THỰC NGHIỆM SỬ DỤNG CƠ SỞ DỮ LIỆU ĐỒ THỊ TRONG HỆ GỢI Ý

Trần Thiên Thành
Khoa Công nghệ thông tin
Trường Đại học Quy Nhơn
Bình Định, Việt Nam
tranthienthanh@qnu.edu.vn

Lê Quyền
Trường THPT Nguyễn Du
Sở Giáo dục và Đào tạo Phú Yên
Phú Yên, Việt Nam
lquyen.thpt.nd@phuyen.edu.vn

Nguyễn Thị Loan
Khoa Công nghệ thông tin
Trường Đại học Quy Nhơn
Bình Định, Việt Nam
nguyenthiloan@qnu.edu.vn

***Tóm tắt* — Mục tiêu của bài báo là thực nghiệm để thấy khả năng ứng dụng của cơ sở dữ liệu mô hình đồ thị trong các hệ gợi ý. Bài báo tập trung thực nghiệm hai nội dung trên cơ sở dữ liệu Neo4j: i) khả năng giải quyết bài toán gợi ý; ii) ưu thế về tốc độ xử lý dữ liệu để gợi ý. Kết quả thực nghiệm cho thấy khi sử dụng cơ sở dữ liệu mô hình đồ thị việc giải quyết bài toán gợi ý sản phẩm cho người dùng khá đơn giản và tốc độ xử lý nhanh, có khả năng đáp ứng cho các hệ gợi ý thời gian thực.**

Keywords: recommendation system; graph database; collaborative filtering; neo4j.

I. ĐẶT VẤN ĐỀ

Hệ gợi ý ngày nay đã được áp dụng rộng rãi trong các hệ thống kinh doanh trực tuyến nhằm gợi ý cho người dùng những sản phẩm phù hợp từ đó làm tăng doanh thu cho doanh nghiệp [3]. Nhiều hệ thống đã ứng dụng thành công như Amazon, Ebay, Netflix,.. Với sự tăng trưởng ngày càng nhanh của các hình thức kinh doanh trực tuyến các hệ thống ngày càng tăng quy mô dữ liệu người dùng, sản phẩm, đánh giá của người dùng,... Do đó để xây dựng một hệ gợi ý tích hợp trong các hệ thống kinh doanh cần tổ chức lưu trữ, tính toán nhanh để đáp ứng nhu cầu gợi ý tức thời cho người dùng. Trong các hệ thống kinh doanh hiện tại, việc lưu trữ đa số dùng dữ liệu mô hình quan hệ. Mô hình này dễ dàng lưu trữ và xử lý dữ liệu nhưng hạn chế trong tính toán và xử lý dữ liệu lớn với nhiều mối quan hệ. Cơ sở dữ liệu mô hình đồ thị với các kỹ thuật lưu trữ tốt, mô hình dữ liệu tập trung nhiều vào các mối quan hệ nên thuận lợi cho các dữ liệu có nhiều mối quan hệ như dữ liệu người dùng mạng xã hội.

Đã có nhiều nghiên cứu ứng dụng mô hình dữ liệu đồ thị vào hệ gợi ý theo các cách tiếp cận khác nhau ([1][2]). Tuy nhiên chưa có nghiên cứu thực nghiệm đánh giá khả năng triển khai hệ gợi ý trên một hệ cơ sở dữ liệu đồ thị cụ thể để thấy được cách triển khai và khả năng ứng dụng. Trong [7] đã đưa ra cách tổ chức cơ sở dữ liệu mô hình đồ thị cho bài toán cơ bản của hệ gợi ý là gợi ý sản phẩm dựa vào lọc cộng tác, bước đầu thực nghiệm cho thấy khả năng triển khai ứng dụng của cách tiếp cận này. Bài báo này tiếp tục kết quả trong [7], tiến hành thực nghiệm trên hai khía: i) Tốc độ xử lý để gợi ý sản phẩm cho người dùng; ii) Đánh giá độ chính xác của các gợi ý.

Bài báo được tổ chức như sau: phần 2 trình bày cách ứng dụng cơ sở dữ liệu mô hình đồ thị cho hệ gợi ý; Phần 3 trình bày hai thực nghiệm về tốc độ xử lý và cách đánh giá độ chính xác khi sử dụng cơ sở dữ liệu mô hình đồ thị cho việc gợi ý. Cuối cùng là kết luận và hướng phát triển.

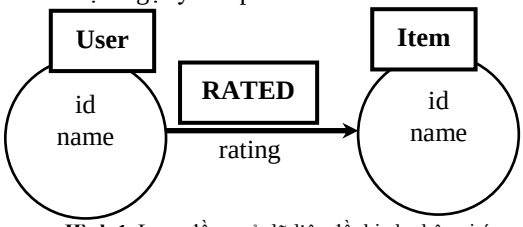
II. SỬ DỤNG CƠ SỞ DỮ LIỆU MÔ HÌNH ĐỒ THỊ TRONG HỆ GỢI Ý

A. Bài toán gợi ý sản phẩm cho người dùng

Một hệ gợi ý gồm *N* người dùng (users) và *M* sản phẩm (items), mỗi người dùng có xếp hạng (rating) của mình về một số sản phẩm. Cần dự đoán mức độ ưa thích của người dùng với các sản phẩm mà họ chưa xếp hạng để gợi ý cho người dùng lựa chọn những sản phẩm này trong những tình huống cụ thể.

B. Tổ chức cơ sở dữ liệu mô hình đồ thị

Trong [7] đã xây dựng lược đồ cơ sở dữ liệu mô hình đồ thị để gợi ý sản phẩm như sau:



Hình 1. Lược đồ cơ sở dữ liệu đồ thị cho hệ gợi ý

Dựa mô hình dữ liệu trên trong [7] đã xây dựng thuật toán gợi ý sản phẩm cho người dùng dựa vào lọc cộng tác sử dụng độ đo Pearson (chi tiết xem trong [7]). Trên cơ sở thuật toán này chúng tôi cài đặt các hàm:

`predictRatingCosin(user, item)`: dự đoán xếp hạng một sản phẩm của một người dùng bằng độ đo Cosin.

`predictRatingPearson(user, item)`: dự đoán xếp hạng một sản phẩm của một người dùng bằng độ đo Pearson.

C. Đánh giá gợi ý

Dựa vào kết quả dự đoán và gợi ý trên chúng tôi xây dựng thuật toán để đánh giá độ chính xác của dự đoán và gợi ý. Chúng tôi sử dụng các độ đo MAE, RMSE để tính độ lệch của dự đoán xếp hạng sản phẩm của người dùng và độ đo RECALL, PRECISION, F1 SCORE cho độ chính xác của gợi ý sản phẩm cho người dùng. Chi tiết các độ đo này có thể xem trong [6].

Để đánh giá độ chính xác ta cần hai tập dữ liệu một tập làm dữ liệu huấn luyện (tính toán) một tập làm dữ liệu đánh giá. Chúng tôi nạp hai tập dữ liệu này lên một cơ sở dữ liệu đồ thị trong đó để phân biệt giữa hai dữ liệu này thì với dữ liệu đánh giá quan hệ giữa người dùng và sản phẩm được đặt tên là RATED_LATE thay cho RATED của dữ liệu huấn luyện.

Thuật toán 1. Đánh giá độ lệch theo RMSE, MAE

Input: Cơ sở dữ liệu *D* đã bao gồm dữ liệu huấn luyện và dữ liệu đánh giá.
Output: Độ lệch của dự đoán xếp hạng theo độ đo RMSE, MAE.
Action:
Duyệt từng cặp người dùng (u) và sản phẩm (i) theo quan hệ RATED_LATE (r):

- Thực hiện tính predictRating = predictRating(u, i)
- Tính độ lệch r.rating – predictRating

Tính độ lệch error = căn bậc hai trung bình tổng độ lệch bình phương (cho RMSE) hoặc trung bình độ lệch (cho MAE).

Câu lệnh trong Cypher có dạng:

```
MATCH(u:User)-[r:RATED_LATE]->(i:Item)
WITH u, i, abs(r.rating –
predictRating(u, i)) as errMAE,
sum((r.rating – predictRating(u,
i))^2) as errRMSE, count(*) as num
RETURN avg(err) as mae,
sqrt(errRMSE/num) as rmse
```

Để đánh giá độ chính xác, độ phủ của gợi ý sản phẩm cho người dùng chúng tôi sử dụng cách tiếp cận như phân lớp nhị phân bằng cách quy định ngưỡng cho các xếp hạng, chẳng hạn với xếp hạng từ 1 đến 5 quy định: nếu xếp hạng ≥ 3 thì xem như người dùng thích sản phẩm, ngược lại là không thích.

Thuật toán 2. Đánh giá độ chính xác của gợi ý sản phẩm cho người dùng

Input: Cơ sở dữ liệu *D* đã bao gồm dữ liệu huấn luyện và dữ liệu đánh giá.
Output: Độ lệch của gợi ý Precision, Recall, F1
Action:
Duyệt từng cặp người dùng (u) và sản phẩm (i) theo quan hệ RATED_LATE (r):

- Thêm thuộc tính like vào quan hệ RATED_LATE là 1 nếu rating>=3, 0 nếu ngược lại
- Thực hiện tính predictRating = predictRating(u, i)
- Lưu vào quan hệ RATED_LATE thuộc tính like_late = 1 nếu predictRating>=3, ngược lại là 0

Duyệt từng cặp người dùng (u) và sản phẩm (i) theo quan hệ RATED_LATE (r):

- Nếu like = 1 và like_late = 1 thì TP = TP + 1
- Nếu like = 1 và like_late = 0 thì FN = FN + 1
- Nếu like = 0 và like_late = 1 thì FP = FP + 1
- Nếu like = 0 và like_late = 0 thì TN =TN + 1

Precision = TP/(TP + FP)
Recall = TP/(TP + FN)
F1Score = 2 * Precision * Recall / (Precision + Recall)
return Precision, Recall, F1Score

Câu lệnh trong Cypher có dạng:

```
MATCH(u:User)-[r:RATED_LATE]->(i:Item)
SET like = CASE WHEN r.rating >=3 THEN
1 ELSE 0
SET like_late = CASE WHEN
predictRating(u, i)>=3 THEN 1 ELSE 0
WITH u, i, r,
CASE WHEN r.like=1 and r.like_late=1
THEN 1 ELSE 0 as TP,
CASE WHEN r.like=1 and r.like_late=0
THEN 1 ELSE 0 as FN,
CASE WHEN r.like=0 and r.like_late=1
THEN 1 ELSE 0 as FP,
CASE WHEN r.like=0 and r.like_late=0
THEN 1 ELSE 0 as TN
RETURN sum(TP)/(sum(TP)+sum(FP)) as
Precision,
sum(TP)/(sum(TP)+sum(FN)) as Recall,
2*Precision*Recall/(Precision +
Recall) as F1Score
```

III. THỰC NGHIỆM

Mã lệnh thực nghiệm của bài báo được công bố tại:
https://github.com/thanhtranthien/Recomendation-System-on-Graph-Database.

Dữ liệu thực nghiệm: là các bộ dữ liệu Movielens download từ web site https://grouplens.org/datasets/movielens/. Chúng tôi sử dụng 03 bộ dữ liệu để thực nghiệm:

BẢNG I. CÁC BỘ DỮ LIỆU THỰC NGHIỆM

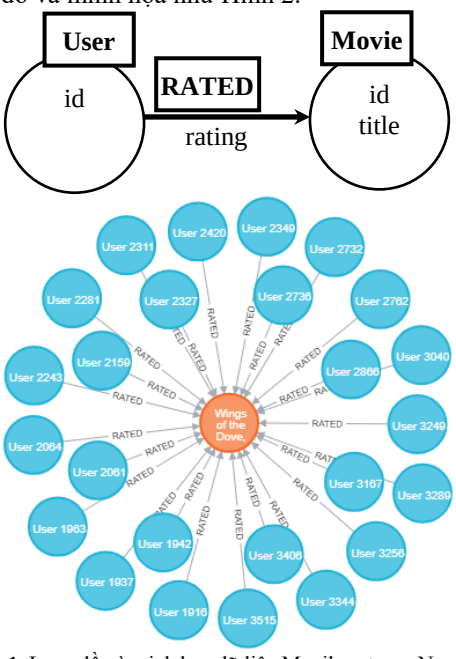
Tên bộ dữ liệu	Số người dùng	Số bộ phim	Số lượt rating
Movielens100K	943	1682	100000
Movielens1M	6040	3952	1000000
Movielens20M	138493	27278	20000263

Cơ sở dữ liệu mô hình đồ thị: Neo4j Desktop phiên bản 1.4.3.

Ngôn ngữ lập trình: Python phiên bản 3.9.4, kết hợp thư viện py2neo.

Máy tính: Laptop CPU Core i5 7440HQ, RAM 16GB, SSD 256GB, Hệ điều hành Windows 10 Pro.

Dữ liệu được nạp lên cơ sở dữ liệu Neo4j có lược đồ và minh họa như Hình 2.



Hình 1. Lược đồ và minh họa dữ liệu Movilens trong Neo4j [7]

Thực nghiệm 1: Đánh giá tốc độ gợi ý.
Mục đích thực nghiệm là đánh giá tốc độ gợi ý các bộ phim cho một người dùng bằng 2 cách: i) Nạp và xử lý trực tiếp dữ liệu từ file .csv bằng Python; ii) Dữ liệu lưu trong cơ sở dữ liệu Neo4j.
Thao tác thực nghiệm: tính và đưa ra danh sách các bộ phim cho một người dùng cụ thể sử dụng độ đo Cosin.
Kết quả thực nghiệm trên 3 bộ dữ liệu như sau:

BẢNG II. KẾT QUẢ THỰC NGHIỆM VỀ TỐC ĐỘ XỬ LÝ

Bộ dữ liệu	Dữ liệu csv	Dữ liệu Neo4j
Movielens100K	0.16s	0.19s
Movielens1M	2.4s	1.6s
Movielens20M	Không xử lý được	24.23s

Qua thực nghiệm cho thấy với dữ liệu lớn thì việc gợi ý trên dữ liệu được lưu trữ trên cơ sở dữ liệu mô hình đồ thị sẽ hiệu quả hơn về thời gian. Qua đó cũng có thể thấy được có thể sử dụng mô hình đồ thị trong các hệ gợi ý thực tế.

Thực nghiệm 2. Đánh giá độ chính xác gợi ý.
Dữ liệu đánh giá: bộ dữ liệu Movielens100K đã được chia làm 2 phần đánh giá và huấn luyện theo tỷ lệ 20%, 80% với 5 cách chia khác nhau được cung cấp bởi Movilens Group. Cách đánh giá được thực hiện theo Thuật toán 1 và Thuật toán 2.
Kết quả đánh giá như trong Bảng 3.

BẢNG III. KẾT QUẢ THỰC NGHIỆM ĐỘ CHÍNH XÁC

	RMSE	MAE	PRE.	REC.	F1 SCORE
Cosin	1.047	0.831	0.870	0.895	0.882
Pearson	0.987	0.764	0.897	0.869	0.883

IV. KẾT LUẬN

Qua việc thực nghiệm ta thấy việc sử dụng cơ sở dữ liệu mô hình đồ thị để lưu trữ và tính toán cho gợi ý có 2 ưu điểm: i) Tốc độ tính toán nhanh vì trong chức năng lọc cộng tác dùng nhiều thao tác duyệt quan hệ; ii) Công cụ truy vấn của các cơ sở dữ liệu (cụ thể là Cypher của Neo4j trong thực nghiệm) đủ khả năng tính toán cho những chức năng cơ bản của một hệ gợi ý và sử dụng những câu lệnh khá đơn giản.

Trong các nghiên cứu tiếp theo chúng tôi sẽ khai thác các thư viện của cơ sở dữ liệu đồ thị kết hợp với các kỹ thuật Graph Learning để giải bài toán gợi ý bằng phương pháp học máy. Cùng với đó chúng tôi cũng tiếp tục phát triển các kỹ thuật để tăng tốc độ xử lý bằng các kỹ thuật tính toán song song trên cụm máy tính để có thể gợi ý với cơ sở dữ liệu lớn.

TÀI LIỆU THAM KHẢO

[1] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, Qing He, "A Survey on Knowledge Graph-Based Recommender Systems", IEEE Transactions on Knowledge and Data Engineering, 2020.

[2] Noor Mohammedali, "Recommendation System Based on Graph Database Techniques", International Research Journal of Engineering and Technology, Vol. 06, 2019.

[3] Aggarwal, Charu C., "Recommender Systems", Springer, 2016.

[4] Aleksa Vukotic, Nicki Watt, "Neo4j in Action", Manning Publications, 2015

[5] LipiShah, HetalGaudani, PremBalani, "Survey on Recommendation System", International Journal of Computer Applications, Volume 137, No. 7, 2016.

[6] Vũ Sơn Lâm, Lê Quang Hùng, Nguyễn Văn Vinh, "Đánh giá hệ gợi ý: Khảo sát và thực nghiệm", Hội thảo quốc gia lần thứ XXIII: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – Quảng Ninh, 5-6/11/2020.

[7] Trần Thiên Thành, "Sử dụng cơ sở dữ liệu mô hình đồ thị trong hệ gợi ý", Kỳ yếu Hội thảo CITA 2021, Trường Đại học Công nghệ thông tin truyền thông Việt – Hàn, Đà Nẵng, 2021.