Lời cảm ơn

Lời đầu tiên em xin được gửi lời cảm ơn chân thành đến các thầy giáo, cô giáo thuộc trường đại học Bách Khoa Hà Nội. Đặc biệt là các thầy, cô giáo thuộc Viện Công nghệ thông tin và Truyền thông. Chính các thầy cô giáo đã trang bị cho em những kiến thức quý báu trong thời gian em học tập và nghiên cứu tại trường. Đồng thời em cũng xin được gửi lời cảm ơn đặc biệt đến TS. Thân Quang Khoát, ThS. Ngô Văn Linh, các thầy là người đã chỉ dẫn tận tình, cho em những kinh nghiệm quý báu để em có thể hoàn thành đồ án tốt nghiệp này. Các thầy luôn động viên, giúp đỡ em trong những thời điểm khó khăn.

Em cũng xin gửi lời cảm ơn chân thành tới các thầy cô phòng nghiên cứu Khoa học dữ liệu thuộc Viện Công Nghệ Thông Tin và Truyền Thông đã tạo điều kiện cho em thực hành thử nghiệm trên các máy tính, thiết bị của phòng nghiên cứu.

Em xin gửi lời cảm ơn tới gia đình và bạn bè. Lời động viên tinh thần từ gia đình và bạn bè luôn là động lực để em tiến lên phía trước.

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Hệ gợi ý sử dụng dữ liệu phản hồi ẩn (*implicit feedback*) ngày càng được quan tâm nghiên cứu phát triển. Gần đây, các hệ gợi ý sử dụng cả dữ liệu tiềm ẩn cho thấy độ chính xác cao hơn các hệ thống chỉ sử dụng phản hồi tường minh *(explicit feedback)*. Việc thu thập dữ liệu phản hồi tiềm ẩn có chi phí thấp hơn, số lượng lớn hơn so với dữ liệu phản hồi tường minh. Chính vì vậy nó đang trở thành một lĩnh vực nghiên cứu được phát triển mạnh.

Trong các hệ gợi ý sử dụng thuật toán lọc cộng tác dựa trên sở thích người dùng, thường chỉ sử dụng phản hồi tường minh - explicit feedback (ratings - xếp hạng) hoặc phản hồi ẩn - implicit feedback (xem, click, mua sản phẩm). Explicit feedback chính xác hơn nhưng khó thu thập thông tin từ người dùng trong khi implicit feedback dễ thu thập hơn nhưng lại mang ít thông tin về sở thích của người dùng. Trong các tài liệu hiện nay, các mô hình được phát triển riêng biệt cho một trong hai dạng feedback do sự biểu diễn khác nhau của chúng. Tuy nhiên trong thực tế, implicit feedback của người dùng có thể bổ sung thông tin về explicit feedback. Ta mong muốn có thể sử dụng cả hai hình thức feedback không đồng nhất này của người dùng để tạo ra mô hình gợi ý chính xác hơn.

Vì vậy, trong đồ án này em tìm hiểu và thử nghiệm ba mô hình gợi ý sử dụng phản hồi ẩn từ người dùng, trong đó một mô hình có khả năng kết hợp đồng thời hai loại dữ liệu implicit feedback và explicit feedback để tăng độ chính xác của hệ gợi ý. Trong hai mô hình sử dụng phản hồi ẩn, mô hình NeuMF khá thú vị khi tận dụng được điểm mạnh của mạng nơ-ron cho bài toán gợi ý, nhưng mô hình này có hạn chế là explcit feedback và implcit feedback được sử dụng như nhau. Tất cả những mô hình này em thử nghiệm và đánh giá trên hai bộ dữ liệu thương mại điện tử trong thực tế.

Mục lục

Danh sách các từ viết tắt và thuật ngữ

CF	Collaborative filtering
MF	Matrix factorization
COMF	Co-rating model
VALS	View-enhanced eALS
ALS	Alternating least squares
NCF	Neural Collaborative Filtering
GMF	Generalized Matrix Factorization
MLP	Multi-Layer Perceptron
NeuMF	Neural Matrix Factorization
ReLU	Rectifier
Tanh	Hyperbolic tangent
NDCG	Normalized discounted cumulative gain
HR	Hit Ratio

Danh sách các kí hiệu sử dụng

N	Sỗ lượng người dùng
M	Số lượng sản phẩm
$\mathbf{R}$	Ma trận đánh giá của người dùng và sản phẩm
$\mathbf{P}$	Ma trận nhân tố ẩn của toàn bộ người dùng
$\mathbf{Q}$	Ma trận nhân tố ẩn của toàn bộ sản phẩm
$\mathbf{p}_u$	Vector ẩn của người dùng u
$\mathbf{q}_i$	Vector ẩn của sản phẩm i
$\hat{y}_{ui}$	Dự đoán *implicit* feedback
$\hat{x}_{ui}$	Dự đoán *explicit* feedback

Danh sách hình vẽ

Danh sách bảng

Chương 1 Giới thiệu đề tài

1.1 Đặt vấn đề

Với sự tăng trưởng nhanh chóng và phổ biến của thương mại điện tử, các mặt hàng được trưng bày một cách dễ dàng tới người dùng trên Internet. Do đó, một lượng thông tin khổng lồ đến với mỗi người dùng, nhưng mỗi người đều chỉ có thể theo dõi một lượng thông tin hữu hạn. Vì vậy, nhu cầu chọn lọc thông tin trình diễn cho người dùng ngày càng cần thiết. Thách thức được đặt ra là giúp khách hàng lựa chọn sản phẩm mà họ yêu thích nhất giữa một số lượng lớn những mặt hàng. *Hệ gợi ý* (*Recommendation System*) [1] – một trong những chủ đề đang thú hút nhiều sự chú ý trong nghiên cứu và ngoài doanh nghiệp.

Những năm gần đây, nhiều nhà nghiên cứu tìm cách giải quyết bài toán gợi ý và đạt được nhiều phương pháp giải quyết. Tuy nhiên, dữ liệu ngày càng lớn và phức tạp, những thuật toán dùng tập luật đơn giản khó đáp ứng hiệu quả cho hệ gợi ý. Hơn nữa, thông tin phản hồi tường mình của người dùng về sản phẩm khan hiếm và khó thu thập . Do vậy, gần đây các nghiên cứu về hệ gợi ý có sự dịch chuyển từ phản hồi tường minh sang phản hồi ẩn (xem, click, mua sản phẩm).

Bài toán gợi ý cần nhiều hơn những mô hình học máy *(Machine Learning)* mới để giải quyết những vấn đề khó khăn. Đặc biệt trong bài toán sử dụng phản hồi ẩn từ người dùng.

1.2 Định hướng giải quyết

Các hệ thống gợi ý đóng vai quan trọng trong việc xử lý quá tải thông tin, và được nhiều dịch vụ trực tuyến áp dụng như thương mại điện tử, mạng xã hội,... Giải pháp của những hệ thống gợi ý này là mô hình hóa hành động của người dùng dựa trên lịch sử tương tác của họ (view, click, addtocart, purchase...). Hệ gợi ý thường đi theo một trong hai chiến lược chính: dựa trên nội dung (*content-based*) hoặc lọc cộng tác (*collaborative filtering*) [4].

Content-based: Hệ gợi ý dựa trên nội dung [13] tạo ra một hồ sơ cho mỗi người dùng hoặc sản phẩm. Giả dụ, hồ sơ của một bộ phim có thể chứa những thuộc tính như thể loại, diễn viên tham gia, đạo diễn... Từ hồ sơ, hệ thống có thể gợi ý những sản phẩm phù hợp đến người dùng. Tuy nhiên chiến lược này yêu cầu thu thập nhiều thông tin và đôi khi không dễ dàng để thu thập những thông tin đó.

Collaborative filtering: Lọc cộng tác [14] là một phương pháp phổ biến để xây dựng các hệ thống gợi ý, lọc cộng tác dựa vào phản hồi trong quá khứ của người dùng mà không yêu cầu việc tạo ra những hồ sơ tường minh như hệ gợi ý dựa trên nội dung. Phương pháp này phân tích sự tương đồng giữa người dùng và sự tương đồng giữa sản phẩm để phát hiện những tương tác người dùng – sản phẩm tiềm năng. Phản hồi trong quá khứ của người dùng có thể là những giao dịch trước đó hoặc những đánh giá của người dùng về sản phẩm. Dữ liệu phản hồi của người dùng thường có hai loại: phản hồi tường minh (**explicit feedback**) và (**implicit feedback**). Phản hồi tường minh phản ánh trực tiếp sở thích của người dùng về sản phẩm (như đánh giá từ 1-5 sao cho sản phẩm). Phản hồi ẩn thể hiện tương tác giữa người dùng và sản phẩm (như xem, mua sản phẩm).

Những năm gần đây, các nghiên cứu về hệ gợi ý có sự dịch chuyển từ sử dụng phản hồi tường minh sang sử dụng phản hồi ẩn hoặc kết hợp cả hai loại này. Phản hồi ẩn liên quan đến hành động tự nhiên của người dùng xảy ra khi sử dụng một hệ thống, qua đó ta dễ dàng thu thập. Tuy nhiên phản hồi ẩn ít chính xác và ít chi tiết hơn phản rồi tường minh. Ví dụ, một người dùng có thể xem một bộ phim và không chỉ ra có thích hay không sau khi xem, vì vậy chỉ biết người dùng đã xem, còn hành động xem không chắc chắn rằng người dùng đã thích bộ phim đó. Ta có thể thấy được sự cân bằng giữa số lượng và chất lượng khi so sánh phản hồi tường minh và phản hồi ẩn. Phản hồi tường minh có chất lượng cao hơn nhưng số lượng ít hơn trong khi phản hồi ẩn có chất lượng thấp hơn nhưng số lượng dồi dào hơn. Nhận thấy, hai hình thức phản hồi này thực sự có thể bổ sung cho nhau. Tuy nhiên, phần lớn các mô hình hiện tại chỉ xem xét việc sử dụng phản hồi tường minh và một số các mô hình được thiết kế đặc biệt cho các phản hồi ẩn. Có ít mô hình thiết kế để sử dụng cả hai loại dữ liệu này, mặc dù trong thực tế phản hồi tường minh và phản hồi ẩn đều có thể được thu thập.

Trong phương pháp lọc cộng tác, kĩ thuật phân rã ma trận (MF - Matrix factorization) [9] là một kĩ thuật tốt dựa trên mô hình nhân tố ẩn (*latent factor*). Nhiều nỗ lực đề xuất để cải tiến MF, chẳng hạn như tích hợp với mô hình dựa trên hàng xóm (neighbor-based models) [10]. Mặc dù thấy được hiệu quả của MF cho lọc cộng tác, nhưng độ chính xác có thể bị cản trở bởi hàm đo độ tương tác giữa user và item. Tương tác giữa người dùng và sản phẩm được mô hình hóa bằng tích vô hướng (*inner product*) giữa hai vector nhân tố ẩn của người dùng và sản phẩm. Tác giả Xiangnan He và cộng sự [6] chỉ ra rằng tính đơn giản của hàm tích vô hướng không đủ để khai thác các liên hệ phức tạp giữa các thuộc tính ẩn. Do đó, nhóm tác giả đề xuất mô hình NeuMF để mô hình

hóa tương tác giữa các thuộc tính ẩn của user và item tốt hơn.

Từ những điều trên, trong đồ án này em tìm hiểu và thử nghiệm mô hình View-enhanced eALS (VALS) và NeuMF sử dụng phản hồi ẩn từ người dùng cho hệ gợi ý; mô hình Co-rating Model (COMF) có khả năng giải quyết bài toán kết hợp hai loại dữ liệu implicit và explicit feedback. Mặc dù Neural Collaborative Filtering (NeuMF) chỉ sử dụng implicit feedback nhưng tận dụng được ưu điểm của mạng nơ-ron vào giải quyết bài toán gợi ý. Để đánh giá hiệu quả của các mô hình, em thử nghiệm trên một vài bộ dữ liệu thương mại điện tử. Chi tiết các mô hình và những thử nghiệm đánh giá được trình bày trong các mục tiếp theo.

1.3 Bố cục của đồ án

Đồ án có bố cục như sau:

Chương 1: Giới thiệu đề tài.

Chương 2: Trình bày về cơ sở lý thuyết và những kiến thức liên quan đến việc xây dựng các mô hình gợi ý.

Chương 3: Trình bày các mô hình VALS, NeuMF, COMF. Phân tích điểm mạnh, và hạn chế của các mô hình.

Chương 4: Trình bày phương pháp thử nghiệm, kết quả đánh giá các mô hình và phân tích hiệu quả các mô hình.

Chương 5: Trình bày tóm tắt kết quả đạt được trong đồ án và hướng đi tiếp theo.

Chương 2 Cơ sở lý thuyết

Phần này trình bày các khái niệm và kiến thức cơ bản của các phương pháp được sử dụng trong đồ án. Tập trung vào các khái niệm trong bài toán gợi ý, các phương pháp giải quyết bài toán gợi ý nói chung và bài toán gợi ý sử dụng dữ liệu implicit feedback nói riêng.

2.1 Hệ gợi ý

2.1.1 Khái niệm hệ gợi ý

Hệ gợi ý (*Recommendation system*) là một lớp con của hệ lọc thông tin. Hệ gợi ý có khả năng phán đoán đánh giá của người dùng về một vật phẩm nào đó. Từ những phán đoán này, hệ thống có thể đưa ra gợi ý những vật phẩm phù hợp với người dùng ngay cả khi vật phẩm đó không phổ biến với đa số những người dùng khác. Điều này giúp tiết kiệm thời gian tìm kiếm của người dùng khi mua hàng và tăng khả năng bán sản phẩm của cửa hàng.

Bài toán gợi ý trong học máy có tuổi đời ít hơn so với bài toán phân lớp (classification) hay bài toán hồi quy (regresstion) do thương mại điện tử được thực sự bùng nổ trong khoảng mười năm trở lại đây. Có hai thực thể chính trong một hệ gợi ý là người dùng (**user**) và sản phẩm (**item**).

2.1.2 Bài toán gợi ý

Bài toán gợi ý có các thành phần cơ bản sau:

Đầu vào của bài toán gợi ý: Hai tập đối tượng chính là tập người dùng và tập các sản phẩm. Ta gọi người dùng là **user**, còn sản phẩm trong bài toán là **item**. Trong bài toán gợi ý quảng cáo item là các banner quảng cáo; trong gợi ý phim item là các bộ phim; trong gợi ý tin tức, các item là các tin tức. Ngoài ra tùy các mô hình giải quyết bài toán gợi ý, sẽ cần các thông tin khác như nội dung (*content*) về sản phẩm, lịch sử xem/mua/... của người dùng với các sản phẩm.

Đầu ra của bài toán gợi ý: Tập các item được gợi ý cho từng user. Trong đó tập item được gợi ý có thể được sắp xếp theo mức độ phù hợp với user. Mức độ phù hợp của item càng cao càng có mức ưu tiên cao khi quyết định gợi ý cho người dùng.

Phạm vi áp dụng bài toán gợi ý: Phạm vi bài toán gợi ý bao gồm nhiều lĩnh vực. Điển hình như thương mại điện tử, báo chí online, phim truyện,.. Các đối tượng người dùng áp dụng tương ứng là. Người dùng truy cập các trang báo điện tử cần được gợi ý các tin tức bài báo thuộc lĩnh vực người đó quan tâm. Người dùng truy cập vào các trang bán hàng online cần được gợi ý các sản phẩm phù hợp với nhu cầu mua sắm.

Như vậy, mục đích của bài toán gợi ý là từ một tập user và một tập item, cần đưa ra được các item phù hợp với nhu cầu của mỗi user.

2.1.3 Dữ liệu phản hồi từ người dùng

Dữ liệu phản hồi của người dùng thường có hai loại: phản hồi tường minh (**explicit feedback**) và phản hồi ẩn (**implicit feedback**).

Phản hồi tường minh: Là loại phản hồi thường ở dạng xếp hạng (ratings) của người dùng đối với các sản phẩm. Ví dụ các trang Amazon, Lazada có mục đánh giá sản phẩm theo mức thang từ 1 – 5 sao tương ứng từ rất không hài lòng đến rất hài lòng. Ở đây, ta biết người dùng thích hoặc không thích một sản phẩm ở mức nào, nhưng dữ liệu này rất khó thu thập do người dùng thường không dành thời gian để xếp hạng các sản phẩm. Phản hồi tường minh nắm bắt sở thích của người dùng theo cách trực tiếp và chi tiết nên phản hồi tường minh có giá trị cao nhưng dữ liệu phản hồi tường minh thường không sẵn có và rất khó khăn trong việc thu thập.

Phản hồi ẩn: Là những hành vi phản ánh một cách gián tiếp thái độ, sự quan tâm của người dùng đến sản phẩm. Phản hồi ẩn có thể là lịch sử mua hàng, click chuột, lịch sử tìm kiếm,... Tuy giá trị thông tin của phản hồi ẩn không cao bằng phản hồi tường minh nhưng việc thu thập lại nhanh chóng và chi phí thấp.

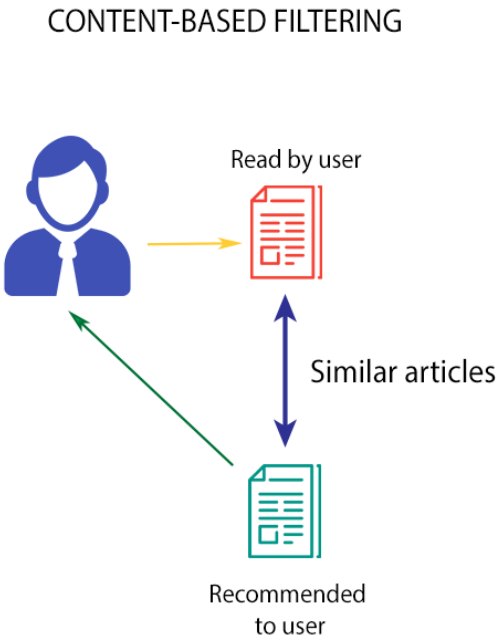
2.2 Một số hướng tiếp cận bài toán gợi ý

Có hai hướng tiếp cận chính với bài toán gợi ý là *lọc dựa trên nội dung* (*Content-based Filtering*) và *lọc cộng tác* (*Collaborative Filtering*) [4]

2.2.1 Lọc dựa trên nội dung (Content-based Filtering)

Content-based Filtering đánh giá đặc tính của items được gợi ý. Ví dụ: một user xem rất nhiều các bộ phim có tính hài hước, vậy thì gợi ý một bộ phim trong cơ sở dữ liệu có chung đặc tính hài hước tới user này, ví dụ phim Ghét

thì yêu thôi. Như tên gọi của phương pháp này, việc gợi ý phụ thuộc vào thuộc tính của các sản phẩm, áp dụng tốt đối với các sản phẩm giàu nội dung như các sản phẩm thuộc lĩnh vực truyền thông, quảng cáo, y tế. Đặc biệt cách tiếp cận này có thể gợi ý cho các sản phẩm mới, thích hợp khi danh sách sản phẩm được cập nhật liên tục và giải quyết tốt vấn đề *Cold Start* [12] trong bài toán gợi ý. Hình 1 mô phỏng về cách tiếp cận này.



Hình 1: Hướng tiếp cận Content-based Filtering
(Nguồn `https://www.matrixanalyticscorp.com/recommendation-systems.html`)

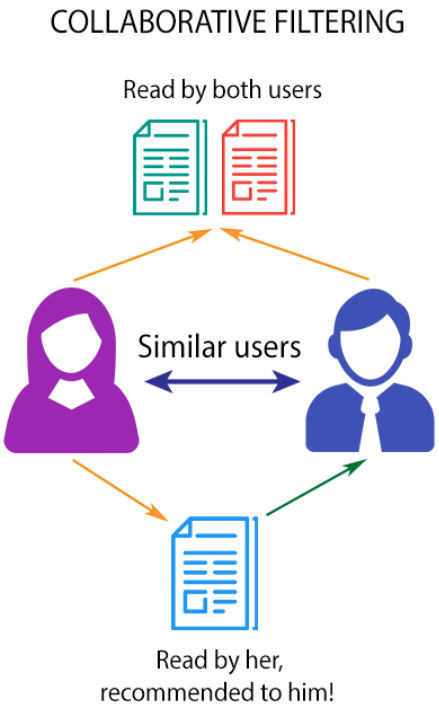
Các bước chính trong việc xây dựng hệ gợi ý dựa trên phương pháp Content-based Filtering:

- Biểu diễn mỗi sản phẩm dưới dạng một vector thuộc tính.
- Gợi ý các sản phẩm tương tự nhau thông qua tính toán độ tương đồng giữa các vector đại diện cho các sản phẩm.
- Hoặc có thể xây dựng hồ sơ của người dùng theo các thuộc tính sản phẩm và gợi ý sản phẩm có thuộc tính phù hợp với hồ sơ người dùng.

Hướng tiếp cận này trong nhiều trường hợp không hiệu quả. Ví dụ như người dùng vừa mua một chiếc điện thoại, người dùng đó có nhu cầu được gợi ý các phụ kiện điện thoại đó, chứ không phải gợi ý chiếc điện thoại tương tự.

2.2.2 Lọc cộng tác (Collaborative Filtering)

Collaborative Filtering [3] hệ thống gợi ý items dựa trên sự tương quan (similarity) giữa các users và/hoặc items. Có thể hiểu rằng ở phương pháp này một item được gợi ý tới một user dựa trên những users có hành vi tương tự. Lọc cộng tác hoạt động dựa trên lịch sử những hành vi của người dùng như: view, click, order,... để tìm quy luật tương tác giữa người dùng và các sản phẩm. Hình 2 mô phỏng cách tiếp cận này. Hệ thống gợi ý dựa trên cách tiếp cận này không quan tâm đến các thuộc tính của các sản phẩm, nó có khả năng khai thác thông tin ngoài phạm vi các thuộc tính về sản phẩm và người dùng. Mô hình huấn luyện có thể xây dựng dựa trên hành vi của một người dùng, hoặc hiệu quả hơn nó có thể khai thác từ nhiều người dùng khác có cùng đặc điểm. Khi làm việc với hành vi của người dùng khác, lọc cộng tác sử dụng việc nhóm để tạo ra đề xuất dựa trên những người dùng tương tự. Về bản chất, nó lọc dựa trên những người dùng có cùng sở thích, hay những người dùng có những hành vi tương tự.



Hình 2: Hướng tiếp cận Collaborative Filtering
(Nguồn `https://www.matrixanalyticscorp.com/recommendation-systems.html`)

Phương pháp này có hai cách tiếp cận nhỏ hơn là: *User-User* và *Item-Item*

User-User: Tư tưởng của cách tiếp cận này là tìm ra các nhóm người dùng có đặc điểm giống nhau. Dự đoán mức độ thích sản phẩm i của một người dùng U dựa vào mức độ thích sản phẩm i của những người cùng nhóm với U. Các bước xây dựng của hướng tiếp cận này như sau:

- Biểu diễn mỗi người dùng bằng một vector dựa trên các sản phẩm đã tương tác.
- Với người dùng U, tính độ tương đồng giữa những người dùng khác đối với U thông qua các vector biểu diễn. Tìm ra một nhóm người dùng "gần" U.
- Ước tính độ phù hợp của người dùng U với các sản phẩm dựa vào lịch sử của nhóm người dùng "gần" với U.
- Có thể chọn k người dùng tương tự với U nhất, hoặc chọn tất cả người dùng nhưng thêm trọng số để ưu tiên những người giống U hơn.

Item-Item: Ý tưởng của hướng tiếp cận này là từ tập các sản phẩm mà người dùng U đã tương tác trước đó, tập sản phẩm này có chung đặc điểm với sản phẩm B. Từ đó dự đoán mức độ mà U sẽ thích sản phẩm B phụ thuộc vào mức độ U thích tập sản phẩm trước đó. Cách xây dựng của hướng tiếp cận này:

- Biểu diễn mỗi sản phẩm bằng một vector những người dùng đã tương tác với nó.
- Tính độ tương đồng giữa các sản phẩm.
- Đối với người dùng U, tìm các sản phẩm tương tự với các sản phẩm U đã tương tác.
- Gợi ý cho U các sản phẩm nói trên, có thể xếp hạng dựa trên tiêu chí như trọng số cao, nhiều người dùng tương tác.

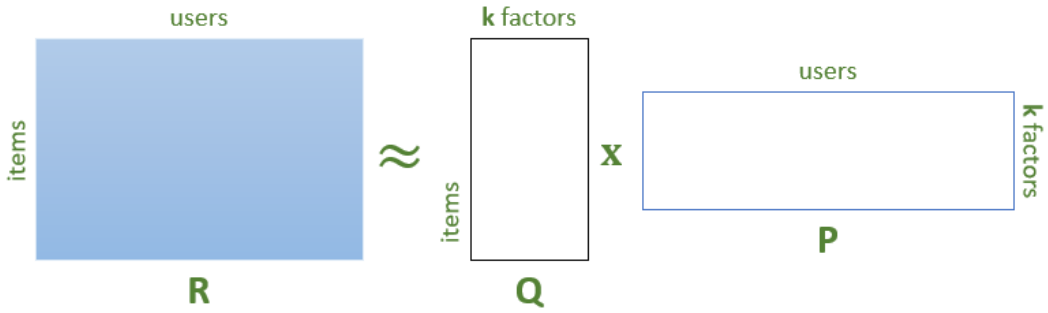
Hướng giải quyết lọc cộng tác này khá tốt nhưng vẫn có nhược điểm là không thể gợi ý những item mới, chưa hề có trong lịch sử của bất cứ người dùng nào. Hiện tượng *Cold Start* trong các hệ gợi ý.

2.2.3 Phân rã ma trận (Matrix factorization)

Là một kỹ thuật trong hướng tiếp cận lọc cộng tác của bài toán gợi ý. Ý tưởng chính đằng sau Matrix Factorization **(MF)** [9] cho hệ gợi ý là tồn tại các thuộc tính ẩn (*latent features*) mô tả sự liên quan giữa các *items* và *users*. Ví dụ với hệ thống gợi ý các bộ phim, thuộc tính ẩn có thể là hình sự, chính trị, hành động, hài,...; cũng có thể là một sự kết hợp nào đó của các thể loại này; hoặc cũng có thể là bất cứ điều gì mà không thực sự cần đặt tên.

Mỗi item mang thuộc tính ẩn ở một mức độ nào đó tương ứng với các hệ số trong vector biểu diễn cho item đó gọi là vector $\mathbf{i}$. Hệ số càng cao tương ứng với việc mang tính chất đó càng cao. Tương tự, mỗi user cũng sẽ có những xu hướng

thích những thuộc tính ẩn nào đó và được mô tả bởi các hệ số trong vector biểu diễn cho user đó gọi là vector **u**. Hệ số càng cao tương ứng với việc user thích các bộ phim có thuộc tính ẩn đó. Giá trị của biểu thức **iu** sẽ cao nếu các thành phần tương ứng của **i** và **u** đều cao. Điều này nghĩa là item mang thuộc tính ẩn mà user thích, vậy nên gợi ý item này cho user đó.



Hình 3: Kĩ thuật phân tích ma trận

Trong phương pháp này, ta cố gắng xấp xỉ ma trận $\mathbf{R} \in \mathbb{R}^{M \times N}$ (Ma trận xếp hạng của M items và N users) bằng tích của hai ma trận $\mathbf{Q} \in \mathbb{R}^{M \times K}$ và $\mathbf{P} \in \mathbb{R}^{K \times N}$

Hàm lỗi: Việc xây dựng hàm lỗi là được dựa trên những thành phần đã được điền trong ma trận **R** là ma trận lịch sử tương tác của người dùng và sản phẩm.

$$\mathcal{L}(\mathbf{Q}, \mathbf{P}) = \frac{1}{2s} \sum_{u=1}^{N} \sum_{i:r_{iu}=1} (r_{iu} - \mathbf{q}_i \mathbf{p}_u)^2 + \frac{\lambda}{2} (\|\mathbf{Q}\|_F^2 + \|\mathbf{P}\|_F^2) \tag{1}$$

trong đó $r_{iu} = 1$ nếu *item* thứ i đã được đánh giá bởi *user* thứ u, còn $\|\cdot\|_F^2$ là *Frobineous norm*, tức là căn bậc hai của tổng bình phương tất cả các phần tử của ma trận (giống với *norm 2* trong vector), **s** là toàn bộ ratings đã có. Thành phần thứ nhất chính là trung bình sai số của mô hình. Thành phần thứ hai trong hàm lỗi là l_2 *regularization* giúp tránh vấn đề *overfiting*.

Việc tối ưu đồng thời $\mathbf{Q}, \mathbf{P}$ tương đối phức tạp, thay vào đó phương pháp được sử dụng là *Coordinate Ascent*, lần lượt tối ưu một ma trận trong khi cố định ma trận kia, tới khi hội tụ.

Tối ưu hàm lỗi:

Khi cố định **Q**, việc tối ưu **P** được đưa về tối ưu hàm:

$$\mathcal{L}(\mathbf{P}) = \frac{1}{2s} \sum_{u=1}^{N} \sum_{i:r_{iu}=1} (r_{iu} - \mathbf{q}_i \mathbf{p}_u)^2 + \frac{\lambda}{2} \|\mathbf{P}\|_F^2 \tag{2}$$

Khi cố định $\mathbf{P}$, việc tối ưu $\mathbf{Q}$ được đưa về tối ưu hàm:

$$\mathcal{L}(\mathbf{Q}) = \frac{1}{2s}\sum_{u=1}^{N}\sum_{i:r_{iu}=1}(r_{iu} - \mathbf{q}_i\mathbf{p}_u)^2 + \frac{\lambda}{2}\|\mathbf{Q}\|_F^2 \tag{3}$$

Hai bài toán này sẽ được tối ưu bằng phương pháp *Gradient Descent*. Công thức (2) có thể được tách thành N bài toán nhỏ, mỗi bài toán ứng với việc đi tối ưu một cột của ma trận $\mathbf{P}$:

$$\mathcal{L}(\mathbf{p}_u) = \frac{1}{2s}\sum_{i:r_{iu}=1}(r_{iu} - \mathbf{q}_i\mathbf{p}_u)^2 + \frac{\lambda}{2}\|\mathbf{p}_u\|_2^2 \tag{4}$$

Vì biểu thức trong dấu $\sum$ chỉ phụ thuộc vào các *items* đã được *rated* bởi *user* đang xét, ta có thể đơn giản nó bằng cách đặt $\hat{\mathbf{Q}}_u$ là ma trận được tạo bởi các hàng tương ứng với các *items* đã được **rated** bởi user đó và $\hat{\mathbf{r}}_u$ là các ratings tương ứng. Khi đó:

$$\mathcal{L}(\mathbf{p}_u) = \frac{1}{2s}\|\hat{\mathbf{r}}_u - \hat{\mathbf{Q}}_u\mathbf{p}_u\|^2 + \frac{\lambda}{2}\|\mathbf{p}_u\|_2^2 \tag{5}$$

và đạo hàm của nó:

$$\frac{\partial\mathcal{L}(\mathbf{p}_u)}{\partial\mathbf{p}_u} = -\frac{1}{s}\hat{\mathbf{Q}}_u^T(\hat{\mathbf{r}}_u - \hat{\mathbf{Q}}_u\mathbf{p}_u) + \lambda\mathbf{p}_u \tag{6}$$

Vậy công thức cập nhật cho mỗi cột của $\mathbf{P}$ là:

$$\mathbf{p}_u = \mathbf{p}_u - \eta(-\frac{1}{s}\hat{\mathbf{Q}}_u^T(\hat{\mathbf{r}}_u - \hat{\mathbf{Q}}_u\mathbf{p}_u) + \lambda\mathbf{p}_u) \tag{7}$$

Tương tự như trên, đặt $\hat{\mathbf{P}}_i$ là ma trận được tạo bởi các cột tương ứng với các *users* đã tương tác **rated** với item đó và $\hat{\mathbf{r}}_i$ là các ratings tương ứng. Khi đó công thức cập nhật cho mỗi hàng của $\mathbf{Q}$ sẽ có dạng:

$$\mathbf{q}_i = \mathbf{q}_i - \eta(-\frac{1}{s}(\hat{\mathbf{r}}_i - \mathbf{q}_i\hat{\mathbf{P}}_i)\hat{\mathbf{P}}_i^T + \lambda\mathbf{q}_i) \tag{8}$$

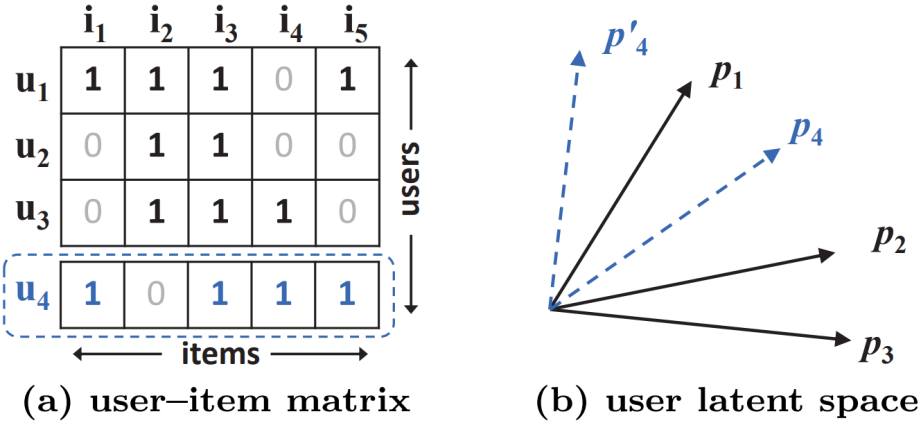
Trong các bài toán thực tế, số lượng *users* và *items* thường rất lớn. Việc tìm ra các mô hình đơn giản giúp dự đoán đánh giá của các users với các items cần được thực hiện một cách nhanh nhất có thể. Với phương pháp MF việc học có thể phức tạp vì phải lặp đi lặp lại việc tối ưu một ma trận khi cố định ma trận còn lại, nhưng việc dự đoán đơn giản vì ta chỉ cần lấy tích của 2 vector **u** và **i**, mà mỗi vector thì có độ dài nhỏ. Vì thế việc dự đoán không yêu cầu khả năng

tính toán cao. Việc này khiến nó phù hợp với các mô hình có tập dữ liệu lớn.

2.3 Lọc cộng tác với mạng nơ-ron

2.3.1 Nhược điểm của kĩ thuật phân rã ma trận (MF)

Phương pháp MF mô hình hóa tương tác hai chiều giữa các thuộc tính ẩn của người dùng và sản phẩm. MF giả sử các chiều trong không gian ẩn đó là độc lập và khi tính ra giá trị tương tác bằng cách tổ hợp các chiều đó lại với cùng một trọng số. Như vậy, MF có thể coi là một mô hình tuyến tính của các thuộc tính ẩn. Xét ví dụ dưới đây:



Hình 4: Dữ liệu mẫu và vector người dùng trên không gian thuộc tính ẩn [6]

Ví dụ sau đây (Hình 4) chứng minh rằng hàm tích trong (*inner product function*) có thể giới hạn chất lượng của MF. Do MF chiếu người dùng và sản phẩm lên cùng một không gian ẩn, độ tương đồng giữa 2 người dùng có thể được đánh giá sử dụng phép nhân vô hướng hoặc có thể sử dụng độ đo Cosine giữa 2 vector ẩn được tạo. Không mất tính tổng quát, ta sử dụng độ đo Jaccard đo tương tự giữa 2 người dùng trong dữ liệu ban đầu mà sau đó MF cần có khả năng nắm bắt được.

Độ đo Jaccard: Đặt $\mathcal{R}_u$ là tập các *item* mà *user* **u** có tương tác, khi đó độ đo tương đồng *Jaccard* của hai user i và j được định nghĩa là:

$$s_{i,j} = \frac{|\mathcal{R}_i| \cap |\mathcal{R}_j|}{|\mathcal{R}_i| \cup |\mathcal{R}_j|} \tag{9}$$

Từ số liệu của ba dòng đầu tiên của hình (a). Ta tính ra được $s_{23}(0.66) > s_{12}(0.5) > s_{13}(0.4)$. Như vậy, quan hệ trên không gian của p_1, p_2, p_3 có thể được

vẽ như trong hình (b). Bây giờ ta xét user u_4. Ta có $s_{41}(0.6) > s_{43}(0.4) > s_{42}(0.2)$, điều này có nghĩa u_4 tương tự nhất với u_1, tiếp theo là u_3 và u_2. Tuy nhiên, do mô hình MF đưa vector người dùng lên cùng 1 không gian ẩn, vậy nên có 2 cách đặt vector của user u_4 dựa trên MF sao cho thỏa mãn gần u_1 nhất như hình vẽ trên (2 đường p'_4 và p_4). Và cả hai trường hợp này đều không thể thỏa mãn được tính chất ngay từ ban đầu, đó là u_4 gần u_3 hơn u_2. Từ đó cho thấy rằng MF không thể mô tả lại được độ đo Jaccard hay chính là độ đo sự tương tự giữa các người dùng trong trường hợp này.

Ví dụ trên đã chứng tỏ được MF có một số giới hạn nhất định khi cố định sử dụng một hàm tích trong (inner product) đơn giản để dự đoán cho một tương tác phức tạp giữa người dùng và sản phẩm. Một trong những cách khắc phục điều này đó là sử dụng hệ số ẩn K lớn hơn, tuy nhiên, việc này sẽ làm mất tính tổng quát của mô hình, hay chính là vấn đề overfitting, đặc biệt trong ma trận thưa.

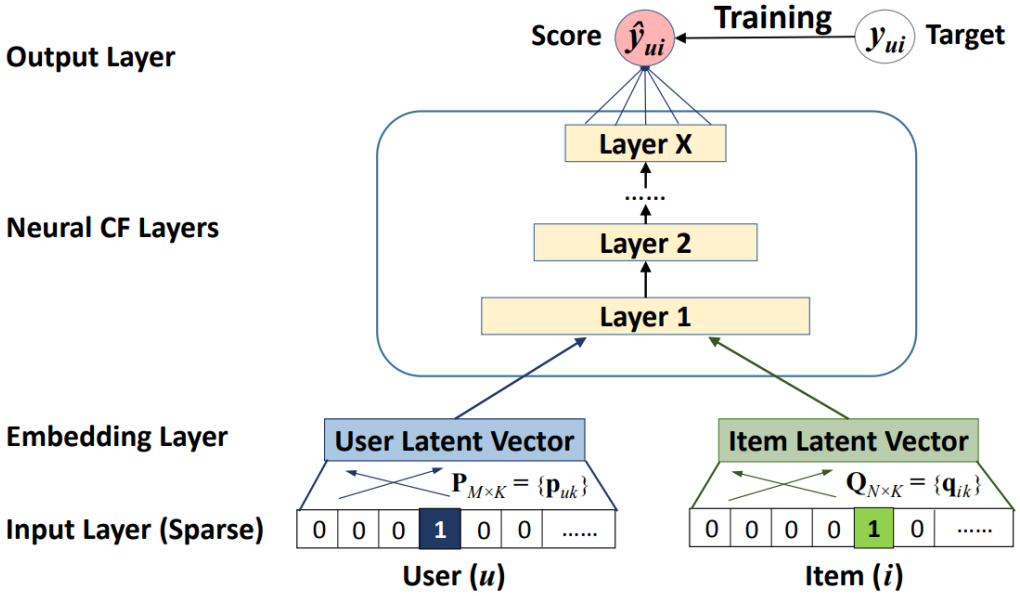
2.3.2 Neural Collaborative Filtering (NCF)

Từ những nhược điểm của kĩ thuật MF có thể chỉ ra được việc cần phải thiết kế một hàm tương tác tốt hơn để mô hình hóa việc tương tác giữa các thuộc tính ẩn (*latent features*) của *user* và *item*. Phép toán inner product, đơn giản chỉ là việc kết hợp bằng cách nhân các thuộc tính ẩn tuyến tính, có thể không đủ để nắm bắt cấu trúc phức tạp dữ liệu tương tác của người dùng. Do đó mô hình NCF [6] đi khai thác phương pháp MF và thực hiện nó bằng mạng nơ-ron.

Mô hình NCF được nhóm tác giả Xiangnan He và cộng sự đưa ra vào năm 2017. Cấu trúc NCF tổng quát được thể hiện trong Hình 5.

Ta có thể thấy trong Hình 5, đi từ dưới lên mô hình NCF gồm những tầng sau:

- Tầng input: bao gồm các thuộc tính vector của *user* và *item*. Như trong hình trên, các vector đó là biểu diễn dang one-hot (vector với chỉ một trường có giá trị bằng 1, các trường còn lại có giá trị 0) dùng để biểu diễn định danh của *user (userID))* và *item (itemID)*
- Trên tầng input là tầng embedding. Tầng embedding kết nối đầy đủ *fully connected* với tầng input. Qua tầng embedding, biểu diễn one-hot (dạng thưa) của *use* và *item* trở thành vector dầy đặc *(dense vector)*. Biểu diễn này có thể xem là biểu diễn người dùng và vật phẩm bằng các thuộc tính ẩn.



Hình 5: Mô hình NCF tổng quát [6].

- Trên tầng embedding (tầng 1 đến tầng n) là các tầng thuộc mạng nơ-ron mà tác giả gọi là tầng *neural collaborative*. Các tầng này có nhiệm vụ ánh xạ từ các vector embedding đến vector đầu vào của tầng output.
- Tầng trên cùng là tầng output. Tầng này chứa điểm số dự đoán $\hat{y}_{ui}$

Việc huấn luyện NCF được thực hiện bằng cách tối thiểu hóa hàm lỗi giữa điểm số dự đoán $\hat{y}_{ui}$ với giá trị mục tiêu (target) y_{ui} tương ứng.

Điểm số dự đoán được tính như sau:

$$\hat{y}_{ui} = f(\mathbf{P}^T\mathbf{v}_u^U, \mathbf{Q}^T\mathbf{v}_i^I|\mathbf{P}, \mathbf{Q}, \Theta_f) \tag{10}$$

Trong đó:

- $\mathbf{v}_u^U, \mathbf{v}_i^U$ lần lượt là các ma trận hàng (vector) one-hot biểu diễn cho *user* u và *item* i.
- $\mathbf{P} \in \mathbb{R}^{M\times K}$ và $\mathbf{Q} \in \mathbb{R}^{K\times N}$ lần lượt là các ma trận tầng embedding chiếu $\mathbf{v}_u^U, \mathbf{v}_i^U$ vào không gian các thuộc tính ẩn. $\mathbf{P}$ và $\mathbf{Q}$ là ma trận trọng số giữa tầng input và tầng embedding.
- Θ_f là tập các tham số mô hình của hàm f

Do hàm f ở đây là ánh xạ đầu vào đầu ra của một mạng nơ-ron nên ta có thể viết f theo công thức sau:

$$f(\mathbf{P}^T\mathbf{v}_u^U, \mathbf{Q}^T\mathbf{v}_i^I) = \phi_{out}(\phi_X(...\phi_2(\phi_1(\mathbf{P}^T\mathbf{v}_u^U, \mathbf{Q}^T\mathbf{v}_i^I))..)) \tag{11}$$

Ở đây:

- ϕ_{out} là hàm ánh xạ đầu vào tới đầu ra của tầng output.
- $\phi_n, \phi_{n-1}, ..., \phi_2, \phi_1$ lần lượt là các hàm ánh xạ đầu vào tới đầu ra tại các tầng n, n-1,. . . , 2, 1 trong mạng nơ-ron.

Hàm mục tiêu của NCF:

Giả sử trong các tương tác người dùng và sản phẩm, ta kí hiệu $\mathcal{Y}$ là tập các tương tác xảy ra và quan sát được, $\mathcal{Y}^-$ là tập các tương tác không quan sát được hay chính là các tương tác không xảy ra trong thực tế, w_{ui} là hệ số của cặp học *(u,i)*. Ta có hàm mục tiêu là hàm độ lệch bình phương giữa điểm dự đoán và điểm thực tế:

$$L_{sqr} = \sum_{(u,i)\in\mathcal{Y}\cup\mathcal{Y}^-} w_{ui}(y_{ui} - \hat{y}_{ui})^2 \tag{12}$$

Hàm mục tiêu là hàm cực tiểu bình phương sẽ thường phù hợp cho các bài toán mà tập dữ liệu có phân phôi gần như phân phối *Gauss*, mà loại phân phối này thì thường không phù hợp với dữ liệu sinh ra từ tương tác người dùng sản phẩm. Lý do bởi tương tác người dùng có giá trị 1 khi xảy ra tương tác, 0 nếu không có bất cứ tương tác nào giữa người dùng và sản phẩm. Giá trị của hàm chỉ có thế là 0 hoặc 1. Chính vì sự khác biệt này, ta cần một độ đo khác đặc biệt và phù hợp cho loại dữ liệu như trên.

Ta đưa đầu ra phải nằm trong khoảng $(0, 1)$ nhờ hàm activate của lớp cuối cùng. Giá trị đầu ra thể hiện độ liên quan giữa người dùng và sản phẩm. Điểm càng cao thì độ liên quan càng cao. Để học tham số của mô hình, ta sử dụng hàm *likelihood* có công thức như sau:

$$p(\mathcal{Y}, \mathcal{Y}^-|\mathbf{P}, \mathbf{Q}, \Theta_f) = \prod_{(u,i)\in\mathcal{Y}} \hat{y}_{ui} \prod_{(u,j)\in\mathcal{Y}^-} (1 - \hat{y}_{uj}). \tag{13}$$

Lấy *logarithm* của hàm và đổi dấu, ta có:

$$\begin{aligned} L &= -\sum_{(u,i)\in\mathcal{Y}} \log \hat{y}_{ui} - \sum_{(u,j)\in\mathcal{Y}^-} \log(1 - \hat{y}_{uj}) \\ &= -\sum_{(u,i)\in\mathcal{Y}\cup\mathcal{Y}^-} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{uj}) \end{aligned} \tag{14}$$

Đây chính là hàm mục tiêu của các mô hình xây dựng trên mô hình NCF, có thể được tối ưu dựa vào các phương pháp tối ưu truyền thống như *SGD* hay *Adam*. Có thể thấy nó tương tự như hàm mục tiêu *binary cross-entropy loss* hay

còn được gọi là *log-loss*. Ta đã đưa bài toán gợi ý về bài toán phân loại nhị phân. Với các dữ liệu thuộc tập $\mathcal{Y}^-$, ta chọn ngẫu nhiên theo phân phối chuẩn từ các sản phẩm mà người dùng chưa tương tác và đưa vào tập dữ liệu học với một tỉ lệ nhất định.

Chương 3 Một số mô hình sử dụng phản hồi ẩn

Chương này mô tả ý tưởng xây dựng các mô hình VALS, NeuMF và COMF. Mô hình VALS với mong muốn cải thiện hiệu quả hệ gợi ý thông qua dữ liệu xem của người dùng; mô hình NeuMF xử lý implicit feedback bằng mạng nơ-ron; Còn mô hình COMF có khả năng kết hợp hai loại dữ liệu implicit và explicit feedback. Qua đó phân tích điểm mạnh và những hạn chế của từng mô hình.

3.1 View-enhanced eALS (VALS)

Mô hình View-enhanced eALS (VALS) [2] được tác giả Jingtao Ding và nhóm nghiên cứu đề xuất trong bài báo *Improving implicit recommender systems with view data* năm 2018. Mô hình với mục đích cải thiện hệ thống gợi ý bởi tích hợp dữ liệu xem (view). View-enhanced eALS mô hình hóa tương tác các cặp sự kiện giữa purchased (đã mua), viewed (đã xem) và non-viewed (chưa xem). Loại dữ liệu xem này cung cấp tín hiệu có giá trị về sở thích của người dùng, có thể bổ sung cho dữ liệu mua hàng trong hai mặt. Đầu tiên, nếu người dùng xem một mặt hàng, bất kể có mua hay không, ít nhất nó cũng phản ánh rằng người dùng quan tâm đến mặt hàng đó (tức là tín hiệu tích cực, so với các mặt hàng không được xem). Thứ hai, nếu người dùng xem một sản phẩm nhưng không mua nó sau đó, điều đó có nghĩa là mặt hàng đó ít được người dùng quan tâm (nghĩa là tín hiệu tiêu cực, so với các mặt hàng đã mua). Như vậy, dữ liệu xem có thể được xem là một phản hồi trung gian giữa mua và chưa xem, làm phong phú hơn mức độ của phản hồi ẩn để phân biệt rõ hơn sở thích của người dùng.

Để tìm hiểu mô hình VALS trước hết cần một số kí kiệu cơ bản sau. Ma trận tương tác giữa user-item $\mathbf{R} \in \mathbb{R}^{M \times N}$, M và N kí hiệu số lượng users và items, tương ứng $\mathcal{R}$ kí hiệu tập các cặp user-item có tương tác. Với user u, vector $\mathbf{p}_u \in \mathbb{R}^K$ kí hiệu vector thuộc tính ẩn (*latent feature vector*) K-chiều.. Tập $\mathcal{R}_u$ kí hiệu tập của những item được tương tác bởi u. Tương ứng, với item i kí hiệu $\mathbf{q}_i$ và $\mathcal{R}_i$ được sử dụng. Ma trận $\mathbf{P} \in \mathbb{R}^{M \times K}$ và $\mathbf{Q} \in \mathbb{R}^{N \times K}$ kí hiệu ma trận nhân tố ẩn (*latent factor matrix*) cho các users và items. Ở mô hình phân rã ma trận (MF), mỗi phần tử r_{ui} của $\mathbf{R}$ được ước lượng bởi $\hat{r}_{ui} = \mathbf{p}_u^T \mathbf{q}_i$ ($\mathbf{R} \approx \mathbf{P} \times \mathbf{Q}^T$).

Để học ra nhân tố ẩn của user/item, phương pháp eALS [7] trước đó đưa ra hàm lỗi sau, nó gán $r_{ui} = 0$ cho những sản phẩm chưa xem bởi u:

$$\mathbf{J} = \sum_{(u,i) \in \mathcal{R}} \omega_{ui} (\hat{r}_{ui} - r_{ui})^2 + \sum_{u=1}^{M} \sum_{i \notin \mathcal{R}_u} s_j \hat{r}_{ui}^2 + \lambda (\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) \tag{15}$$

ở đây ω_{ui} kí hiệu trọng số của cặp $(u, i) \in \mathcal{R}$. Được xác định bởi một chiến lược trọng số nhận biết phổ biến, s_i biểu thị rằng mặt hàng i mà u chưa tương tác được đánh giá tiêu cực (*negative*).

Dựa trên eALS, phương pháp View-enhanced eALS (VALS) ở đây xem xét làm thế nào để dự đoán hiệu quả sở thích của user từ cả dữ liệu mua (purchase data) và xem (view data). Phương pháp này sử dụng hai khoảng cách lề trung gian để mô hình hóa mức độ thích của user đối với tương tác viewed, nó sẽ thấp hơn tương tác purchased nhưng lại cao hơn non-viewed. Để thuận tiện, ta cần bảng tổng hợp một số kí hiệu trong bảng 1.

Kí hiệu	Mô tả
M, N, K	Số lượng users, items, factors
$\mathbf{P}, \mathbf{p}_u$	Ma trận nhân tố ẩn và vector của user
$\mathbf{Q}, \mathbf{q}_i$	Ma trận nhân tố ẩn và vector của item
$\mathcal{R}, \mathcal{R}_u, \mathcal{R}_i$	Tập các cặp (u, i) có tương tác mua (purchased), những item mua bởi u, những user đã mua i
$\mathcal{V}, \mathcal{V}_u, \mathcal{V}_i$	Kí hiệu tương tự cho tương tác xem (view)
$\mathcal{RV}, \mathcal{RV}_u, \mathcal{RV}_i$	Kí hiệu tương tự cho hợp, $\mathcal{R} \cup \mathcal{V}$
$\hat{r}_{ui}, \hat{r}_{uv}, \hat{r}_{uj}$	Dự đoán của user u về mua item v, xem item i và chưa xem item j
ω_{ui}	Trọng số của tương tác mua của cặp (u, i)
s_j	Trọng số của item j trong tập dữ liệu non-view
c_v	Trọng số của item v trong tập dữ liệu view
γ_1, γ_2	Giá trị lề giữa $\hat{r}_{ui}$, $\hat{r}_{uv}$ và $\hat{r}_{uj}$
λ	Tham số Regularization

Bảng 1: Danh sách kí hiệu dùng cho mô hình VALS

Trong thương mại điện tử, khi user u xem item i, ta có thể cho giá trị dự đoán $\hat{r}_{uv}$ ở khoảng giữa những item j ($\hat{r}_{uj}$) chưa xem và những item i ($\hat{r}_{ui}$) đã mua. Tuy nhiên khó khăn là chọn giá trị $\hat{r}_{uv}$ cho những user khác nhau, để xử lý vấn

đề này hàm mục tiêu của phương pháp VALS được thiết kế là:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{eALS} + \mathcal{L}_{Reg} + \mathcal{L}_{view} \\ &= \sum_{(u,i)\in\mathcal{R}} \omega_{ui}(\hat{r}_{ui} - r_{ui})^2 + \sum_{u=1}^{M}\sum_{j\notin\mathcal{RV}_u} s_j\hat{r}_{ui}^2 + \lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) + \\ &\sum_{(u,v)\in\mathcal{V}} c_v[\sum_{i\in\mathcal{R}_u}(\gamma_1 - (\hat{r}_{ui} - \hat{r}_{uv}))^2 + \sum_{j\notin\mathcal{RV}_u}(\gamma_2 - (\hat{r}_{uv} - \hat{r}_{uj}))^2]\end{aligned} \tag{16}$$

Để ý thấy rằng hàm mục tiêu có thể chia thành ba thành phần, hai thành phần đầu tiên đại diện cho lỗi phán đoán và regularization, còn thành phần cuối L_{view} mô tả mức độ trung gian giữa tín hiệu xem. Cụ thể, dự đoán $\hat{r}_{uv}$ được tối ưu hóa để nhỏ hơn $\hat{r}_{ui}$ với mức lề là γ_1, ở thành phần $(\gamma_1 - (\hat{r}_{ui} - \hat{r}_{uv}))^2$. Tương tự, $\hat{r}_{uv}$ được tối ưu để cao hơn $\hat{r}_{uj}$ bằng một lề khác là γ_2. Bằng cách thay đổi cặp giá trị (γ_1, γ_2), ta có khả năng điều chỉnh phạm vi của điểm số dự đoán trên những item đã xem, từ đó tìm kiếm mức độ thích hợp nhất cho tín hiệu xem.

Điểm mạnh của mô hình:

- Mô hình hóa được giả thuyết về sở thích của một người dùng về những sản phẩm đã xem thấp hơn những sản phẩm đã mua nhưng lại cao hơn sản phẩm chưa xem.
- Sử dụng được dữ liệu xem vào hệ gợi ý, nhiều mô hình khác dựa trên phân rã ma trận không tận dụng được dữ liệu này.

Những hạn chế của mô hình:

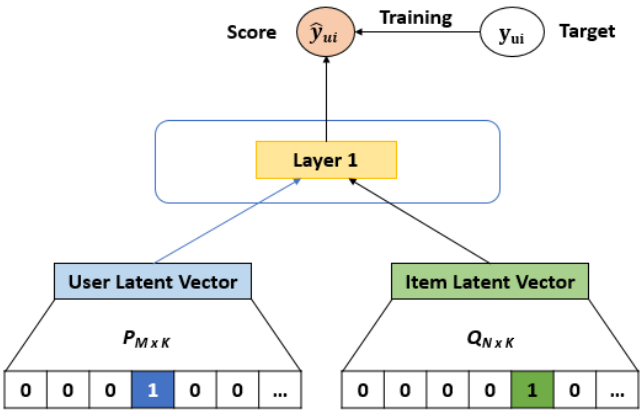
- Mô hình có khá nhiều tham số, việc này sẽ gây khó khăn trong quá trình lựa chọn tham số.
- Đối với những sản phẩm mà người dùng chưa xem, mô hình coi như người dùng có mức độ thích những sản phẩm đó rất thấp. Nhưng thực tế, có nhiều sản phẩm người dùng thích nhưng chưa xem nó vì chưa biết tới những sản phẩm đó.

3.2 Neural Matrix Factorization (NeuMF)

Trong mục này mô tả các mô hình mạng nơ-ron xây dựng trên cơ sở mô hình NCF sử dụng cho hệ gợi ý. Phân tích cách xây dựng mô hình NeuMF của tác giả Xiangnan He và cộng sự. Để tiếp cận được mô hình NeuMF trước tiên ta cần tìm hiểu hai mô hình GMF và MLP.

3.2.1 Generalized Matrix Factorization (GMF)

Từ mô hình NCF, ta có thể xây dựng được mô hình tổng quát hóa cho phương pháp phân rã ma trận (GMF). Sử dụng mô hình NCF với một layer trong mạng nơ-ron, ta được mô hình như sau:



Hình 6: Mô hình GMF.

Đặt vector ẩn của user $\mathbf{p}_u = \mathbf{P}^T \mathbf{v}_u^U$ và vector ẩn của item $\mathbf{q}_i = \mathbf{Q}^T \mathbf{v}_i^I$. Ta định nghĩa hàm ánh xạ của tầng thứ nhất là:

$$\phi(\mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u \odot \mathbf{q}_i, \tag{17}$$

Ở đây $\odot$ kí hiệu phép nhân element-wise của vector. Khi đó ta chiếu vector tới tầng output:

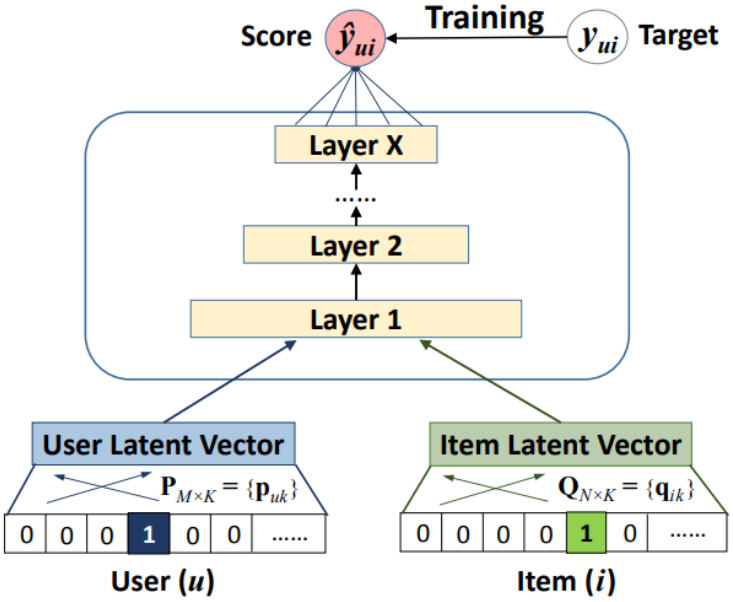
$$\hat{y}_{ui} = a_{out}(\mathbf{h}^T(\mathbf{p}_u \odot \mathbf{q}_i) \tag{18}$$

Ở đây a_{out} và $\mathbf{h}$ kí hiệu hàm kích hoạt *(activation function)* và trọng số cạnh *(edge weight)* của tầng output. Một cách trực quan, nếu ta sử dụng hàm đồng nhất *(identity function)* cho a_{out} và đưa $\mathbf{h}$ là vector tất cả các phần tử bằng 1, ta đạt được mô hình MF.

Với mô hình mạng theo kiểu NCF như trên, ta đã có thể mở rộng so với phương pháp MF truyền thống. Vi dụ ta có thể bỏ đi quy ước h không nhất thiết là 1 vector có các trọng số giữa các chiều như nhau. Khi ấy, mạng nơ-ron sẽ có thể học ra các nhân tố ẩn quan trọng trong vector nhân tố ẩn, một khả năng mà MF không thể làm được. Hoặc ta có thể sử dụng một hàm phi tuyến tính a_{out} giúp mô hình tương thích với các dữ liệu phi tuyến tốt hơn mô hình MF truyền thống. Mô hình này được gọi là mô hình tổng quát hóa phân tích ma trận - GMF.

3.2.2 Multi-Layer Perceptron (MLP)

Sau khi xây dựng GMF, ta nhận thấy rằng GMF có khả năng tổng quát hóa phương pháp MF. Tuy nhiên, GMF lại sử dụng hàm nhân từng phần tử $\odot$ (elementwise) để mô tả tương tác giữa người dùng và sản phẩm, điều này sẽ chỉ mô tả được các tương tác dựa trên mô hình tuyến tính. Dễ nhận thấy rất nhiều cách kết hợp 2 vector ẩn với nhau. Phương pháp MLP đề xuất cách nối hai vector thành một vector duy nhất, sau đó sử dụng thêm nhiều tầng ẩn tiếp theo để có được hàm mới mô tả tương tác giữa người dùng và sản phẩm. Việc này sẽ giúp hàm tương tác được đa dạng hóa hơn, có thể nắm bắt được tính chất phi tuyến của dữ liệu.



Hình 7: Mô hình MLP.

Xét mô hình NCF với cấu trúc như Hình 7 với L layer trong mạng nơ-ron. Để áp dụng MLP *(Multi-Layer Perceptron)*, ta cài đặt mô hình như sau:

$$
\begin{aligned}
\mathbf{z_1} &= \phi_1(\mathbf{p}_u, \mathbf{q}_i) = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix} \\
\mathbf{z_2} &= \phi_2(\mathbf{z_1}) = a_2(\mathbf{z_1}\mathbf{W_2} + \mathbf{b_2}) \\
&\dots \\
\mathbf{z_L} &= \phi_L(\mathbf{z_{L-1}}) = a_L(\mathbf{z_{L-1}}\mathbf{W_L} + \mathbf{b_L}) \\
\hat{y}_{u,i} &= a_{out}(\mathbf{z_L} = \sigma(\mathbf{z_L})
\end{aligned}
\tag{19}
$$

Trong đó:

- $\mathbf{W_x}$: Ma trận trọng số giữa tầng x-1 và tầng x. Giả sử tầng x-1 có m nơ-ron, tầng x có n nơ-ron thì cỡ của ma trận trọng số là (m x n).
- $\mathbf{b_x}$: vector hàng biểu diễn độ lệch *(bias)* của dữ liệu tại tầng x.
- a_x: hàm kích hoạt tại layer x của mạng nơ-ron.

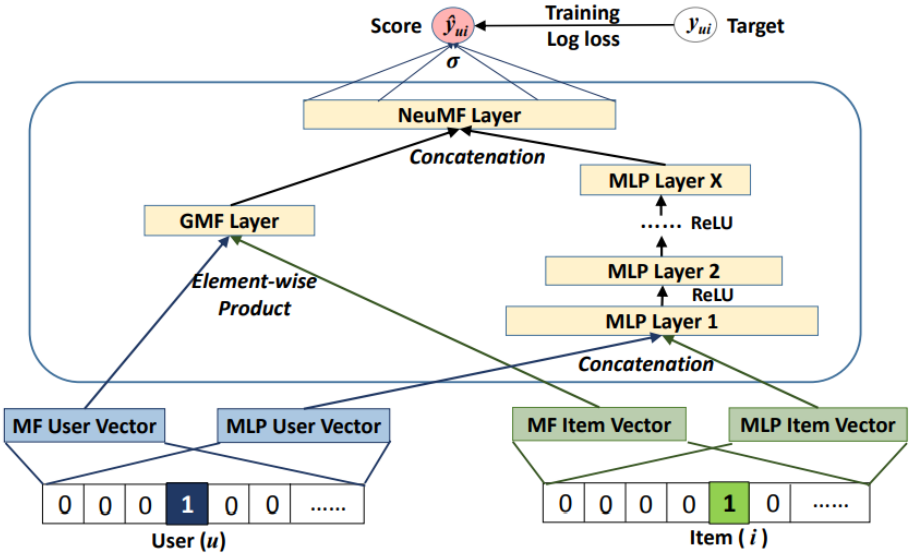
Đối với hàm tác động của MLP layers, ta có thể sử dụng những hàm như sigmoid, hyperbolic tangent (tank), và Rectifier (ReLU),... Tuy nhiên có một số đặc điểm của các hàm này như sau:

- Hàm **sigmoid** đưa giá trị nhận được của perceptron về khoảng (0, 1), thích hợp cho các bài toán học mà đầu ra nằm trong khoảng (0, 1). Tuy nhiên, hàm sigmoid bị hiện tượng "bão hòa" (*saturation*) khi giá trị đầu vào gần 0 hoặc 1. Khi đó gradient quá nhỏ và mô hình không "học" được gì từ dữ liệu đó.
- Hàm **tanh** có độ dốc xung quanh tâm đồ thị lớn hơn hàm sigmoid, điều này cho phép quá trình học diễn ra nhanh hơn và hiệu quả hơn. Hàm tanh chỉ khắc phục được một phần nhỏ vấn đề saturation của hàm sigmoid cơ bản.
$$tanh(x) = 2\sigma(2x) - 1$$
- Hàm **ReLU** chỉ chứa các phép tính cơ bản và đơn giản (không chứa phép lũy thừa như hàm sigmoid và tanh), nên chi phí tính toán thấp. Hàm ReLU hiệu quả hơn hàm sigmoid và hàm tanh vì giải quyết được vấn đề saturation
$$ReLU(x) = max(0, x)$$

Trong đồ án hàm ReLU được dùng làm kích hoạt tại các tầng ẩn của mạng nơ-ron.

3.2.3 Neural Matrix Factorization model (NeuMF)

Để kết hợp phân tích ma trận tổng quát (GMF) và perceptron đa tầng (MLP), tác giả Xiangnan He và cộng sự giới thiệu mô hình NeuMF [6] kết hợp GMF và MLP với thiết kế như sau:



Hình 8: Mô hình NeuMF (Neural Matrix Factorization model).

Công thức tính toán cho mô hình:

$$\begin{aligned}
\phi^{GMF} &= \mathbf{p}_u^G \odot \mathbf{q}_i^G, \\
\phi^{MLP} &= a_L(\mathbf{W}_L^T(a_{L-1}(...a_2(\mathbf{W}_2^T \begin{bmatrix} \mathbf{p}_u^M \\ \mathbf{q}_i^M \end{bmatrix} + \mathbf{b}_2)...)) + \mathbf{b}_L), \\
\hat{y}_{ui} &= \sigma(\mathbf{h}^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix},
\end{aligned} \tag{20}$$

Ở đây:

- $\mathbf{p}_u^G, \mathbf{p}_u^M$ lần lượt là vector các nhân tố ẩn biểu diễn người dùng u từ GMF và MLP.
- $\mathbf{q}_i^G, \mathbf{q}_i^M$ lần lượt là vector các nhân tố ẩn biểu diễn sản phẩm i từ GMF và MLP.
- ϕ^{GMF} là vector đầu ra của mô hình GMF.
- ϕ^{MLP} là vector đầu ra của mô hình MLP.
- $\mathbf{h}$ là trọng số cạnh cho vector đầu ra của mô hình

Hàm ReLU là hàm tác động cho MLP. Mô hình này kết hợp tính chất tuyến tính của MF và tính chất phi tuyến của mạng nơ-ron đa tầng để mô hình hóa tương tác của người dùng với sản phẩm. ϕ là hàm tác động của output, ta sử

dụng hàm sigmoid để đưa đầu ra vào khoảng (0, 1).

Điểm mạnh của mô hình:

- Kết hợp hai phương pháp mô hình hóa sự tương đồng giữa user với item là GMF và MLP.
- Ứng dụng mạng nơ-ron học sâu vào bài toán gợi ý. Tổng quát hóa kĩ thuật phân rã ma trận trong mô hình mạng nơ-ron. Từ đó sử dụng được tính chất phi tuyến để mô hình hóa tương tác giữa người dùng và sản phẩm.

Những hạn chế của mô hình:

- NeuMF nhận dữ liệu đầu vào chỉ quan tâm tới user đã tương tác với item chưa. Hay nói cách khác, NeuMF xử lý implicit feedback và explicit feedback theo cùng một phương thức. Như vậy chưa tận dụng được giá trị của explicit feedback và dễ bị ảnh hưởng từ nhiễu của implicit feedback.
- Cũng giống như các phương pháp dựa trên phân rã ma trận khác, NeuMF chưa xử lý được vấn đề Cold Start.

3.3 Co-rating Model (COMF)

Mô hình Co-rating [11] được tác giả Nathan N. Liu và cộng sự đề xuất năm 2010. Mô hình được mô tả như sau.

Đặt $\mathbf{U} = \{u_1, u_2, ..., u_m\}$ là tập m người dùng và $\mathbf{I} = \{q_1, q_2, ..., q_n\}$ biểu thị tập n sản phẩm. Cả phản hồi tường minh và phản hồi ẩn là những tương tác có thể quan sát được giữa $\mathbf{U}$ và $\mathbf{I}$ phản ánh tương tác của người dùng đối với sản phẩm. Phản hồi tường minh thường ở ratings dạng số, độ lớn cho biết số lượng người dùng thích một sản phẩm cụ thể. Kí hiệu $\mathbf{X} \in \mathbb{R}^{m \times n}$ là ma trận các phản hồi tường minh quan sát được. Các phần tử $x_{ui} \in \mathbb{R}$ biểu thị xếp hạng của người dùng u về sản phẩm i và $\mathbf{S}^* \subseteq \mathbf{P} \times \mathbf{Q}$ biểu thị cho tập các cặp user-item có phản hồi tường mimh. Các loại phản hồi ẩn bao gồm các hành động khác nhau mà người dùng thực hiện trên các sản phẩm được hệ thống tự động theo dõi. Ví dụ phản hồi ẩn như click vào sản phẩm, tin tức,... Trong trường hợp tổng quát phản hồi ẩn được biểu diễn bởi $y_{ui} \in \{-1, +1\}$ tương ứng với nếu người dùng u đã tương tác với sản phẩm i thì $y_{ui} = +1$, ngược lại $y_{ui} = -1$. Kí hiệu $\mathbf{Y} \in (-1, +1)^{m \times n}$ biểu thị ma trận phản hồi ẩn quan sát được và sử dụng $\mathbf{S}^+ \subseteq \mathbf{U} \times \mathbf{I}$ và $\mathbf{S}^- \subseteq \mathbf{U} \times \mathbf{I}$ kí hiệu cho tập các cặp user-item mà y_{ui} tương ứng bằng +1 và -1.

Không giống như phản hồi tường minh, y_{ui} là các chỉ số yếu và mơ hồ về sự

yêu thích của người dùng. Ví dụ: người dùng có thể không thực sự thích một bài hát hoặc một tin tức mà anh ta chỉ vô tình nghe hoặc đọc. Việc nghe hay đọc nó có thể là biểu hiện anh ta thực sự thích một bài hát hay một tin tức hoặc có thể do anh ta không nghe hay đọc vì anh ta chưa biết những sản phẩm đó. Nếu không có thêm thông tin, sẽ rất khó khăn để phân biệt những sự mơ hồ như thế trong các phản hồi ẩn và hầu hết các nghiên cứu hiện tại dựa trên giả định rằng các sản phẩm mà $y_{ui} = +1$ phù hợp hơn các sản phẩm mà $y_{ui} = -1$.

Giữa Implicit và Explicit feedback có hai điểm khác nhau quan trọng là:

- Với phản hồi tường minh, x_{ui} chỉ được biết đến với những $(u, i) \in \mathbf{S}^*$, trong khi phản hồi ẩn giá trị của y_{ui} nằm trong tất cả $(u, i) \in \mathbf{U} \times \mathbf{I}$.
- phản hồi tường minh là một số cụ thể, thể hiện bản chất và độ lớn của xếp hạng x_{ui} tương ứng với sở thích của người dùng trong khi phản hồi ẩn, các giá y_{ui} cần được phân loại.

Để thuận tiện, ta kí hiệu I_u^* để chỉ tập item i mà người dùng u có phản hồi tường minh ($(u, i) \in \mathbf{S}^*$). Tương tự, ta định nghĩa I_u^+ và I_u^- lần lượt là tập sản phẩm mà $y_{ui} = +1$ và -1. Đối xứng với điều này, ta định nghĩa tập người dùng U_i^*, U_i^+ và U_i^- có liên kết với sản phẩm i.

Co-Rating Model:

Với mô hình phân rã ma trận, ma trận $\mathbf{X} \in \mathbb{R}^{m \times n}$ (m người dùng và n sản phẩm) được xấp xỉ bởi tích hai ma trận $\mathbf{P} \in \mathbb{R}^{k \times m}$ và $\mathbf{Q} \in \mathbb{R}^{k \times n}$:

$$\mathbf{X} \approx \mathbf{P}^T \times \mathbf{Q} \tag{21}$$

Trong mô hình, mỗi người dùng u được mô hình hóa bởi một vector ẩn $\mathbf{p}_u$, nơi cột thứ u của $\mathbf{P}$ và tương tự mỗi sản phẩm i đại diện bởi vector $\mathbf{q}_i$ là cột thứ i của Q. Công thức dự đoán cho $\hat{r}_{ui}$ có dạng sau:

$$\hat{r}_{ui} = \mathbf{p}_u^T . \mathbf{q}_i = \sum_{f=1}^{k} \mathbf{q}_{fi} \mathbf{p}_{fu} \tag{22}$$

Thông thường, các ma trận P và Q được tối ưu thông qua giải bài toán tối ưu:

$$argmin_{\mathbf{P},\mathbf{Q}} = \mathcal{L}(\mathbf{P}, \mathbf{Q}) + \lambda \mathcal{R}(\mathbf{P}, \mathbf{Q}) \tag{23}$$

Hàm lỗi $\mathcal{L}(\mathbf{P}, \mathbf{Q})$ định lượng việc phải xấp xỉ như thế nào, thành phần chuẩn hóa $\mathcal{R}(\mathbf{P}, \mathbf{Q})$ để tránh vấn đề overfitting và tham số λ điều khiển mức độ quan

trọng giữa hai thành phần trên.

Đối với phản hồi tường minh, ta mô hình hóa gần đúng các xếp hạng quan sát được trong $\mathbf{S}^*$. Điều này đưa ra việc tính hàm lỗi cho phản hồi tường minh:

$$\mathcal{L}_E(\mathbf{P}, \mathbf{Q}) = \sum_{(u,i)\in\mathbf{S}^*} (x_{ui} - \mathbf{p}_u^T.\mathbf{q}_i)^2 \tag{24}$$

Tương tự lỗi bình phương của phản hồi ẩn được đưa ra cho sự xấp xỉ của hai loại phản hồi -1 và +1:

$$\mathcal{L}_I(\mathbf{P}, \mathbf{Q}) = \sum_{u=1}^{m}\sum_{i=1}^{n}(y_{ui} - \mathbf{p}_u^T.\mathbf{q}_i)^2 \tag{25}$$

Thành phần regularization là một Frobenius norm được sử dụng bởi nhiều phương pháp phân rã ma trận lọc cộng tác:

$$\mathcal{R}(\mathbf{P}, \mathbf{Q}) = \|\mathbf{P}\|_F^2 + \|\mathbf{P}\|_F^2 \tag{26}$$

Khi cả phản hồi tường minh và ẩn có sẵn cùng với một nhóm người dùng và sản phẩm, ta muốn điểm số cho dự đoán $\hat{r}_{ui}$ phản ánh tùy chọn của người dùng thể hiện trong cả phản hồi tường minh và ẩn. Tuy nhiên, có hai điều xảy ra. Thứ nhất, thang số của x_{ui} và y_{ui} có thể rất khác nhau. Ví dụ, x_{ui} có thể nằm trong khoảng từ 1 đến 10 trong khi y_{ui} chỉ nhận hai giá trị -1 và +1. Thứ hai, độ chính xác của phản hồi tường minh và ẩn rất khác nhau, với phản hồi ẩn độ chính xác ít chính xác hơn nhiều so với phản hồi tường minh. Để giải quyết hai vấn đề này, ta chuẩn hóa giá trị x_{ui} và y_{ui} ban đầu thành thang điểm chung từ 0 đến 1 và chỉ định tầm quan trọng khác nhau đối với phản hồi tường minh và ẩn. Đó là ý tưởng của mô hình **Co-rating (COMF)** và hàm mục tiêu như sau:

$$agrmin_{\mathbf{P},\mathbf{Q}}(\mathcal{L}_E(\mathbf{P}, \mathbf{Q}) + \eta\mathcal{L}_I(\mathbf{P}, \mathbf{Q}) + \lambda\mathcal{R}(\mathbf{P}, \mathbf{Q})) \tag{27}$$

trong đó lỗi của phản hồi tường minh và ẩn được cộng vào với nhau và bổ sung tham số η để cân nhắc tầm quan trọng tương đối của các loại phản hồi tường minh và ẩn.

Mô hình này mở rộng thuật toán *Alternating Least Square (ALS)* để phân rã ma trận đơn [8] cho việc học mô hình Co-rating. Để giải bài toán tối ưu ở trên đầu tiên ta cố định $\mathbf{Q}$ và đặt đạo hàm của hàm mục tiêu theo $\mathbf{p}_u$ bằng 0, từ đó

tính được biểu thức của $\mathbf{p}_u$ cho tối thiểu hóa hàm lỗi là:

$$\mathbf{p}_u = (\mathbf{Q}_{I_u^*}\mathbf{Q}_{I_u^*}^T + \eta\mathbf{Q}^T\mathbf{Q} + \lambda\mathbf{I})^{-1}(\mathbf{Q}_{I_u^*}.\mathbf{x}_{:u} + \eta\mathbf{Q}_{I_u^+}.\mathbf{y}_{:u}^+) \tag{28}$$

ở đây $\mathbf{Q}_{I_u^*}$ và $\mathbf{Q}_{I_u^+}$ kí hiệu ma trận con của $\mathbf{Q}$ bởi phép lấy các cột của $\mathbf{Q}$ trong I_u^* và I_u^+. Và $\mathbf{x_{:u}}$ là một vector $|I_u^*|$ chiều được hình thành bởi xếp hạng của u đối với các mục trong I_u^*, $\mathbf{y_{:u}^+}$ là một vector $|I_u^+|$ chiều của 1. Tương tự như vậy, ta có được biểu thức cập nhật cho các thành phần của sản phẩm bằng cách cố định $\mathbf{P}$. Ngoài ra, bài toán tối ưu này có thể sử dụng các phương pháp tối ưu phổ biến như Gradient Descent, Adam để tìm nghiệm.

Điểm mạnh của mô hình:

- Kết hợp hai loại dữ liệu implicit và explicit feedback trong một hệ thống gợi ý nhằm bổ sung thông tin cho nhau giúp tăng cường chất lượng gợi ý.
- Việc cân nhắc tầm quan trọng giữa implicit và explicit feedback được điều chỉnh qua tham số η. Điều này là cần thiết khi áp dụng trên các tập dữ liệu khác nhau.

Những hạn chế của mô hình:

- Cũng giống như các phương pháp dựa trên phân rã ma trận khác, COMF vẫn chưa xử lý được vấn đề Cold Start.

Để kiểm chứng chất lượng của các mô hình trên. Trong chương tiếp theo sẽ trình bày các công cụ, chiến lược để thực nghiệm.

Chương 4 Thử nghiệm đánh giá

Chương này trình bày kết quả thử nghiệm và đánh giá khả năng gợi ý của các mô hình. Để đánh giá hiệu quả của các mô hình, đồ án sử dụng hai bộ dữ liệu trong thương mại điện tử là Retailrocket và MovieLens 1M.

4.1 Cài đặt mô hình

Để cài đặt và đánh giá mô hình VALS đồ án sử dụng ngôn ngữ Java, còn hai mô hình COMF và NeuMF đồ án sử dụng ngôn ngữ python 3.7 với thư viện Tensorflow. Tensorflow là một thư viện mã nguồn mở giúp triển khai các mô hình học máy nhanh chóng và thuận tiện. Tensorflow khá phổ biến vì ưu điểm cho hiệu năng tính toán cao. Ngoài ra Tensorflow còn có kiến trúc linh hoạt cho phép triển khai một cách dễ dàng trên nhiều nền tảng khác nhau như CPUs, GPUs.

Một số thông tin cấu hình của thiết bị sử dụng cho quá trình thử nghiệm:

Tên	Thông tin
Hệ điều hành	Ubuntu 16.04
CPU	Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
CPUs	8
RAM	8GB
HDD	500GB

Bảng 2: Cấu hình phần cứng sử dụng

4.2 Mục tiêu đánh giá

Phần thử nghiệm dưới đây nhằm mục tiêu:

- Phân tích ảnh hưởng của các tham số đối với từng mô hình.
- So sánh hiệu quả giữa các mô hình.
- Các mô hình có khả năng kết hợp dữ liệu implict và explicit feedback có mang lại hiệu quả tốt hơn không?

4.3 Tập dữ liệu sử dụng

Để đánh giá hiệu quả của các mô hình, trong đồ án thử nghiệm trên 2 tập dữ liệu là Retailrocket và MovieLens 1M.

4.3.1 Retailrocket

Mục đích công bố tập dữ liệu là để thúc đẩy các nghiên cứu trong lĩnh vực hệ gợi ý với phản hồi ẩn. Đây là bộ dữ liệu về thương mại điện tử, ghi lại lịch sử tương tác của người dùng đối với một sản phẩm. Bao gồm các hành động sau: view (hành động xem một sản phẩm), add to cart (hành động thêm một sản phẩm vào một giỏ hàng), transaction (hành động mua hàng, tạo một giao dịch với sản phẩm). Đối với dữ liệu view sẽ được chuyển thành dữ liệu implicit feedback và dữ liệu add to cart và transaction sẽ được chuyển thành dữ liệu explicit feedback. Việc huấn luyện mô hình sẽ cố gắng tối ưu gợi ý những sản phẩm cho người dùng đối với hành động add to cart và transaction.

Thông tin thêm về tập dữ liệu xem tại: `https://www.kaggle.com/retailrocket/ecommerce-dataset/home`.

4.3.2 MovieLens 1M

Bộ cơ sở dữ liệu MovieLens 1M được công bố vào 2/2003 bởi GroupLens. Bộ cơ sở dữ liệu này bao gồm 1,000,000 (1M) ratings từ khoảng 6000 users cho 4000 bộ phim.

Thông tin thêm về tập dữ liệu xem tại: `https://grouplens.org/datasets/movielens/1m/`.

4.3.3 Tiền xử lý dữ liệu

Để giảm kích thước của bộ dữ liệu (số lượng users, items) và tính thưa (mật độ tương tác giữa những users và các items) bước tiền xử lý đã loại bỏ đi những user có số lần mua hàng nhỏ hơn 2 lần, và chỉ giữ lại những item đã được các user còn lại tương tác.

Tiền xử lý cho mô hình VALS:

- Bộ dữ liệu Retailrocket: Những sự kiện addtocart hay purchase cùng được đưa về loại purchase. Còn sự kiện view được giữ nguyên vẫn là view.

- Bộ dữ liệu MovieLens 1M: Những ratings từ 4.0 trở lên được coi là purchase, còn lại đưa về dữ liệu view.

Tiền xử lý cho mô hình NeuMF: Do mô hình NeuMF chỉ sử dụng implicit feedback nên với cả hai bộ dữ liệu tất cả các tương tác xảy ra giữa các cặp (user, item) được đưa về 0 và 1, tương ứng với có tương tác và chưa tương tác.

Tiền xử lý cho mô hình COMF:

- Bộ dữ liệu Retailrocket: Những sự kiện addtocart hay purchase cùng được đưa về loại explicit feedback. Còn sự kiện view được coi là implicit feeback.
- Bộ dữ liệu MovieLens 1M: Những ratings từ 4.0 trở lên được coi là explicit feedback, còn lại đưa về dữ liệu implicit feeback.

Bảng 3 dưới đây cho bức tranh mô tả sơ bộ về các tập dữ liệu. Trong đó $\#Interaction$ là số lượng tương tác giữa tất cả người dùng và những sản phẩm. $Sparsity$ (độ thưa) được tính bởi công thức $Sparsity = \#Interaction/(\#User \times \#Item)$

Dataset	#Interaction	#User	#Item	Sparsity
Retailrocket	116,451	4,161	39,606	99.92%
MovieLens 1M	1,000,035	6,037	3,705	95.52%

Bảng 3: Thống kê ước lượng về các tập dữ liệu

4.4 Phương pháp đánh giá và độ đo sử dụng

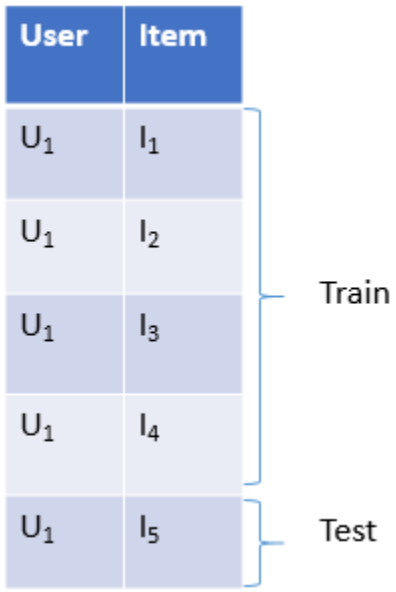
4.4.1 Phương pháp đánh giá

Để kiểm thử khả năng gợi ý của các mô hình, ta sử dụng phương pháp *leave-one-out*. Ở đây item có tương tác explicit lần cuối cùng được tách ra để kiểm thử (*testing*) và mô hình được huấn luyện trên phần dữ liệu còn lại. Vì việc sắp xếp lại tất cả các item cho mỗi user trong quá trình kiểm thử mất quá nhiều thời gian, ở đây ta thực hiện theo chiến lược phổ biến là lấy ra 1000 item ngẫu nhiên mà user chưa tương tác sau đó sắp xếp, đánh giá item test cùng với 1000 item này. Cụ thể như sau:

- Đối với mỗi người dùng **u** ta lọc ra item mà người dùng có tương tác explicit

lần cuối cùng để làm mẫu kiểm thử (mẫu test), gọi item này là $item_{gt}$.

- Xây dựng tập T bằng cách lấy mẫu ngẫu nhiên 1000 items chưa được người dùng **u** tương tác, sau đó thêm mẫu test $item_{gt}$ vào tập T ($T = T \cup item_{gt}$)
- Xếp hạng 1001 sản phẩm trong tập T sau khi mô hình đã được training. Sau đó lấy danh sách top-K sản phẩm được mô hình cho là phù hợp nhất với người dùng u. Việc lấy mẫu 1000 sản phẩm để kiểm thử thay vì kiểm thử trên toàn bộ sản phẩm của tập dữ liệu giúp giảm thiểu thời gian đánh giá mô hình.
- Kiểm tra xem mẫu test $item_{gt}$ có nằm trong *top-K* mẫu được xếp hạng hay không và đánh giá xem vị trí xuất hiện của mẫu test đó trong *top-K* như thế nào.



Hình 9: Chia dữ liệu huấn luyện và thử nghiệm

Hình 9 minh họa phép lấy mẫu để kiểm thử. Để đánh giá mô hình, đồ án sử dụng hai độ đo phổ biến dùng để đánh giá hệ gợi ý là *Hit Ratio* và *Normalized Discounted Cumulative Gain* [5] được trình bày dưới đây.

4.4.2 Độ đo Hit ratio (HR@K)

Với mỗi user, mô hình đánh giá và đưa ra danh sách top-K các items để gợi ý cho user đó. Danh sách xếp hạng này được xếp hạng theo thứ tự giảm dần của điểm số dự đoán mức độ quan tâm của user với item.

Với độ đo Hit Ratio thường được sử dụng trong các đánh giá top-K. Điểm số

hit được tính như sau:

Nếu mẫu kiểm thử $item_{gt}$ nằm trong danh sách top-K các sản phẩm mô hình đưa ra thì $hit = 1$, ngược lại $hit = 0$.

4.4.3 Độ đo Normalized Discounted Cumulative Gain (NDCG@K)

Nếu chỉ quan tâm mẫu test có nằm trong top-K mẫu được xếp hạng hay không thì chưa thể hiện được tính chính xác của việc sắp thứ tự xếp hạng, điều này rất quan trọng trong thực tế. Chính vì vậy, độ đo NDCG sẽ đánh giá cả về thứ tự mà mẫu test xuất hiện trong danh sách top-K đưa ra.

Điểm số của NDCG được tính như sau:

- Nếu $item_{gt}$ nằm trong danh sách top-K:

$$ndcg = \frac{\log 2}{\log(i+1)}$$

- Ngược lại thì $ndcg = 0$

4.5 Thiết lập tham số

Các mô hình có một số tham số chung sau:

- **Kích thước vector ẩn k**: Thể hiện số chiều vector ẩn của người dùng và sản phẩm. Khi kích thước vector ẩn tăng thì thông tin ẩn của người dùng và sản phẩm phong phú hơn, hàm mục tiêu được tối ưu tốt hơn nhưng kéo theo đó khả năng mô hình bị *Overfitting* tăng. Trong đồ án em thử nghiệm với các giá trị là [8, 16, 32, 64].
- **Trọng số** λ: là trọng số của thành phần Regularization trong hàm lỗi để tránh vấn đề Overfitting. Trong đồ án $\lambda = 0.005$
- **Số vòng lặp (hoặc epoch)**: Số lần lặp trên toàn bộ tập dữ liệu huấn luyện, trong đồ án số iterations/epochs được sử dụng là 100.
- **top-K**: Top K trong đánh giá độ đo Hit ratio (HR@K) và Normalized Discounted Cumulative Gain (NDCG@K). Trong đồ án sử dụng top-10, tương ứng với HR@10 và NDCG@10.

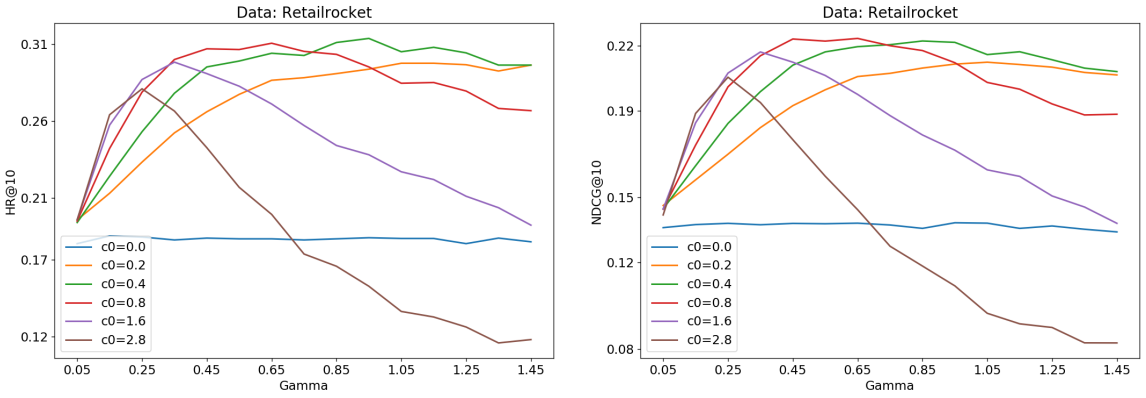
Để đánh giá ảnh hưởng của các tham số riêng của từng mô hình và so sánh giữa các mô hình, các tham số chung này được giữ cố định.

4.6 Ảnh hưởng của tham số đối với mô hình VALS

4.6.1 Ảnh hưởng của tham số c_v, γ_1 và γ_2

Để phân tích ảnh hưởng của trọng số c_v (trọng số của dữ liệu xem) và γ_1, γ_2 (hai mức lề tương ứng giữa sản phẩm mua và chưa xem đối với sản phẩm xem) ta thiết lập như sau:

- Không mất tính tổng quát ta có thể đặt đồng bộ $c_v = c_0/M$, M là số lượng sản phẩm.
- Tương tự, đặt đồng bộ $\gamma_1 = \gamma_2 = \gamma$.
- $\omega_{ui} = 1$: Trọng số của dữ liệu mua.
- $s_j = 0.025$: Trọng số của dữ liệu chưa xem.
- Kích thước vector ẩn = 64.

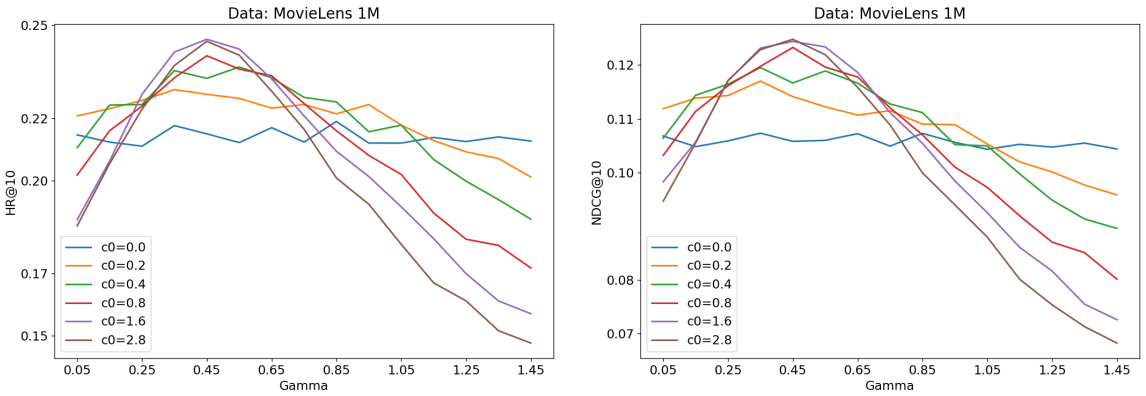


(a) Đo bởi HR@10 (b) Đo bởi NDCG@10

Hình 10: Ảnh hưởng của c_0, γ trên bộ dữ liệu Retailrocket

Hình 10 và Hình 11 là biểu đồ lưới phân tích ảnh hưởng của c_0 và γ trên hai tập dữ liệu Retailrocket và MovieLens 1M. Giá trị c_0 nằm ở dải [0.0, 0.2, 0.4, 0.8, 1.6, 2.8] còn γ từ 0.05 tới 1.45 với bước nhảy là 0.1.

Qua thử nghiệm trên, khi c_0 quá nhỏ hoặc quá lớn, hiệu quả của mô hình đều bị giảm xuống, điều đó chứng tỏ rằng trọng số của tương tác xem cần được xem xét cẩn trọng. Với một người dùng **u**, giá trị γ_1 là khoảng cách lề của mức độ phù hợp giữa những sản phẩm **i** đã mua và sản phẩm **v** từng xem, còn γ_2 là hai khoảng cách lề của mức độ phù hợp giữa những sản phẩm **v** từng xem và sản phẩm **j** chưa xem. Với mọi giá trị c_0 ta thấy rằng giá trị tốt nhất của γ là 0.45. Điều này rất hợp lý vì ta mong muốn điểm số dự đoán của những cặp (**u, i**) có



(a) Đo bởi HR@10 (b) Đo bởi NDCG@10

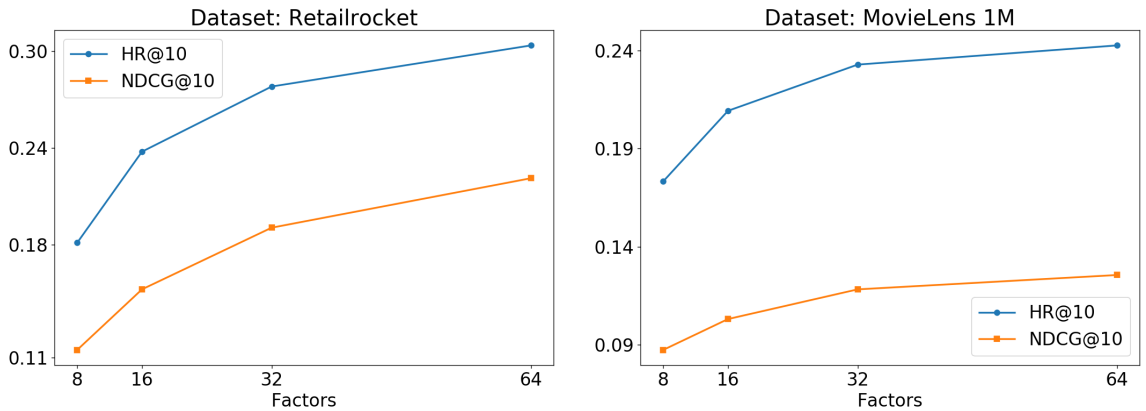
Hình 11: Ảnh hưởng của c_0, γ trên bộ dữ liệu MovieLens 1M

tương tác mua bằng 1, còn (**u, j**) không có tương tác (chưa xem) bằng 0.

Theo sự thăm dò tham số ở trên, ta cố định γ và c_0 theo hiệu quả tốt nhất được đo bằng HR@10 và NDCG@10. Cụ thể với Retailrocket, $c_0 = 0.8$, $\gamma = 0.45$; còn với MovieLens 1M, $c_0 = 1.6$, $\gamma = 0.45$. Những giá trị này được sử dụng cho các thử nghiệm khác của VALS.

4.6.2 Ảnh hưởng của kích thước vector ẩn

Qua mục phân tích ảnh hưởng của trọng số c_v, γ_1 và γ_2, ta chọn ra bộ c_0 và γ tốt nhất cho từng tập dữ liệu. Để xét ảnh hưởng của tham số kích thước vector ẩn. Ta cố định các tham số còn lại như đã được lựa chọn ở mục 4.6.1.



(a) Retailrocket (b) MovieLens 1M

Hình 12: Ảnh hưởng của kích thước vector ẩn với VALS

Kích thước vector ẩn là tham số quan trọng, ảnh hưởng khá nhiều đến chất

lượng của mô hình. Kích thước vector ẩn thể hiện số thuộc tính ẩn học được cho người dùng và sản phẩm qua đó để thể hiện được độ tương đồng giữa người dùng và sản phẩm.

Hình 12 cho thấy, trên cả hai tập dữ liệu khi kích thước vector ẩn tăng chất lượng mô hình cũng tăng theo.

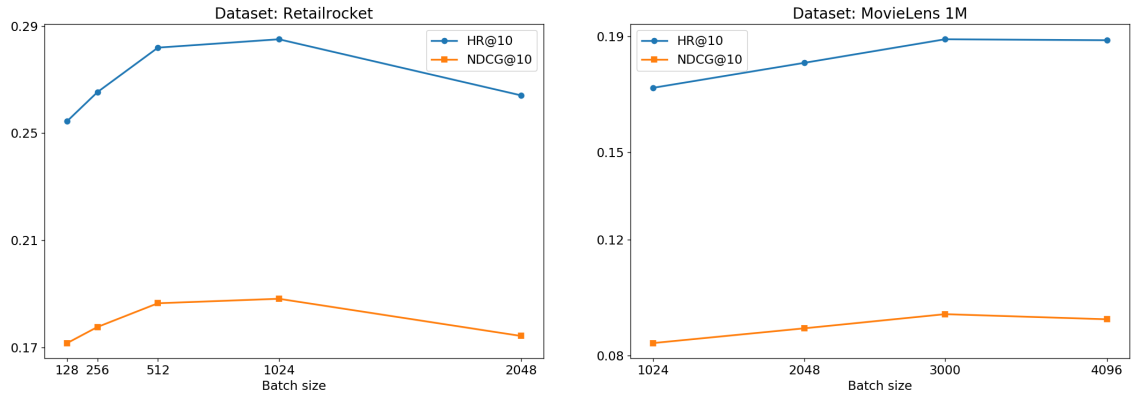
4.7 Ảnh hưởng của tham số đối với mô hình NeuMF

Trong đồ án mô hình NeuMF được cài đặt số tầng ẩn của MLP là 2 tầng ẩn với kích thước [2*k, k] (k là kích thước vector ẩn).

4.7.1 Ảnh hưởng của batch size

Batch size là số lượng mẫu dữ liệu được đưa vào mỗi lượt lan truyền của mạng trong quá trình học. Việc tìm ra batch size tốt cho mô hình trên tập Retailrocket được thực hiện với những giá trị [128, 256, 512, 1024, 2048], còn trên tập MovieLens 1M thực hiện trên các giá trị [1024, 2048, 3000, 4096]. Để điều tra ảnh hưởng của batch size đối với mô hình NeuMF. Ngoài các tham số chung được cố định, các tham số khác của mô hình được thiết lập là:

- *Learning Rate* = 0.001: Tốc độ học của mô hình khi được huấn luyện bằng giải thuật Adam.
- Kích thước vector ẩn = 64.



(a) Retailrocket (b) MovieLens 1M

Hình 13: Ảnh hưởng của batch size với NeuMF

Trong Hình 13 cho thấy khi batch size thay đổi thì hiệu quả của mô hình cũng thay đổi theo. Đối với batch size = 2048, trên tập Retailrocket chất lượng của

mô hình bị giảm, còn trên tập MovieLens 1M vẫn khá tốt. Trở lại với Bảng 3 (Thống kê ước lượng về các tập dữ liệu) ta thấy rằng tập Retailrocket có độ thưa rất cao 99.92%, còn tập MovieLens 1M có độ thưa thấp hơn 95.52%. Ta có thể phán đoán rằng, khi tập dữ liệu có độ thưa cao NeuMF cần batch size nhỏ, còn khi tập dữ liệu có độ thưa thấp NeuMF cần batch size lớn hơn.

Độ lớn của batch size có ảnh hưởng tới tốc độ huấn luyện mô hình NeuMF, với batch size lớn mô hình sẽ được huấn luyện nhanh hơn. Chi tiết ảnh hưởng của batch size với tốc độ huấn luyện được cho trong Bảng 4.

Qua thử nghiệm ảnh hưởng của batch size ở trên, ta chọn batch size = 1024 cho tập Retailrocket và batch size = 3000 cho tập MovieLens 1M. Lựa chọn batch size này được cố định để sử dụng trong các thử nghiệm khác của NeuMF.

Dataset	Batch size	time/epoch
Retailrocket	128	83s
	256	66s
	512	33s
	1024	17s
	2048	9s
MovieLens 1M	128	271s
	256	146s
	512	79s
	1024	48s
	2048	32s

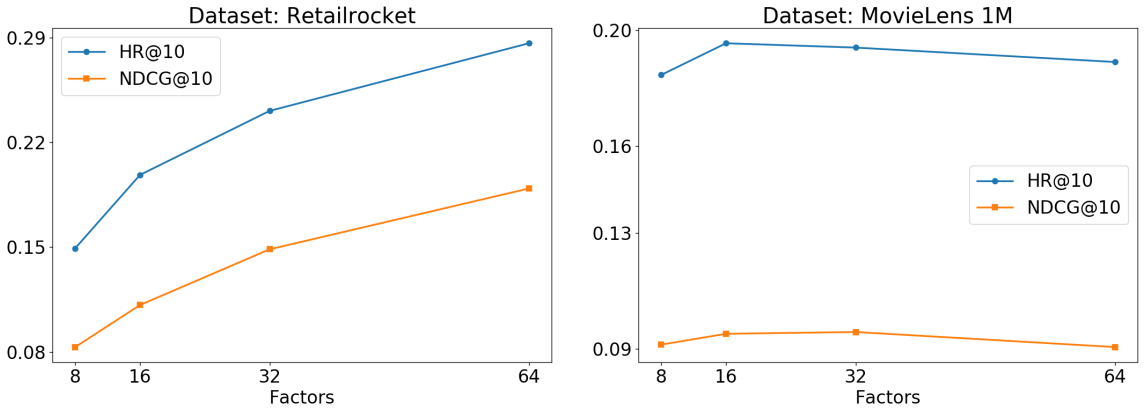
Bảng 4: Ảnh hưởng của batch size với tốc độ huấn luyện NeuMF

4.7.2 Ảnh hưởng của kích thước vector ẩn

Để điều tra ảnh hưởng của kích thước vector ẩn đối với mô hình NeuMF. Các tham số khác của mô hình được giữ cố định là:

- *Learning Rate* = 0.001: Tốc độ học của mô hình khi được huấn luyện bằng giải thuật Adam.
- *Batch size* được chọn theo kết quả mục 4.7.1

Quan sát Hình 14 ta có nhận xét như sau. Đối với tập dữ liệu Retailrocket khi

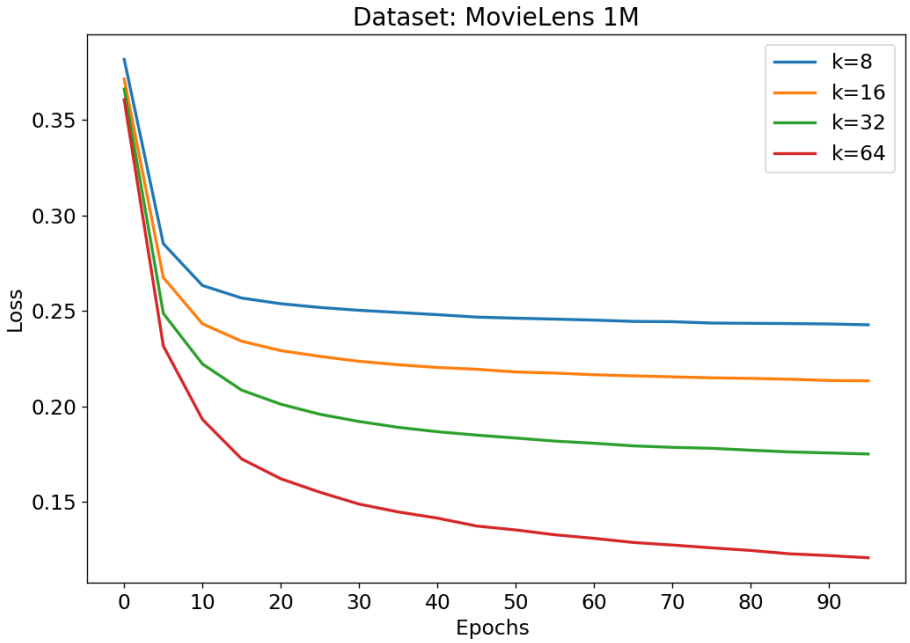


(a) Retailrocket (b) MovieLens 1M

Hình 14: Ảnh hưởng của kích thước vector ẩn với NeuMF

kích thước vector ẩn tăng thì chất lượng mô hình cũng tăng theo, nhưng trên tập dữ liệu MovieLens 1M hiệu quả tốt nhất khi kích thước vector ẩn ở ngưỡng 16 và 32. Theo như được đề cập ở mục 4.5 khi kích thước vector ẩn tăng có khả năng làm cho mô hình rơi vào trạng thái *Overfitting*, liệu rằng điều này có đúng hay không? Để làm rõ được điều này ta quan sát mức độ giảm của hàm lỗi ở Hình 15.

Nhắc lại rằng, ở những thử nghiệm trong đồ án hiệu quả của các mô hình được đo trên tập test, tập này hoàn toàn không được đưa vào huấn luyện ở các mô hình. Ở một góc nhìn khác ta có thể coi tập test tương ứng với tập tối ưu tham số (*Validation Set*). Qua Hình 14 và Hình 15 ta có thể thấy được rằng, với kích thước vector ẩn là 64, mặc dù giá trị lỗi trên tập train được giảm rất sâu, nhưng độ chính xác phán đoán trên tập test bị giảm. Do đó ta khẳng định rằng, trên tập dữ liệu MovieLens 1M khi kích thước vector ẩn quá lớn, mô hình NeuMF bị *Overfitting*.

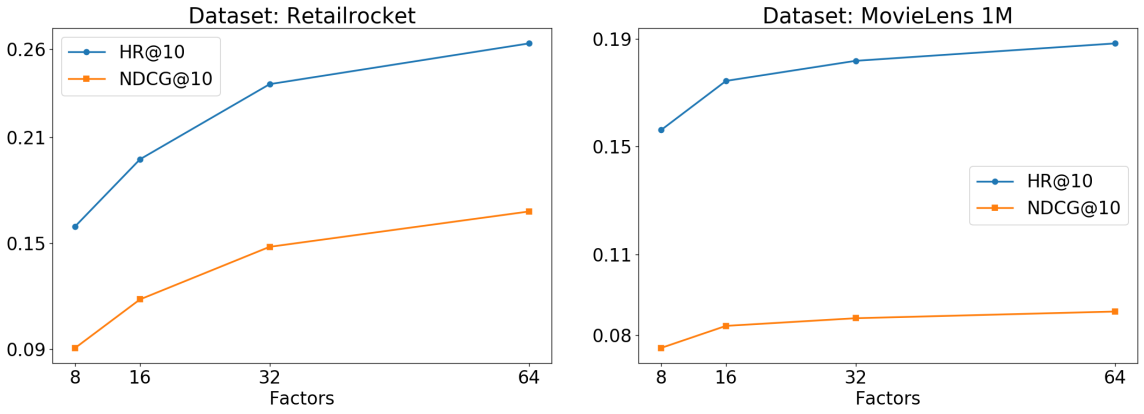


Hình 15: Giá trị hàm lỗi trên tập MovieLens 1M với các kích thước vector ẩn

4.8 Ảnh hưởng của tham số đối với mô hình COMF

4.8.1 Ảnh hưởng của kích thước vector ẩn

Hình 16 thể hiện ảnh hưởng của kích thước vector ẩn thông qua hai độ đo HR@10 và NDCG@10 trên hai tập dữ liệu. Thông qua kết quả Hình 16 thấy được kích thước vector ẩn tăng đồng nghĩa với việc chất lượng của mô hình cũng tăng theo.

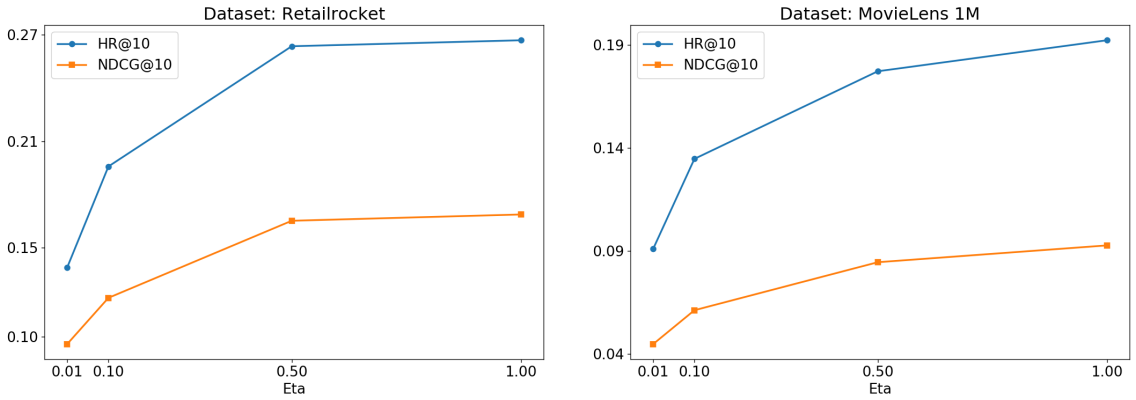


(a) Retailrocket (b) MovieLens 1M

Hình 16: Ảnh hưởng của kích thước vector ẩn với COMF

4.8.2 Ảnh hưởng của tham số eta

Trong Hình 17, ta thấy được việc bổ sung thông tin cho nhau giữa phản hồi tường minh và phản hồi ẩn, khi thay đổi tầm quan trọng giữa chúng cũng làm thay đổi chất lượng của mô hình. Từ đồ thị ta thấy giữa phản hồi tường minh và phản hồi ẩn, hai thông tin này có tầm ảnh hưởng tương đối cân bằng nhau quả đó thấy được thông tin về phản hồi ẩn khá quan trọng khi bổ sung cho phản hồi tường minh để tăng chất lượng của mô hình.



(a) Retailrocket (b) MovieLens 1M

Hình 17: Ảnh hưởng của eta với COMF

4.9 Đánh giá hiệu quả giữa VALS, NeuMF và COMF

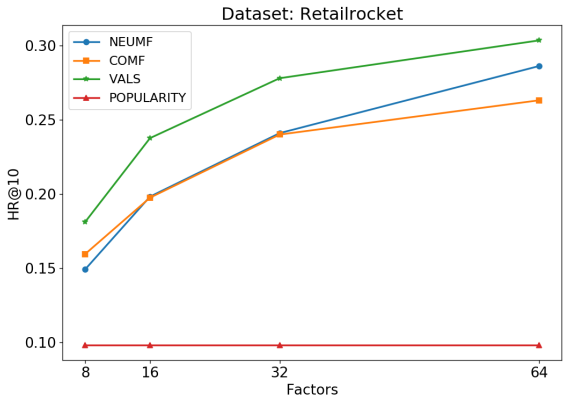
Để so sánh hiệu quả của các mô hình VALS, NeuMF và COMF, đồ án so sánh trên hai tập dữ liệu và đánh giá bằng HR@10 và NDCG@10. Các tham số chung của các mô hình được thiết lập cùng bộ tham số, các tham số riêng được đặt theo kết quả điều tra ảnh hưởng đã được trình bày trong các mục ở trên. Kết quả của việc so sánh được mô tả chi tiết dưới.

Trong phần này em cài đặt thêm thuật toán gợi ý theo thống kê tần suất người dùng sử dụng sản phẩm. Tức sản phẩm có lượng người dùng tương tác cao thì khả năng được gợi ý càng cao, điểm số dự đoán tương tác của cặp (**u, i**) bằng số lượng người dùng sử dụng **i** ta đặt tên cho mô hình này là **POPULARITY**.

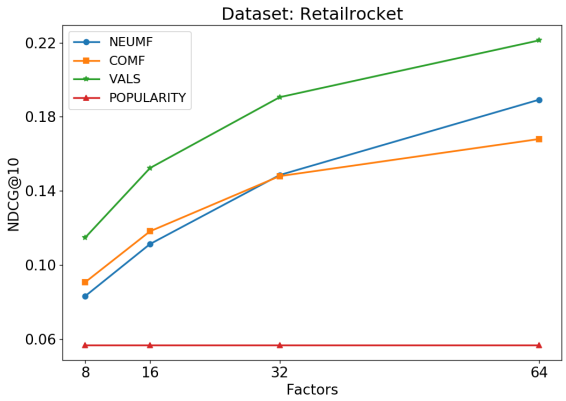
4.9.1 So sánh các mô hình theo kích thước vector ẩn

Hình 18 là so sánh hiệu quả của các mô hình trên tập dữ liệu Retailrocket theo các kích thước vector ẩn khác nhau. Hình a) là sử dụng độ đo HR@10, còn

b) sử dụng độ đo NCDG@10.



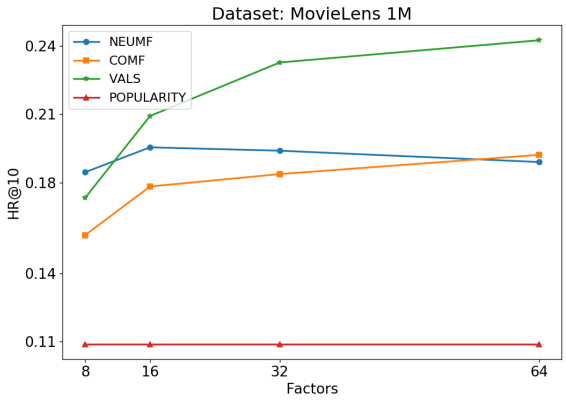
(a) HR@10



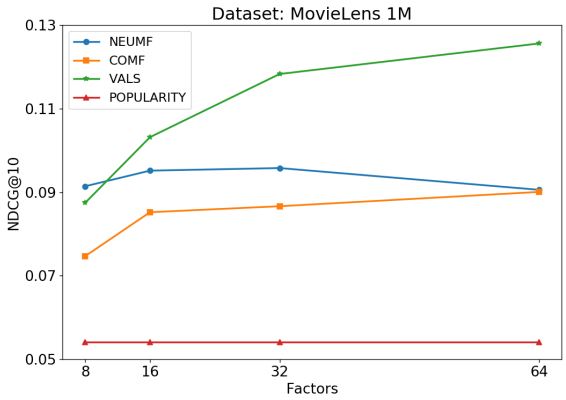
(b) NDCG@10

Hình 18: So sánh các mô hình trên tập Retailrocket

Tương tự, Hình 19 là kết quả so sánh trên tập dữ liệu MovieLens 1M.



(a) HR@10



(b) NDCG@10

Hình 19: So sánh các mô hình trên tập MovieLens 1M

Nhìn chung, mô hình VALS cho hiệu quả tốt nhất trên cả hai tập dữ liệu. Phương pháp gợi ý dựa trên thống kê *POPULARITY* quá đơn giản và không hướng đến được sở thích riêng của từng người dùng. Do vậy, *POPULARITY* cho kết quả rất thấp và tất nhiên các mô hình khác sẽ vượt qua nó.

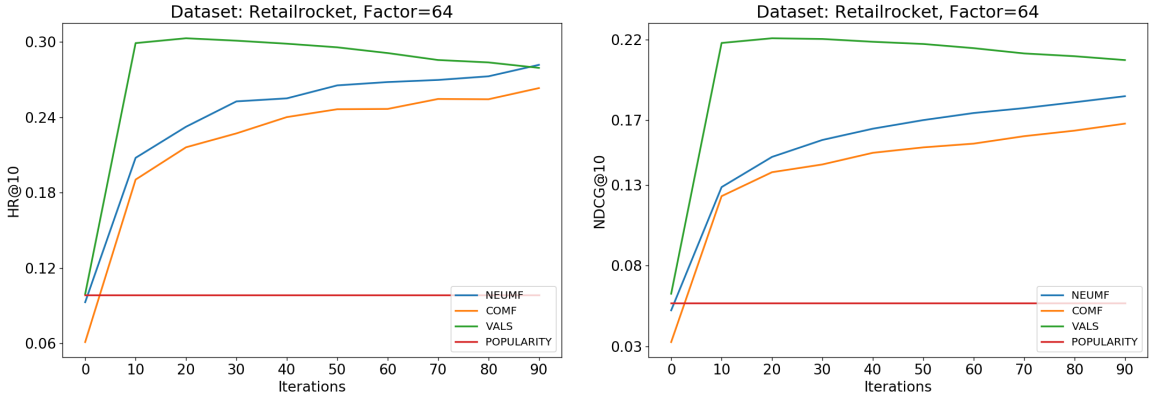
Trên tập dữ liệu MovieLens 1M như phân tích ở mục 4.7.2 mô hình NeuMF bị *Overfitting* đối với kích thước vector ẩn lớn, dẫn tới với kích thước vector ẩn là 8 thì cho hiệu quả tốt nhất so với các mô hình khác. Nhưng nếu lấy kết quả tốt nhất của NeuMF (khi kích thước vector ẩn là 8) so với kết quả tốt nhất của VALS (khi kích thước vector ẩn là 64) thì NeuMF vẫn còn kém hơn rất nhiều.

4.9.2 So sánh các mô hình theo các vòng lặp/epochs

Để thấy rõ hơn hiệu quả giữa các mô hình, dưới đây là một số biểu đồ so sánh các mô hình qua các vòng lặp (Iterations/Epochs).

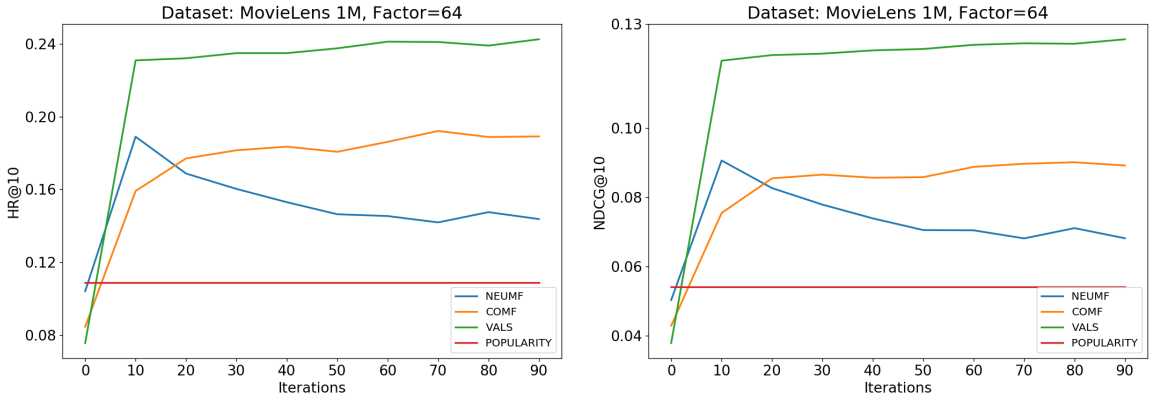
Kích thước vector ẩn bằng 64:

Đối với kích thước vector ẩn bằng 64, ta có Hình 20 và Hình 21 là biểu đồ so sánh các mô hình trên hai tập dữ liệu Retailrocket và MovieLens 1M. Qua đó ta thấy rằng VALS cho kết quả tốt nhất trên cả hai tập dữ liệu trong thời gian khá nhanh (sau 10 vòng lặp). Còn NeuMF và COMF, đối với tập Retailrocket mô hình NeuMF cho kết quả tốt hơn COMF một chút; đối với tập tập MovieLens 1M kết quả tốt nhất của NeuMF và COMF ở mức ngang nhau nhưng NeuMF đạt được kết quả tốt nhất nhanh hơn (sau 10 epochs).



(a) HR@10 (b) NDCG@10

Hình 20: So sánh các mô hình trên tập Retailrocket, Factor = 64



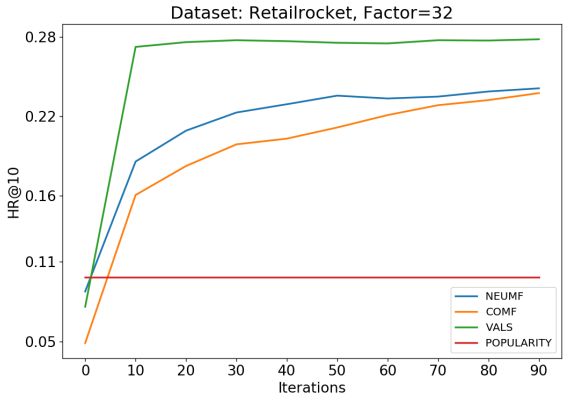
(a) HR@10 (b) NDCG@10

Hình 21: So sánh các mô hình trên tập MovieLens 1M, Factor = 64

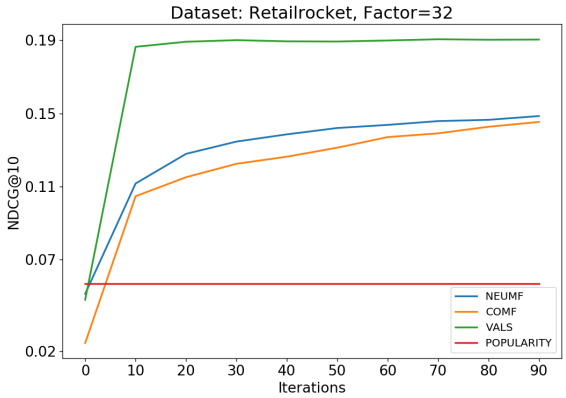
*** Một số kết quả thử nghiệm khác**

Kích thước vector ẩn bằng 32

Hình 22 và Hình 23 là biểu đồ so sánh các mô hình với Factor = 32

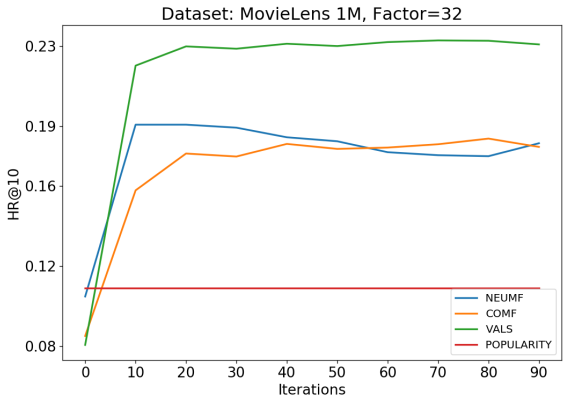


(a) HR@10

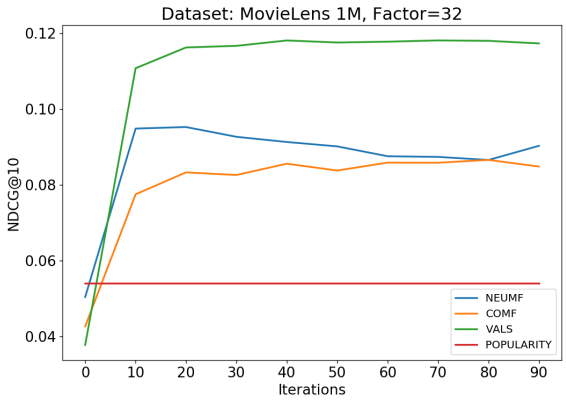


(b) NDCG@10

Hình 22: So sánh các mô hình trên tập Retailrocket, Factor = 32



(a) HR@10

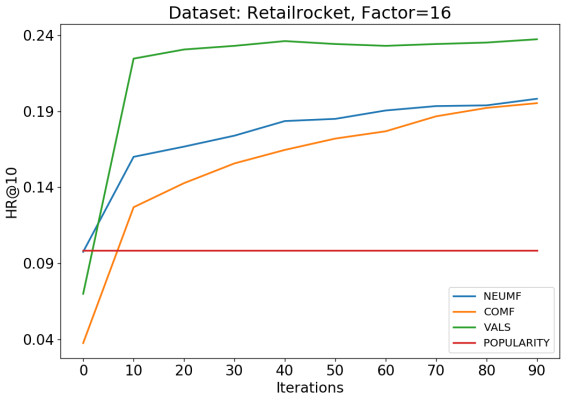


(b) NDCG@10

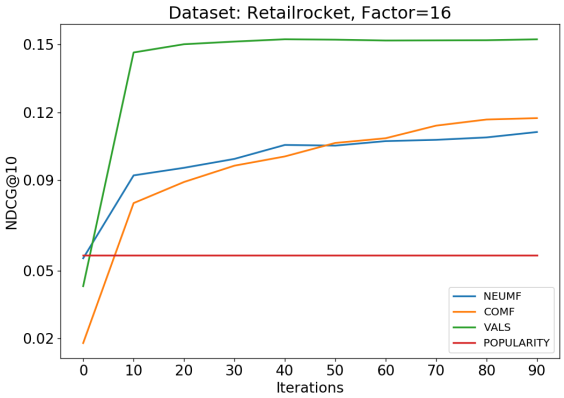
Hình 23: So sánh các mô hình trên tập MovieLens 1M, Factor = 32

Kích thước vector ẩn bằng 16

Hình 24 và Hình 25 là biểu đồ so sánh các mô hình với Factor = 16

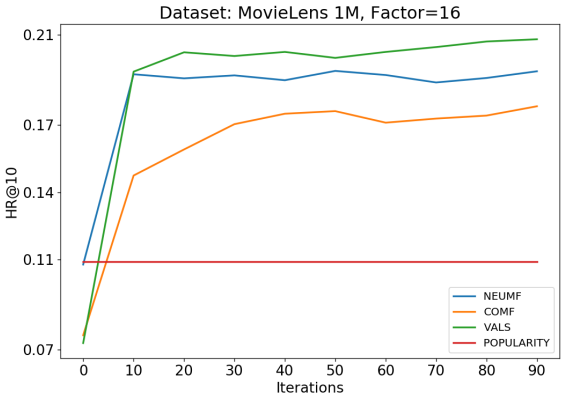


(a) HR@10

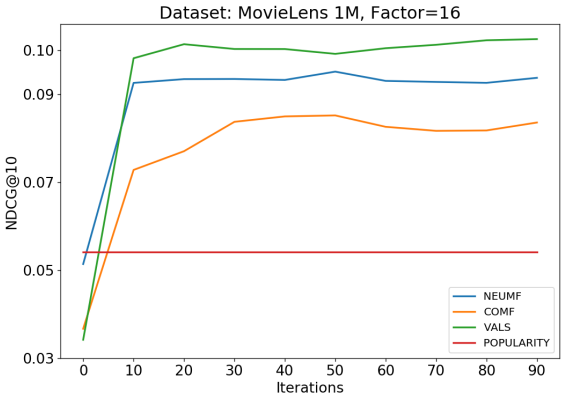


(b) NDCG@10

Hình 24: So sánh các mô hình trên tập Retailrocket, Factor = 16



(a) HR@10

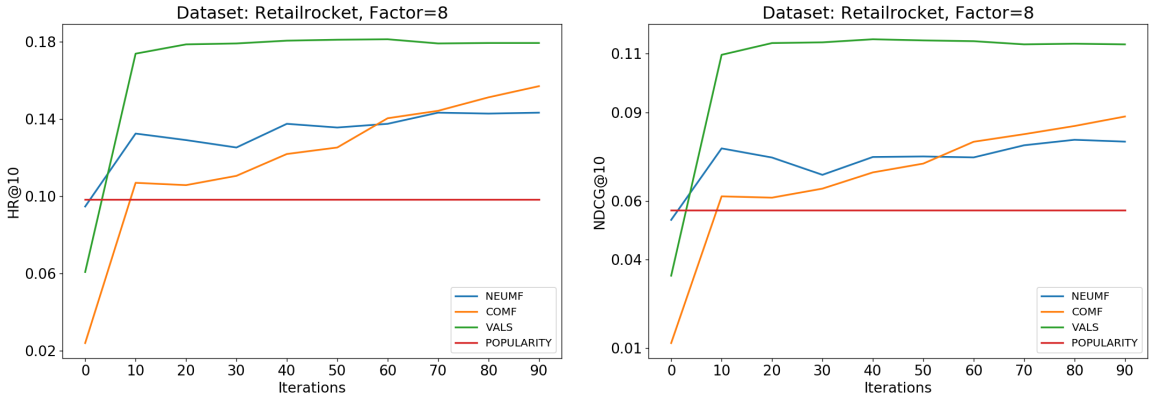


(b) NDCG@10

Hình 25: So sánh các mô hình trên tập MovieLens 1M, Factor = 16

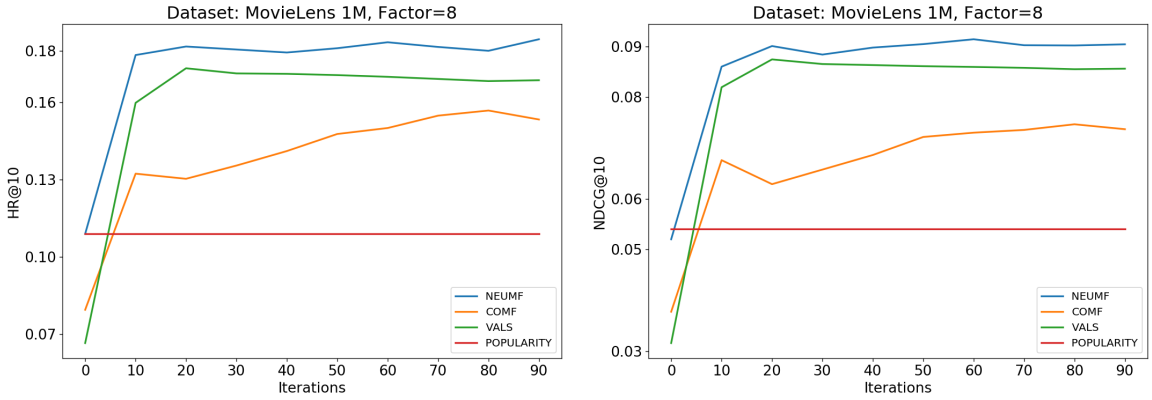
Kích thước vector ẩn bằng 8

Hình 26 và Hình 27 là biểu đồ so sánh các mô hình với Factor = 8



(a) HR@10 (b) NDCG@10

Hình 26: So sánh các mô hình trên tập Retailrocket, Factor = 8



(a) HR@10 (b) NDCG@10

Hình 27: So sánh các mô hình trên tập MovieLens 1M, Factor = 8

4.9.3 Đánh giá chung

Qua thử nghiệm trên, ta tổng hợp một số đánh giá chung cho các mô hình như sau:

- Trong hầu hết các thử nghiệm cho thấy mô hình VALS cho hiệu quả tốt nhất. Mô hình VALS mặc dù được thiết kế chỉ sử dụng dữ liệu implicit feedback, nhưng khi áp dụng trên tập MovieLens 1M ta đã chuyển đổi các ratings >= 4 tương đương với tương tác mua. Điều đó cho thấy VALS có khả năng kết hợp dữ liệu explicit với implicit feedback khi ta tiền xử lý dữ liệu đầu vào phù hợp.

- Hai mô hình NeuMF và COMF đều có những ưu điểm riêng rất tốt. NeuMF có được những tính chất tốt từ mạng nơ-ron để cải tiến phương pháp phân rã ma trận, COMF kết hợp được implicit và explicit feedback. Nhưng kết quả thử nghiệm cho thấy tùy trên bộ dữ liệu và kích thước vector ẩn có

trường hợp mô hình này tỏ ra hiệu quả hơn mô hình kia và ngược lại. Ta có thể nhận xét chung rằng hai mô hình cho kết quả ở ngưỡng ngang bằng nhau.

- Việc kết hợp implicit và explict feedback cũng như sử dụng mạng nơ-ron trong hệ gợi ý là các hướng đi nhiều tiềm năng cần nghiên cứu phát triển.

Chương 5 Kết luận

Chương này tóm tắt lại các vấn đề đã thực hiện và các kiến thức thu được sau khi thực hiện đồ án và các hướng nghiên cứu tiếp theo đối với bài toán.

5.1 Các kết quả đạt được trong quá trình làm đồ án

Trong đồ án này em thực hiện được các công việc sau:

- Tìm hiểu bài toán gợi ý nói chung và bài toán gợi ý sử dụng phản hồi ẩn từ người dùng nói riêng. Phân tích sự khác nhau giữa hai loại phản hồi implicit feedback và explicit feedback, hai loại phản hồi này cần có phương pháp kết hợp để đạt được hiệu quả cao cho hệ gợi ý.
- Tìm hiểu và thử nghiệm các mô hình VALS, NeuMF và COMF trên hai tập dữ liệu Retailrocket và MovieLens 1M. Mỗi mô hình có những ưu nhược điểm khác nhau, nhưng về kết quả thử nghiệm hiện tại cho thấy mô hình VALS đạt kết quả tốt nhất trên hai tập dữ liệu
- Phân tích ảnh hưởng của các tham số đối với từng mô hình. Qua phân tích thu được kinh nghiệm lựa chọn tham số của mô hình, bài toán lựa chọn tham số rất quan trọng khi đưa mô hình áp dụng trọng thực tiễn.
- Trong quá trình làm đồ án, ngoài những kiến thức thu được khi tìm hiểu và thử nghiệm các mô hình cho bài toán gợi ý, em còn có được kinh nghiệm trong việc xây dựng các mô hình cho hệ gợi ý. Đồng thời là những phương pháp để so sánh và đánh giá các mô hình học máy.

5.2 Các hướng nghiên cứu trong tương lai

Các hướng phát triển trong tương lai của đồ án:

- Nhược điểm chung của các mô hình trên là do phát triển dựa trên lọc cộng tác nên gặp vấn đề cold-start, nghĩa là đối với những người dùng và sản phẩm mới mô hình không thể đưa ra gợi ý. Với những người dùng hay sản phẩm mới, mô hình cần phải huấn luyện lại khiến cho tốn kém tài nguyên, thời gian và khó xử lý kịp thời trong những hệ gợi ý yêu cầu tính tức thời (*Realtime*). Nhược điểm này cần được nghiên cứu khắc phục, khi đó các mô hình trên sẽ ổn định hơn và áp dụng trong thực tế tốt hơn.
- Mô hình VALS và NeuMF đang làm việc với dữ liệu phản hồi ẩn, điều này

sẽ lãng phí giá trị của những phản hồi tường minh thu thập được. Cải tiến hai mô hình này có khả năng kết hợp được phản hồi tường minh là một hướng đi rất đáng để nghiên cứu.

- Ngoài ra trong các hệ gợi ý còn nhiều thông tin tương tác giữa người dùng và sản phẩm có thể được kết hợp để bổ sung thông tin giúp quá trình gợi ý được cải thiện hơn.
- Trong tương lai em sẽ có gắng tìm hiểu và phát triển các hướng đi trên, mong rằng sẽ cải thiện cho các mô hình và có khả năng áp dụng tốt cho những nhu cầu thực tế.

Tài liệu tham khảo

Tài liệu

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[2] Jingtao Ding, Guanghui Yu, Xiangnan He, Yuhan Quan, Yong Li, Tat-Seng Chua, Depeng Jin, and Jiajie Yu. Improving implicit recommender systems with view data. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3343–3349, 2018.

[3] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. Collaborative filtering recommender systems. *Found. Trends Hum.-Comput. Interact.*, 4(2):81–173, February 2011.

[4] Alexander Felfernig, Michael Jeran, Gerald Ninaus, Florian Reinfrank, Stefan Reiterer, and Martin Stettinger. Basic approaches in recommendation systems. In *Recommendation Systems in Software Engineering*, 2014.

[5] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1661–1670, New York, NY, USA, 2015. ACM.

[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 173–182, Republic and Canton of Geneva, Switzerland, 2017.

[7] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 549–558, New York, NY, USA, 2016. ACM.

[8] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington,

DC, USA, 2008. IEEE Computer Society.

[9] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug 2009.

[10] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

[11] Nathan N. Liu, Evan W. Xiang, Min Zhao, and Qiang Yang. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1445–1448, New York, NY, USA, 2010. ACM.

[12] Nitin Mishra, Vimal Mishra, and Saumya Chaturvedi. Tools and techniques for solving cold start recommendation. In *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, IML '17, pages 11:1–11:6, New York, NY, USA, 2017. ACM.

[13] Michael J. Pazzani and Daniel Billsus. The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.

[14] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295. ACM, 2001.