua} = \sigma(\mathbf{h}_u^\top \cdot \mathbf{h}_a) \quad (13)$$

trong đó $\sigma(\cdot)$ là hàm sigmoid. Các mô hình biểu diễn đồ thị để dự đoán liên kết được đánh giá theo AUC (diện tích phía dưới đường cong ROC) và độ chính xác trung bình (AP). tác giả tổng kết kết quả trung bình của 10 lần chạy ứng với từng mô hình biểu diễn đồ thị trong Bảng 5.

Từ Bảng 5, có thể thấy MAGNN vượt trội hơn rõ rệt so với các mô hình tham chiếu khác. Mô hình truyền thống mạnh nhất ở đây là metapath2vec, mô hình này học từ các chuỗi nút được tạo từ các random walk được định hướng bởi một metapath duy nhất. MAGNN đạt được các chỉ số cao hơn so với metapath2vec, cho thấy rằng việc xem xét một metapath duy nhất là không hoàn toàn tối ưu. Trong số các mô hình tham chiếu có dạng GNN, HAN thu được kết quả tốt nhất vì nó nhận biết được tính không đồng nhất và kết hợp nhiều metapath. MAGNN đạt được mức cải thiện xấp xỉ khoảng 6% so với HAN. Kết quả này một lần nữa giúp củng cố thêm cho khẳng định của nhóm tác giả rằng metapath context của các nút là rất quan trọng đối với việc biểu diễn nút.

F. Nghiên cứu phân tách ảnh hưởng của từng thành phần trong MAGNN (RQ4)

Để xác thực tính hiệu quả của từng thành phần trong mô hình của tác giả, tác giả tiếp tục tiến hành thử nghiệm trên các biến thể MAGNN khác nhau. Trong phạm vi báo cáo này, tác giả tổng hợp kết quả thu được từ ba bộ dữ liệu trên cả ba nhiệm vụ trong Bảng 6. Lưu ý rằng tất cả các chỉ số được thể hiện đối với tác vụ phân loại nút (tức là Macro-F1 và Micro-F1) là giá trị trung bình của các chỉ số theo các tỷ lệ tập huấn luyện khác nhau (giải thích chi tiết trong Mục V.C). Ở đây MAGNN_{rot} là mô hình được đề xuất của tác giả sử dụng bộ mã hóa dựa trên phép quay quan hệ, tức là mô hình được sử dụng để cạnh tranh với các mô hình tham chiếu khác trong Bảng 3, 4 và 5. Gọi MAGNN_{rot} là mô hình tham chiếu, MAGNN_{feat} là mô hình tương đương mà không sử dụng các đặc trưng về nội dung của nút; MAGNN_{nb} chỉ xem xét các nút lân cận dựa trên metapath; MAGNN_{sm} chỉ xem xét

một metapath tốt nhất; MAGNN_{avg} chuyển sang sử dụng mã hóa tuyến tính cấu hình metapath. Ngoài trừ những khác biệt nêu trên, tất cả các cài đặt khác đều giống nhau đối với các biến thể MAGNN này. Lưu ý rằng MAGNN_{feat} trên tập dữ liệu Last.fm tương đương với MAGNN_{rot} vì tập dữ liệu này không chứa các thuộc tính của nút.

Có thể thấy, bằng cách sử dụng các thuộc tính nội dung nút, MAGNN_{rot} đạt được sự cải thiện hiệu suất đáng kể so với MAGNN_{feat} , điều này cho thấy sự cần thiết của việc biến đổi nội dung nút để kết hợp với các đặc trưng khác của nút. So sánh MAGNN_{nb} với MAGNN_{avg} , MAGNN_{linear} và MAGNN_{rot} , ta thấy rằng việc tổng hợp các cấu hình metapath thay vì tìm các lân cận dựa trên metapath mang lại sự gia tăng về hiệu suất, điều này xác nhận tính hiệu quả của việc tổng hợp intra-metapath. Bên cạnh đó, sự khác biệt giữa kết quả của MAGNN_{sm} và MAGNN_{rot} cho thấy rằng hiệu suất của mô hình được cải thiện đáng kể bằng cách kết hợp nhiều metapath trong quá trình tổng hợp inter-metapath. Cuối cùng, kết quả của MAGNN_{avg} , MAGNN_{linear} và MAGNN_{rot} cho thấy rằng cách mã hóa dựa trên phép quay quan hệ giúp cải thiện MAGNN một lượng nhỏ. Đáng chú ý là MAGNN_{linear} lại kém hiệu quả hơn MAGNN_{avg} . Tuy nhiên, cả ba biến thể MAGNN sử dụng các bộ mã hóa khác nhau vẫn luôn hoạt động tốt hơn so với mô hình tham chiếu tốt nhất, HAN.

G. Trực quan hóa (RQ5)

Ngoài các đánh giá định lượng trên về các mô hình biểu diễn đồ thị, tác giả cũng trực quan hóa các cách biểu diễn nút để tiến hành đánh giá định tính các kết quả biểu diễn đồ thị. tác giả chọn ngẫu nhiên 30 cặp người dùng-nghệ sĩ từ tập test có nhãn positive trong bộ dữ liệu Last.fm, sau đó chiếu kết quả biểu diễn của các nút này vào không gian 2 chiều bằng cách sử dụng phương pháp t-SNE. Trong phần này, tác giả minh họa kết quả trực quan hóa của LINE, ESIm, GCN và MAGNN trong Hình 4, trong đó các điểm màu đỏ và điểm màu xanh lá cây lần lượt biểu thị người dùng và nghệ sĩ.

Dựa trên hình ảnh trực quan này, ta có thể nhanh chóng nhận ra sự khác biệt giữa các mô hình biểu diễn đồ thị về khả năng học của chúng đối với các đồ thị không đồng nhất. Là một mô hình biểu diễn đồ thị đồng nhất truyền thống, LINE không thể phân chia nút người dùng và nút nghệ sĩ thành hai nhóm khác nhau một cách hiệu quả. Ngược lại, ESIm, một mô hình không đồng nhất truyền thống, có thể phân vùng gần đúng hai loại nút. Nhờ kiến trúc GNN mạnh mẽ và bằng cách chọn các metapath phù hợp, một mô hình GNN đồng nhất như GCN có thể cô lập các loại nút khác nhau và mã hóa thông tin tương quan của các cặp nghệ sĩ-người dùng vào các biểu diễn nút. Từ Hình 4, chúng ta có thể thấy rằng mô hình MAGNN mà tác giả đề xuất thu được kết quả biểu diễn đồ thị tốt nhất, với hai nhóm người dùng và nghệ sĩ được phân tách rõ ràng và mối tương quan phù hợp của các cặp nghệ sĩ-người dùng.

VI. KẾT LUẬN

Trong phạm vi báo cáo này, nhóm tác giả đề xuất mô hình MAGNN để giải quyết ba hạn chế đặc trưng của các phương

Bảng VI
KẾT QUẢ PHÂN TÍCH ẢNH HƯỞNG CỦA TỪNG THÀNH PHẦN

Variant	IMDb				DBLP				Last.fm	
	Macro-F1	Micro-F1	NMI	ARI	Macro-F1	Micro-F1	NMI	ARI	AUC	AP
MAGNN _{feat}	48.87	50.36	5.82	5.30	92.80	93.32	77.17	82.15	N/A	N/A
MAGNN _{nb}	58.45	58.84	12.87	11.98	92.61	93.15	77.64	82.60	93.68	92.95
MAGNN _{sm}	56.77	56.64	11.90	11.84	93.19	93.69	79.48	84.39	92.54	91.52
MAGNN _{avg}	59.66	59.78	13.64	15.27	93.13	93.44	79.31	84.30	98.63	98.57
MAGNN _{linear}	57.80	57.96	9.80	8.49	93.21	93.52	78.95	83.89	98.56	98.48
MAGNN _{rot}	60.43	60.63	15.58	16.74	93.51	93.94	80.81	85.54	98.91	98.93

pháp biểu diễn đồ thị không đồng nhất hiện nay là (1) loại bỏ các thuộc tính nội dung nút, (2) loại bỏ các nút trung gian dọc theo metapath và (3) chỉ xem xét một metapath duy nhất. Cụ thể, MAGNN được xây dựng dựa trên ba thành phần chính: (1) chuyển đổi nội dung nút, (2) tổng hợp intra-metapath và (3) tổng hợp inter-metapath để xử lý từng hạn chế tương ứng kể trên. Ngoài ra, tác giả xác định định nghĩa về mã hóa cấu hình metapath, từ đó có thể trích xuất thông tin cấu trúc và ngữ nghĩa ẩn sâu trong các cấu hình metapath. tác giả đề xuất một số hàm mã hóa, trong đó bao gồm một hàm lấy ý tưởng từ mô hình biểu diễn đồ thị tri thức RotatE [24]. Trong các thử nghiệm, MAGNN đạt được kết quả tốt hơn trên cả ba bộ dữ liệu thực tế đối với các tác vụ phân loại nút, phân cụm nút và dự đoán liên kết. Nghiên cứu bóc tách ảnh hưởng của từng thành phần cũng chứng minh hiệu quả của ba thành phần chính của MAGNN trong việc tăng hiệu suất biểu diễn đồ thị. Các nghiên cứu tiếp theo có thể tập trung tìm kiếm các phương pháp áp dụng quy trình biểu diễn đồ thị này cho tác vụ dự đoán xếp hạng (hệ khuyến nghị) với dữ liệu người dùng-mặt hàng được thể hiện bởi các biểu đồ tri thức không đồng nhất [30].

TÀI LIỆU

- [1] James Atwood and Don Towsley. 2016. Diffusion-Convolutional Neural Networks. In NIPS. 1993–2001.
- [2] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and koray kavukcuoglu. 2016. Interaction Networks for Learning about Objects, Relations and Physics. In NIPS. 4502–4510.
- [3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In NIPS. 2787–2795.
- [4] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In RecSys.
- [5] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation Learning for Attributed Multiplex Heterogeneous Network. In SIGKDD. 1358–1368.
- [6] Hongxu Chen, Hongzhi Yin, Weiqing Wang, Hao Wang, Quoc Viet Hung Nguyen, and Xue Li. 2018. PME: Projected Metric Embedding on Heterogeneous Networks for Link Prediction. In SIGKDD. 1177–1186.
- [7] Kyunghyun Cho, Bart van Merriënboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. CoRR abs/1406.1078 (2014). arXiv:1406.1078
- [8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In NIPS. 3844–3852.
- [9] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. Metapath2Vec: Scalable Representation Learning for Heterogeneous Networks. In SIGKDD. 135–144.
- [10] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. 2017. Protein Interface Prediction using Graph Convolutional Networks. In NIPS. 6530–6539.
- [11] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. HIN2Vec: Explore Metapaths in Heterogeneous Information Networks for Representation Learning. In CIKM. 1797–1806.
- [12] Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. 2009. Graph-based Consensus Maximization Among Multiple Supervised and Unsupervised Models. In NIPS. 585–593.
- [13] Aditya Grover and Jure Leskovec. 2016. Node2Vec: Scalable Feature Learning for Networks. In SIGKDD. 855–864.
- [14] William L. Hamilton, Zhitaoying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In NIPS. 1024–1034.
- [15] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. 2010. Graph Regularized Transductive Classification on Heterogeneous Information Networks. In ECML PKDD. 570–586.
- [16] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In ICLR.
- [17] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated Graph Sequence Neural Networks. In ICLR.
- [18] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In ICLR.
- [19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In ICLR.
- [20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In NIPS. 3111–3119.
- [21] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In SIGKDD. 701–710.
- [22] Jingbo Shang, Meng Qu, Jialu Liu, Lance M. Kaplan, Jiawei Han, and Jian Peng. 2016. Meta-Path Guided Embedding for Similarity Search in Large-Scale Heterogeneous Information Networks. CoRR abs/1610.09769 (2016). arXiv:1610.09769
- [23] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2019. Heterogeneous Information Network Embedding for Recommendation. IEEE Transactions on Knowledge and Data Engineering 31, 2 (2019), 357–370.
- [24] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In ICLR.
- [25] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In WWW. 1067–1077.
- [26] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. CoRR abs/1706.02263 (2017). arXiv:1706.02263
- [27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In NIPS. 5998–6008.
- [28] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In ICLR.

- [29] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural Deep Network Embedding. In SIGKDD. 1225–1234.
- [30] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge Graph Convolutional Networks for Recommender Systems. In WWW. 3307–3313.
- [31] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In WWW. 2022–2032.
- [32] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. 2015. Network Representation Learning with Rich Text Information. In IJCAI. 2111–2117.
- [33] Jiaxuan You, Rex Ying, and Jure Leskovec. 2019. Position-aware Graph Neural Networks. In ICML. 7134–7143.
- [34] Jian Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. 2018. GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. In UAI. 339–349.
- [35] Jian Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. STAR-GCN: Stacked and Reconstructed Graph Convolutional Networks for Recommender Systems. In IJCAI. 4264–4270.