TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC

ĐỒ ÁN 1

PHƯƠNG PHÁP PHÂN RÃ MA TRẬN VỚI THUẬT TOÁN TỐI ƯU ADAM TRONG HỆ THỐNG GỢI Ý

Họ tên: Nguyễn Mạnh Thắng

MSSV: 20195915

Lớp: Toán Tin 01

GV hướng dẫn: TS. Nguyễn Thị Ngọc Anh

Hà Nội, 2022

Mục lục

LỜI MỞ ĐẦU

Hệ thống gợi ý dựa trên các mô hình học máy đã và đang đạt được nhiều các nhà bán lẻ điện tử quan tâm và một trong những yếu tố làm nên sự thành công đó chính là áp dụng mô hình phân rã ma trận. Mô hình này có khả năng dự đoán chính xác sở thích, nhu cầu của khách hàng, từ đó kết nối kịp thời với các doanh nghiệp để cung cấp nhiều sự lựa chọn sản phẩm. Mô hình phân rã ma trận là một mô hình đơn giản nhưng mang lại hiệu quả cao do nó có thể nắm bắt được sự tương tác giữa người dùng với sản phẩm. Do đó, trong khuân khổ Đồ án 1 trong chương trình đạo tạo Toán-Tin của viện Toán ứng dụng và Tin học, em xin phép được trình bày cách để xây dựng hệ thống gợi ý dựa trên mô hình phân rã ma trận với thuật toán tối ưu ADAM.

Cụ thể hơn, báo cáo này sẽ trình bày phương pháp phân rã ma trận bằng cách phân tách ma trận User-item thành hai ma trận có rank thấp khác. Hai ma trận thành phần này thể hiện các embedding ứng với người dùng và sản phẩm, sau đó sẽ tính xấp xỉ các giá trị đã biết và điền các giá trị còn thiếu của ma trận User-Item. Em huấn luyện mô hình sử dụng hàm mất mát của tích hai ma trận, tiếp sau đó sẽ sử dụng thuật toán ADAM để tối ưu hàm mất mát đó. Cuối cùng, em sẽ dùng phương pháp sai số trung bình phương (RMSE) để đánh giá kết quả của mô hình trên bộ dữ liệu Movielen-1m.

Sau khi thực nghiệm, mô hình đã có kết quả khả quan khi chỉ số RMSE thấp, thời gian huấn luyện nhanh trên bộ dữ liệu lớn.

Quy tắc viết tắt

RS	Hệ thống gợi ý
MF	Phân rã ma trận
user	Người dùng
item	Sản phẩm
SGD	Hướng giảm Gradient ngẫu nhiên
ADAM	Thuật toán tối ưu ADAM
EMA	Trung bình trượt số mũ
RMSE	Chỉ số bình phương trung bình

LỜI CẢM ƠN

Báo cáo này được thực hiện và hoàn thành tại viện Toán ứng dụng và Tin học trực thuộc Trường Đại học Bách Khoa Hà Nội, nằm trong nội dung học phần Đồ án I của kì học 2021-2.

Em xin được dành lời cảm ơn tới cô Nguyễn Thị Ngọc Anh - Bộ môn toán ứng dụng, là giảng viên đã trực tiếp hướng dẫn và gợi ý cho em đề tài rất thú vị này. Ngoài ra, em gửi lời cảm ơn đến anh Đặng Tiến Đạt - Toán tin k62, bạn Nguyễn Đình Thái - Hệ thống thông tin quản lý k64 đã có những đóng góp ý kiến để báo cáo được hoàn thành một cách tốt nhất.

Dù đã rất cố gắng tuy nhiên đồ án vẫn không tránh khỏi những hạn chế cần khắc phục. Vì vậy, em rất mong quý thầy cô đưa ra những ý kiến góp ý bổ ích để đồ án này tiếp tục được phát triển và có những kết quả mới tốt hơn. Em xin chân thành cảm ơn!

Hà Nội, tháng 08 năm 2022
Sinh viên thực hiện
Nguyễn Mạnh Thắng

Chương 1: GIỚI THIỆU

1.1 Lý do chọn đề tài

Có nhiều hướng tiếp cận để xây dựng nên một hệ thống gợi ý. Tùy thuộc vào nguồn thông tin có được, nhu cầu thực tế, đặc thù riêng của từng lĩnh vực mà mỗi hệ tư vấn sẽ có phương pháp và thuật toán phù hợp cho riêng mình. Kỹ thuật phân rã ma trận (Matrix Factorization) là một trong những kỹ thuật được sử dụng để xây dựng một hệ thống gợi ý. Kỹ thuật này được đánh giá cao nhờ khả năng cải thiện độ chính xác của các kỹ thuật khác, thêm vào đó hoạt động tốt với các dữ liệu thưa thớt và đặc biệt nó sẽ khai thác được sự tương tác thật giữa người dùng và sản phẩm. Đây cũng là lý do mà em chọn đề tài "PHƯƠNG PHÁP PHÂN RÃ MA TRẬN VỚI THUẬT TOÁN TỐI ƯU ADAM TRONG HỆ THỐNG GỢI Ý".

1.2 Mục tiêu của đề tài

Mục tiêu của đồ án hướng đến là cấu trúc của mô hình phân rã ma trận với thuật toán ADAM. Từ đó, đề xuất mô hình này vào hệ thống gợi ý để dự đoán xếp hạng người dùng dành cho bộ phim trong tập dữ liệu Movielen 1m. Mục tiêu sẽ đạt được thông qua các nội dung chính sau:

- Nắm rõ khái niệm về thế nào là một hệ thống gợi ý
- Đề xuất phương pháp phân rã ma trận sử dụng thuật toán tối ưu ADAM
- Áp dụng mô hình để dự đoán đánh giá người dùng cho các bộ phim thuộc tập dữ liệu

1.3 Phạm vi và đối tượng nghiên cứu

Đối tượng nghiên cứu: Phương pháp phân rã ma trận với thuật toán tối ưu ADAM

Phạm vi nghiên cứu: Đồ án này sẽ tập trung quan tâm ra cách để tạo ra một hệ thống gợi ý dựa trên phương pháp phân rã ma trận

1.4 Phương pháp nghiên cứu

Để đồ án đạt được kết quả tốt em đã thực hiện các phương pháp sau:

- Phương pháp điều tra, quan sát khoa học
- Phương pháp mô hình hóa
- Phương pháp thực nghiệm
- Phương pháp phân tích và đánh giá dữ liệu

1.5 Bố cục đồ án

Chương 2 giới thiệu về hệ thống gợi ý, các mô hình của hệ thống gợi ý và ứng dụng của nó. Chương 3 sẽ trình bày phương pháp phân rã ma trận và thuật toán tối ưu ADAM. Chương 4 trình bày trực quan hóa về bộ dữ liệu Movielen 1m. Chương 5 đưa ra kết luận cho đồ án và hướng nghiên cứu tiếp theo.

CHƯƠNG 2: TỔNG QUAN VỀ HỆ THỐNG GỢI Ý

2.1 Giới thiệu về hệ thống gợi ý

Hệ thống gợi ý (Recommendation systems - RS) là một dạng của hệ thống lọc thông tin, nó được sử dụng để dự đoán sở thích hay xếp hạng mà người dùng có thể dành cho một mục thông tin nào đó mà họ chưa xem xét tới trong quá khứ (có thể là bộ phim, bài hát, đoạn video trên youtube,...)[5].

Ví dụ, trong hệ thống bán hàng trực tuyến (chẳng hạn như Shopee), nhằm tối ưu hóa khả năng mua sắm của khách hàng (user), hệ thống quan tâm đến việc nhưng khách hàng nào đã yêu thích những sản phẩm (item) nào bằng cách dựa trên lịch sử tương tác của họ (các tương tác này có thể là tương tác hiện (nút like, đánh giá sao,...) hoặc tương tác ẩn (thời gian xem sản phẩm đó, số lần click chuột,...). Từ đó, hệ thống sẽ dự đoán sở thích của người dùng đó và đưa ra gợi ý phù hợp cho họ. Điều này xảy ra khá thường xuyên khi người dùng tìm kiếm một sản phẩm liên quan đến quần áo đi biển, ngay sau đó Shopee sẽ gợi ý những món phụ kiện đi biển kèm theo như mắt kính, mũ, kem chống nắng,... Ngoài việc thành công trong lĩnh vực thương mại điện tử, hệ thống gợi ý còn được áp dụng tương đối thành công ở trong nhiều lĩnh vực khác:

- Netflix lại quan tâm đến việc dự báo những phim người tiêu dùng thích xem dựa trên kết quả bình chọn trước đó, thói quen xem phim và các đặc tính của phim (thể loại phim, diễn viên).
- Youtube sẽ đưa ra những gợi ý video tương tự những video mà bạn đã xem hoặc đề xuất những video mà đang là chủ đề hot hiện nay.
- Tiktok dựa trên thời gian người dùng xem, thích video và đề xuất những video có nội dung tương tự .
- Facebook sẽ học được từ những hashtag của người dùng để phân tích nội dung bài đăng, sau đó những bài đăng không có hashtag nhưng có chung chủ đề sẽ được gợi ý cho người dùng.

Đặc điểm của việc tư vấn là mang tính cá nhân, nghĩa là nó chỉ phù hợp với một số người dùng (hay một nhóm người dùng) có cùng một số đặc tính đã được khảo sát trước đó. Do vây, hệ thống sẽ gặp khó khăn khi gặp người dùng hoặc sản phẩm mới vì quá ít thông tin được thu thập[3].

Công dụng của hệ thống gợi ý đối với doanh nghiệp:

- Gia tăng doanh số bán hàng: nhờ đáp ứng kịp thời nhu cầu của khách hàng và tư vấn những mặt hàng liên quan thay vì chỉ bán được các sản phẩm đơn lẻ
- Gia tăng sự thỏa mãn của khách hàng: Khách hàng được đáp ứng kịp thời, họ càng muốn sử dụng các tiện ích khác của doanh nghiệp
- Tăng độ trung thành của khách hàng

Đối với khách hàng:

- Giúp họ có thêm nhiều lựa chọn hơn khi mua hàng
- Kịp thời đáp ứng được nhu cầu của mình
- Nắm bắt nhanh nhạy các xu hướng hiện nay

2.2 Phát biểu bài toán

Trong RS, thông thường người ta chỉ cần quan tâm đến ba thông tin chính: người dùng (user), sản phẩm (item), đánh giá (rate) của người dùng lên sản phẩm đó. Các thông tin này được biểu diễn thông qua một ma trận như trong Hình 1, hay còn được gọi là ma trận User-item-rating. Ở đó, mỗi hàng là một user, mỗi cột là một item, và mỗi giá trị của ma trận thể hiện điểm đánh giá của user lên item đó. Các ô có giá trị là những item đã được user đánh giá trong quá khứ, còn những ô trống là chưa được đánh giá. Điểm đặc biệt của hầu hết những ma trận này là sự thưa thớt của giá trị trong đó (người ta còn gọi là ma trận thưa - sparse matrix). Điều này cũng đúng với thực tế khi mà người dùng thường không mấy khi đánh giá những sản phẩm mà họ đã từng sử dụng[3].

	i_1	i_2	i_3	i_4	i_5
u_1	5		4	1	
u_2		3		3	
u_3		2	4	4	1
u_4	4	4	5		
u_5	2	4		5	2

Hình 1: Ma trận biểu diễn sự đánh giá của người dùng cho sản phẩm (User-item matrix)

Công việc chính của RS là dựa vào các ô đã có giá trị ở ma trận trên và điền nốt các ô còn trống. Sau đó sắp xếp thứ tự kết quả dự đoán của từng user và chọn ra Top-N items theo thứ tự, từ đó gợi ý chúng cho người dùng.

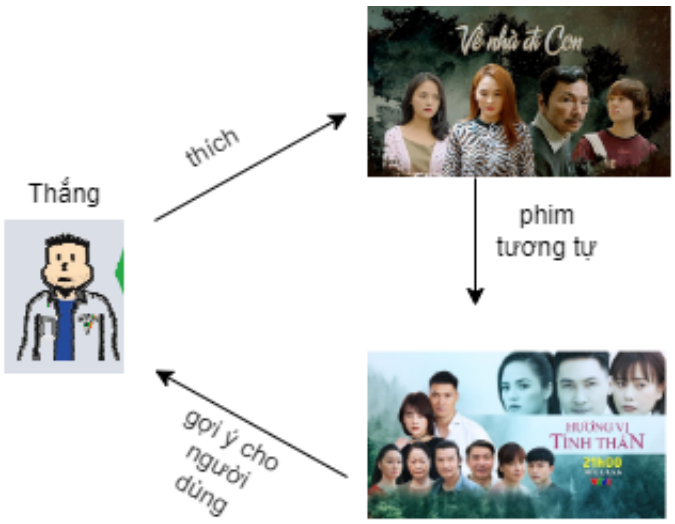
2.3 Giải thuật cho hệ thống gợi ý

Hiện tại trong RS có rất nhiều mô hình được đề xuất, tuy nhiên trong đồ án này em chỉ nêu ra : mô hình lọc nội dung(Content Filtering) và mô hình lọc công tác(Collaborative Filtering)[5] [1].

2.3.1 Mô hình lọc nội dung

Cách tiếp cận lọc nội dung tạo ra một hồ sơ cho mỗi người dùng hoặc sản phẩm để mô tả bản chất của nó. Ví dụ, một hồ sơ phim có thể bao gồm các thuộc tính liên quan đến thể loại của nó, các diễn viên tham gia, sự phổ biến của phòng vé, v.v. Hồ sơ người dùng có thể bao gồm thông tin nhân khẩu học hoặc câu trả lời được cung cấp trên bảng câu hỏi phù hợp. Các hồ sơ cho phép các chương trình liên kết người dùng với các sản phẩm phù hợp. Một ví dụ rất thành công của chiến thuật lọc nội dung là the Music Genome Project của nền tảng nghe nhạc Pandora.Com. Hệ thống này chấm điểm mỗi bài hát dựa trên hàng trăm đặc điểm âm nhạc khác nhau. Các bài hát được phân tích thuộc tính (định nghĩa là genre) sẽ cho ra không chỉ đặc điểm bài hát mà còn là gu âm nhạc của người nghe[5].

Trong hệ thống này, mô hình dự đoán liệu một người dùng có thích một sản phẩm không dựa trên lịch sử dữ liệu của người dùng đó đối với các sản phẩm tương tự. Độ quan tâm của những người dùng khác không được sử dụng. Ví dụ như ở Hình 2 Thắng là một khách hàng của 1 web phim, hệ thống sẽ tìm hiểu xem những đặc tính của Thắng cũng như lịch sử xem của Thắng và thấy anh ấy đã thích bộ phim Về nhà đi con. Sau đó hệ thống cũng sẽ gợi ý bộ phim tương tự cùng thể loại, hoặc cùng nhà phát hành, cùng diễn viên,... và ở đây là bộ phim Hương vị tình thân.



Hình 2: Thắng thích phim Về nhà đi con thì hệ thống sẽ gợi ý phim tương tự đó là Hương vị tình thân

Ưu điểm:

- Việc xây dựng mô hình cho mỗi người dùng độc lập với nhau, vì vậy khi có dữ liệu mới từ những người dùng khác, ta không cần cập nhật mô hình cho người dùng này. Việc này giúp hệ thống có thể phục vụ được lượng người dùng lớn một cách dễ dàng.
- Nếu một sản phẩm rất ít người quan tâm nhưng giống với những sản phẩm khác mà một người dùng từng thích, sản phẩm đó sẽ có cơ hội cao được giới thiệu tới người dùng.

Nhược điểm:

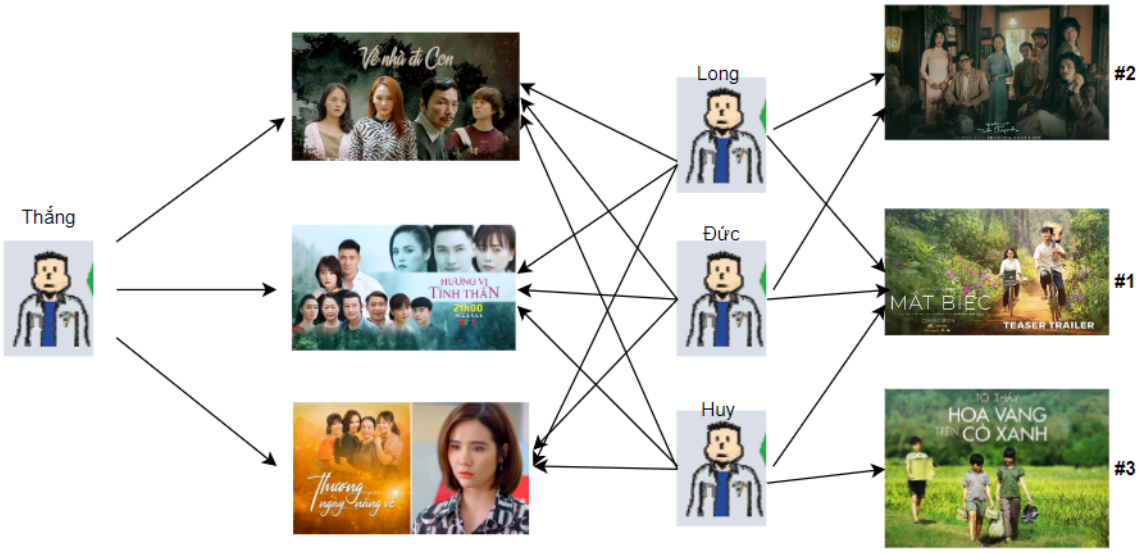
- Thông tin về những người dùng tương tự có thể rất hữu ích nhưng không được khai thác, làm giảm độ chính xác của mô hình.
- Mô hình chỉ dựa trên những dữ liệu đã có mà không mở rộng sự yêu thích của người dùng tới những sản phẩm khác.
- Việc xây dựng đặc trưng cho sản phẩm đôi khi phải được thực hiện thủ công, ví dụ gán nhãn thể loại. Điều này sẽ hạn chế năng lực của hệ thống khi có rất nhiều sản phẩm.

2.3.2 Mô hình lọc cộng tác

Thuật ngữ này được định nghĩa bởi Tapestry, hệ gợi ý đầu tiên trên thế giới. Cách này chỉ dựa trên hành vi của người dùng trước đây. Ví dụ: Giao dịch trước

đó hoặc xếp hạng sản phẩm, mà không yêu cầu tạo hồ sơ rõ ràng. Chiến thuật này phân tích mối liên quan giữa người dùng và các thuộc tính của sản phẩm để định nghĩa quan hệ giữa người dùng - sản phẩm. Chiến thuật này không bị giới hạn miền, phân tích được nhiều khía cạnh dữ liệu mà chiến thuật lọc nội dung gặp khó khi xử lý, tuy nhiên không hiệu quả khi gặp các dữ liệu mới ít thông tin (cold start problem). Hai lĩnh vực chính của lọc cộng tác là các phương pháp lân cận (the neighborhood methods) và các mô hình yếu tố tiềm ẩn (latent factor models)[5].

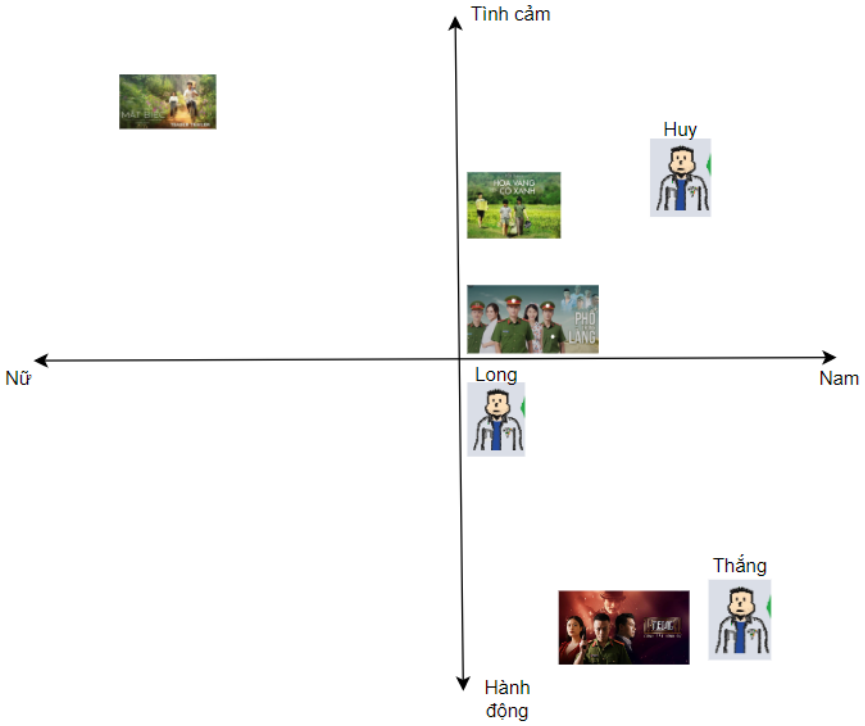
Phương pháp vùng lân cận tập trung vào việc tính toán các mối quan hệ giữa các items hoặc giữa users. Cách tiếp cận theo định hướng đánh giá sản phẩm dựa trên cách người dùng đánh giá của các "sản phẩm hàng xóm". Sản phẩm hàng xóm là các sản phẩm khác có xu hướng nhận được đánh giá tương tự khi được đánh giá bởi cùng một người dùng[5]. Ví dụ: nhìn Hình 3, ta thấy Thắng thích 3 bộ phim, 3 người khác (Long, Đức, Huy) cũng thích 3 bộ phim đó. Trong khi Long, Đức, Huy đều thích một bộ phim giống nhau là Mắt biếc. Trong đó Long, Đức thích phim Em và Trịnh và chỉ có bộ phim Hoa vàng cỏ xanh là được Huy thích. Do đó hệ thống sẽ gợi ý cho Thắng bộ phim Mắt biếc đầu tiên, sau đó là phim Em và Trịnh, cuối cùng là phim Hoa vàng cỏ xanh.



Hình 3: Phương pháp vùng lân cận

Mô hình yếu tố tiềm ẩn là một phương pháp thay thế, đưa ra các đánh giá bằng cách mô tả cả sản phẩm lẫn người dùng theo 20 đến 100 yếu tố được suy ra từ các mẫu đánh giá khác. Trong một khía cạnh nào đó, các yếu tố như vậy bao gồm một sự mô hình hóa bằng máy tính đối với các đặc tính bài hát đã được đề cập ở trên. Đối với phim ảnh, các yếu tố được phát hiện có thể đo lường rõ ràng

như thiên về hài kịch so với bi kịch; phim hành động dành cho người lớn hay dành cho trẻ em; các yếu tố ít được xác định rõ hơn như nhân vật có chiều sâu hay nhân vật độc đáo; hoặc các yếu tố khó giải thích được khác. Đối với người dùng, mỗi yếu tố quyết định mức độ người dùng thích phim như nào để cho điểm số cao trên hệ số phim tương ứng[5].



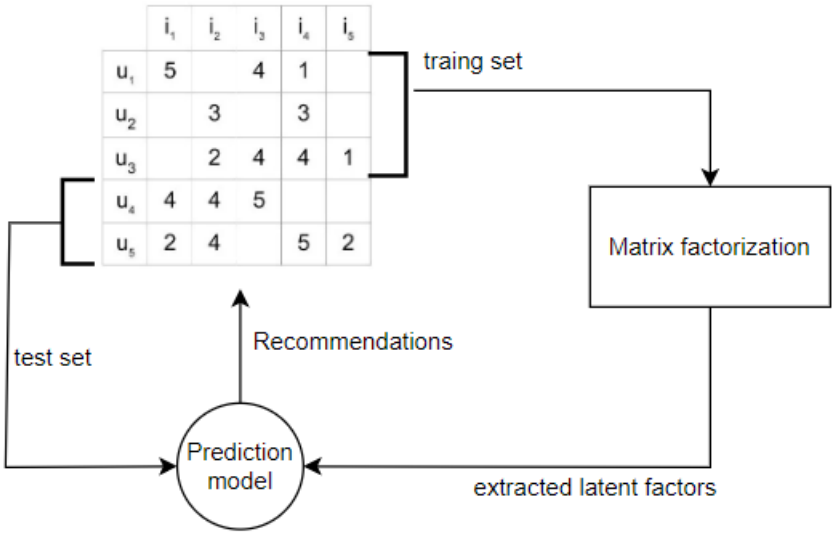
Hình 4: Đồ thị đã mô hình hóa cả người dùng lẫn phim bằng 2 yếu tố: giới tính nam - nữ và tình cảm - hành động

.

Hình 4 cho thấy một số bộ phim nổi tiếng và một số người dùng minh họa có thể thuộc vào hai chiều này. Đối với mô hình này, xếp hạng được dự đoán của người dùng cho một bộ phim, liên quan đến xếp hạng trung bình của phim, sẽ bằng sản phẩm dấu chấm của vị trí của phim và của người dùng trên biểu đồ. Ví dụ, Thắng là nam, rất thích xem những bộ phim hành động, do đó được cho là phù hợp với bộ phim Sinh tử (mang đặc tính của phim hành động), nhưng lại không thích bộ phim Mắt biếc (mang đặc tính phim tình cảm). Long và phim Phố trong làng được đánh giá trung lập do mang đặc tính của cả 2 chiều. Ở đây em sẽ sử dụng kỹ thuật embedding để mô tả đặc tính user và item qua một không gian K chiều thay vì đơn giản chỉ là 2 chiều như trên hình.

CHƯƠNG 3: PHƯƠNG PHÁP PHÂN RÃ MA TRẬN

Một trong những ứng dụng thành công nhất của mô hình yếu tố tiềm ẩn là dựa trên phương pháp phân rã ma trận (Matrix Factorization). Nói theo cách đơn giản, phương pháp này mô tả item và user bởi vector của các yếu tố được suy ra từ việc đánh giá item. Tính tương thích cao giữa nhân tố item và user sẽ được hệ thống gợi ý cho user cái item đó. Những phương pháp này đã trở nên phổ biến trong những năm gần đây bằng cách kết hợp khả năng mở rộng tốt với độ chính xác dự đoán thêm vào đó thì phương pháp này có tính đời thực khá cao[5].

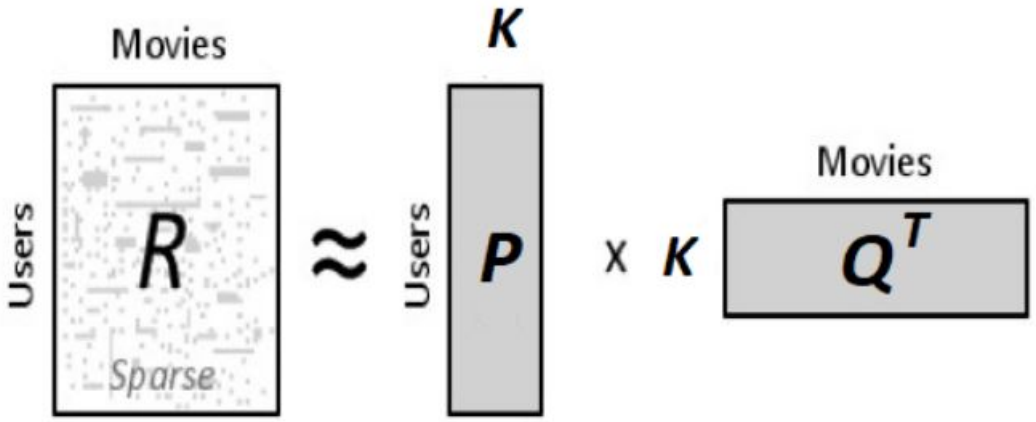


Hình 5: Kiến trúc của RS áp dụng phương pháp MF.

3.1 Phát biểu bài toán

Cho một ma trận thưa $R \in R^{m \times n}$, nhiệm vụ của phương pháp MF là chia ma trận **R** này thành hai ma trận có kích thước nhỏ hơn $P \in R^{m \times K}$ và $Q \in R^{n \times K}$ (Hình 6), sao cho có thể xây dựng lại **R** từ hai ma trận nhỏ này có độ xấp xỉ càng nhỏ càng tốt[5].

Do đó đầu vào của bài toán này là ma trận thưa **R**; đầu ra là hai ma trận **P** và ma trận **Q**.



Hình 6: Ma trận R được xấp xỉ bằng tích của ma trận P và Q^T.

3.2 Dạng cơ bản của phân rã ma trận

Em sẽ biểu diễn ba ma trận trên bằng toán học như sau:

$$R \approx P \times Q^T$$

hay

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mk} \end{bmatrix} \times \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ q_{k1} & q_{k2} & \cdots & q_{kn} \end{bmatrix}^T$$

Mô hình phân rã ma trận ánh xạ users và item vào trong 1 chiều không gian tiềm ẩn chung của không gian K chiều, nhờ vậy mối quan hệ giữa người dùng - sản phẩm được mô hình hóa thành một phần tử thuộc không gian đấy. Mỗi item i được liên kết với một vector $q_i \in R^K$, chính là hàng của ma trận Q bao gồm K nhân tố tiềm ẩn mô tả item i. Và mỗi user được liên kết với một vector $p_u \in R^K$, chính là hàng của ma trận P bao gồm K nhân tố tiềm ẩn mô tả user u. Ta có r_{ui} là đánh giá thật của user p_u dành cho item q_i, tích giữa $q_i * p_u$ thể hiện đánh giá tổng thể mối quan hệ giữa người dùng về item cụ thể và xấp xỉ r_{ui}, điều này dẫn đến công thức sau[5]:

$$\hat{r}_{ui} = p_u {q_i}^T = \sum_{k=1}^{K} p_{uk} q_{ik} \tag{1}$$

Hai tham số p_u , q_i tính được bằng cách sử dụng những thuật toán tối ưu như SGD (Stochastic Gradient Descent)[**Mavridis**, 5] hoặc ADAM để tối thiểu hóa hóa hàm mất mát sau:

$$\min \sum_{(u,i)\in D^{train}} (r_{ui} - \hat{r})^2 = \sum_{(u,i)\in D^{train}} (r_{ui} - p_u q_i^T)^2 = \sum_{(u,i)\in D^{train}} (r_{ui} - \sum_{k=1}^{K} p_{uk} q_{ik})^2 \quad (2)$$

D là tập những đánh giá đã biết trong ma trận **R** (tập training).

Chính tắc hóa (Regularization): Để ngăn chặn sự quá khớp của mô hình (overfitting), người ta sẽ thêm vào hàm mất mát một đại lượng gọi là chính tắc hóa để điều khiển độ lớn của các giá trị p_u và q_i[5]. Do đó hàm mất mát bây giờ trở thành:

$$\min \sum_{(u,i)\in D^{train}} (r_{ui} - p_u q_i^T)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \quad (3)$$

Trong đó λ là hệ số chính tắc hóa ($0 \leqslant \lambda \leqslant 1$ và $\|\|^2$)là chuẩn Frobenius của ma trận[1].

3.3 Phân rã ma trận với dự đoán cơ sở

Một lợi thế của việc sử sụng MF là nó linh hoạt trong việc xử lý nhiều khía cạnh khác nhau của dữ liệu. Như trong công thức (1) giá trị ratings cuối cùng chỉ chỉ ra được mối liên hệ giữa users và items, nhưng trong thực tế các đánh giá đều có những thiên lệch về users và items, đó được gọi là bias hay độ lệch. Ví dụ như có user dễ và khó tính, cũng có những item được đánh giá cao hơn những item khác chỉ vì user thấy các users khác đã đánh giá item đó cao rồi.

Do đó sẽ không chính xác nếu chỉ sử dụng công thức (1) để giải thích đầy đủ giá trị rating. Thay vào đó, hệ thống cố xác định từng phần của những giá trị đó mà độ lệch độc lập của user và item có thể giải thích, chỉ đưa phần tương tác thực sự của dữ liệu vào mô hình - phương thức này còn được gọi là dự đoán cơ sở (Baseline Predictor)[5].

$$b_{ui} = \mu + b_i + b_u \quad (4)$$

- μ : điểm rating trung bình.
- b_u, b_i đại diện cho độ lệch quan sát được của user u và item i.

Ví dụ như giả sử tính rating của user Thắng cho bộ phim Mắt biếc. Điểm trung bình của bộ phim là 3.7, tuy nhiên bộ phim này là được đánh giá cao hơn điểm

[1]Chuẩn Frobenius là căn bậc hai của tổng bình phương tất cả các phần tử của ma trận đó.

trung bình là 0.5, Thắng là 1 người xem khó tính luôn đánh giá phim thấp hơn 0.3 so với mức điểm trung bình. Do đó, xấp xỉ cho bộ phim Mắt biếc được đánh giá bởi Thắng là 3.7+0.5-0.3 = 3.9
Ta thêm biases vào phương trình 1 như sau:

$$\hat{r}_{ui} = \mu + b_i + b_u + p_u q_i^T \tag{5}$$

Mô hình sẽ đánh giá được rating bằng cách tối ưu hàm mất mất mát sau:

$$\min \sum_{(u,i)\in K} \left(r_{ui} - p_u q_i^T - \mu - b_u - b_i\right)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_u^2 + b_i^2) \tag{6}$$

3.4 Kỹ thuật embedding

Embedding là một kỹ thuật đưa một vector có số chiều lớn, thường ở dạng thưa, về một vector có số chiều nhỏ, thường ở dạng dày đặc. Phương pháp này đặc biệt hữu ích với những đặc trưng hạng mục có số phần tử lớn ở đó phương pháp chủ yếu để biểu diễn mỗi giá trị thường là một vector one-zero[2]. Một cách lý tưởng, các giá trị có ý nghĩa tương tự nhau nằm gần nhau trong không gian embedding[4]. Ở trong đề tài này thì việc chọn ra K nhân tố ẩn chính là việc đi Embedding vector user (item) theo không gian K chiều như Hình 4 (user và item được embedding theo không gian 2 chiều từ đó sẽ lấy được các tọa độ của user (item) tạo nên các ma trận Q,P).

Có 3 cách để tính độ tương tự giữa hai vecto embedding: Khoảng cách Euclid, tính tích vô hướng và tương tự cosine.
Khoảng cách Eculid

$$d(e_1, e_2) = \|e_1 - e_2\| = \sqrt{\|e_1\|^2 + \|e_2\|^2 - 2e_1^T e_2}$$

Khoảng cách này không âm và càng nhỏ thì hai vector embedding càng gần nhau.
Tích vô hướng

$$\langle e_1, e_2 \rangle = e_1^T e$$

Tích vô hướng giữa hai vector càng cao thể hiện các embedding càng giống nhau. Giá trị này lớn nếu góc giữa hai vector nhỏ và các vector này có độ dài lớn.
Tương tự cosine

$$\cos(e_1, e_2) = \frac{e_1^T e}{\|e_1\| \,.\, \|e_2\|}$$

Góc giữa hai vector càng nhỏ thì độ tương tự cosin càng cao. Độ tương tự cosin nhỏ nhất bằng -1 nếu hai vector này trái dấu nhau.

[2]Những vector chỉ biểu diễn bởi 2 giá trị 1 và 0

Với đề tài này thì các vector e_1, e_2 tương tự như là các vector p_u và các vector q_i.

3.5 Thuật toán được dùng trong đề tài

Thuật toán ADAM

Thuật toán tối ưu Adam được xem như là sự kết hợp của hai thuật toán tối ưu khác đó chính là thuật toán mở rộng bình phương trung bình (RMSProp) và thuật toán hướng giảm Gradient ngẫu nhiên với Momentum (SGD with Momentum)[3]. Thuật toán sử dụng hai internal states momentum (m) và squared momentum (v) của gradient cho các tham số. Sau mỗi lần huấn luyện, giá trị của m và v được cập nhật lại sử dụng trung bình trượt số mũ (Exponential moving average-EMA). Cập nhật v và m được tiến hành như sau[2]:

$$m_i = \beta_1 m_{i-1} + (1 - \beta_1) g_i$$
$$v_i = \beta_2 v_{i-1} + (1 - \beta_2) g_i^2$$

Trong đó, β được coi là siêu tham số, g là gradient tại bước i. Công thức cập nhật vị trí của nghiệm là:

$$\Theta_{i+1} = \Theta_i - \eta \frac{m_t}{\sqrt{v_t} + \varepsilon}$$

η là learning rate , ε là giá trị được thêm vào để ngăn việc chia cho 0.
Để việc giảm được thực hiện nhanh hơn, thuật toán đã sử dụng hai kỹ thuật:

- Tính EMA của giá trị đạo hàm lưu vào biến m và sử dụng nó là tử số của việc cập nhật hướng. Với ý nghĩa là nếu m có giá trị lớn, thì việc giảm đang đi đúng hướng và chúng ta cần bước nhảy lớn hơn để đi nhanh hơn. Tương tự, nếu giá trị m nhỏ, phần giảm có thể không đi về hướng tối tiểu và chúng ta nên đi 1 bước nhỏ để thăm dò. Đây là phần momentum của thuật toán.
- Tính EMA của bình phương giá trị đạo hàm lưu vào biến v và sử dụng nó là phần mẫu số của việc cập nhật hướng. Với ý nghĩa như sau: Giả sử gradient mang các giá trị dương, âm lẫn lộn, thì khi cộng các giá trị lại theo công thức tính m ta sẽ được giá trị m gần số 0. Do âm dương lẫn lộn nên nó bị triệt tiêu lẫn nhau. Nhưng trong trường hợp này thì v sẽ mang giá trị lớn. Do đó, trong trường hợp này, chúng ta sẽ không hướng tới cực tiểu, chúng ta sẽ không muốn đi theo hướng đạo hàm trong trường hợp này. Chúng ta để v ở phần mẫu vì khi chia cho một giá trị cao, giá trị của các phần cập nhật sẽ nhỏ, và khi v có giá trị thấp, phần cập nhật sẽ lớn. Đây chính là phần tối ưu RMSProp của thuật toán.

[3]2 thuật toán này có thể tìm hiểu thêm ở blog machine learning cơ bản Vũ Khắc Tiệp

Ở đây, m được xem như là moment thứ nhất, v xem như là moment thứ hai, nên thuật toán có tên là "Adaptive moment estimation".

Trong đó Θ chính là các vecto p_u và q_i, tức là ta sẽ tối ưu hàm f theo 2 biến đó chính là p_{uh} và q_{hi} và hàm $f(p,q)$ có dạng như sau:

$$f(p,q) = \sum_{(u,i)\in D^{train}} \left(r_{ui} - p_u q_i^T\right)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{7}$$

suy ra em sẽ đạo hàm f theo 2 giá trị p_{uh} và q_{hi} :

$$\frac{\partial f}{\partial p_{uh}} = \sum_{i:(u,i)\in R} (r_{ui} - \sum_{s=1}^{k} p_{us}q_{si})(-q_{hi}) + \lambda p_{uh} \;\; \forall u \in \{1..m\}, h \in \{1..k\} \tag{8}$$

$$\frac{\partial f}{\partial q_{hi}} = \sum_{i:(u,i)\in R} (r_{ui} - \sum_{s=1}^{k} p_{us}q_{si})(-p_{ih}) + \lambda q_{hi} \;\; \forall i \in \{1..n\}, h \in \{1..k\} \tag{9}$$

Ta có thuật toán ADAM như sau:

Algorithm 1 Thuật toán tối ưu ADAM

$\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.99, \eta = 10^{-8}$ (Tham số mặc định)
$m_0 \leftarrow 0$ // Khởi tạo vecto moment thứ nhất
$v_0 \leftarrow 0$ // Khởi tạo vecto moment thứ hai
$i \leftarrow 0$ // Khởi tạo tham số đếm
while Θ_i chưa hội tụ **do**
    $i \leftarrow i + 1$
    $g_i = \nabla_\Theta f_i(\Theta_{i-1})$ // Tượng tự như công thức (8) và (9)
    $m_i = \beta_1.m_{i-1} + (1-\beta_1).g_i$ // Cập nhật cho vecto m
    $v_i = \beta_2.v_{i-1} + (1-\beta_2).g_i^2$ // Cập nhật cho vecto v
    $\hat{m}_i \leftarrow m_i/(1-\beta_1)$ // Tính bias-corrected cho vecto m
    $\hat{v}_i \leftarrow v_i/(1-\beta_2)$ // Tính bias-corrected cho vecto v
    $\Theta_i \leftarrow \Theta_{i-1} - \eta.\hat{m}_i/(\sqrt{\hat{v}_i} + \varepsilon)$ // Cập nhật tham số cần tối ưu
end while

3.6 Phương pháp đánh giá hiệu quả

Có nhiều phương pháp khác nhau có thể được sử dụng để đánh giá giải thuật như: F-Measure, Area Under the ROC Curve (AUC),... mỗi phương pháp đánh giá sẽ thích hợp cho từng bài toán cụ thể. Trong đề tài này, em sử dụng phương pháp sai số bình phương trung bình (RMSE) để đánh giá chất lượng

của mô hình. Công thức của phương pháp RMSE[3]:

$$RMSE = \sqrt{\frac{1}{|D^{test}|}} \sum_{(u,i) \in D^{test}} (r_{ui} - \hat{r}_{ui})^2$$

CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ

4.1 Khái quát dữ liệu

Ở đề tài này em sử dụng bộ dữ liệu Movielen 1m. Bộ dữ liệu này bao gồm 3 file : RATINGS FILE DESCRIPTION, USERS FILE DESCRIPTION, MOVIES FILE DESCRIPTION. Chứa tất cả 1000209 đánh giá của gần 3900 bộ phim được đánh giá từ 6040 người dùng.

RATINGS FILE DESCRIPTION: UserID::MovieID::Rating::Timestamp

- UserIDs nằm trong khoảng 1 and 6040
- MovieIDs nằm trong khoảng 1 and 3952
- Ratings đánh theo thang điểm từ 0 đến 5
- Timestamp được thể hiện bằng giây
- Mỗi user sẽ đánh giá ít nhất 20 đánh giá

	UserID	MovieID	Rating	Timestamp
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291
5	1	1197	3	978302268
6	1	1287	5	978302039
7	1	2804	5	978300719
8	1	594	4	978302268
9	1	919	4	978301368

Hình 7: Bảng rating.

USERS FILE DESCRIPTION:UserID::Gender::Age::Occupation::Zip-code

- Giới tính: "M" cho Nam and "F" cho Nữ.
- Tuổi:
  - 1: "Under 18"
  - 18: "18-24"
  - 25: "25-34"

  - 35: "35-44"
  - 45: "45-49"
  - 50: "50-55"
  - 56: "56+"
- Nghề nghiệp:
  - 0: "other" or not specified
  - 1: "academic/educator"
  - 2: "artist"
  - 3: "clerical/admin"
  - 4: "college/grad student"
  - 5: "customer service"
  - 6: "doctor/health care"
  - 7: "executive/managerial"
  - 8: "farmer"
  - 9: "homemaker"
  - 10: "K-12 student"
  - 11: "lawyer"
  - 12: "programmer"
  - 13: "retired"
  - 14: "sales/marketing"
  - 15: "scientist"
  - 16: "self-employed"
  - 17: "technician/engineer"
  - 18: "tradesman/craftsman"
  - 19: "unemployed"
  - 20: "writer"

	UserID	Gender	Age	Occupation	Zip-code
0	1	F	1	10	48067
1	2	M	56	16	70072
2	3	M	25	15	55117
3	4	M	45	7	02460
4	5	M	25	20	55455
5	6	F	50	9	55117
6	7	M	35	1	06810
7	8	M	25	12	11413
8	9	M	25	17	61614
9	10	F	35	1	95370

Hình 8: Bảng movie.

MOVIES FILE DESCRIPTION: MovieID::Title::Genres

- Tên của bộ phim sẽ được kèm theo năm phát hành.
- Có các thể loại phim như sau: Action Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western.

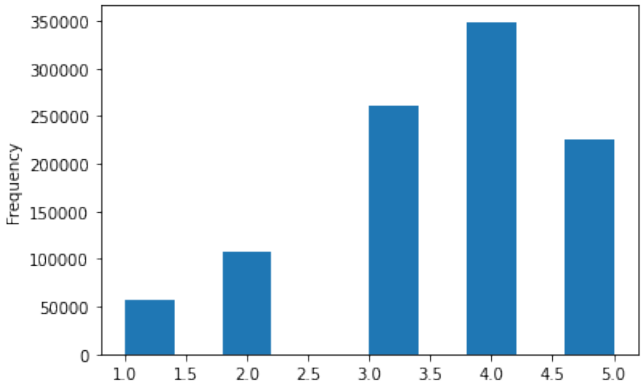
	MovieID	Title	Genres
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action|Crime|Thriller
6	7	Sabrina (1995)	Comedy|Romance
7	8	Tom and Huck (1995)	Adventure|Children's
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action|Adventure|Thriller

Hình 9: Bảng movie.

4.2 Khám phá dữ liệu

RATING

Ta có thể thấy rằng 4 sao là rating nhiều nhất của bộ dữ liệu này. Điều đó chứng tỏ rằng nếu user thích bộ phim này thì họ mới đánh giá bộ phim vì 1 và 2 sao sẽ được đánh giá rất ít.



Số lượng người đánh giá bộ phim nhiều nhất là 2314 đánh giá của user 4169 và ít nhất là 20 đánh giá. Chứng tỏ là tác giả đã lọc những user mà có ít hơn 20 đánh giá, tuy nhiên thì phần lớn user sẽ nằm ở phần ít đánh giá hoặc sẽ không có đánh giá nào.

```
[5] ratings["UserID"].value_counts()

4169    2314
1680    1850
4277    1743
1941    1595
1181    1521
        ...
5725      20
3407      20
1664      20
4419      20
3021      20
Name: UserID, Length: 6040, dtype: int64
```

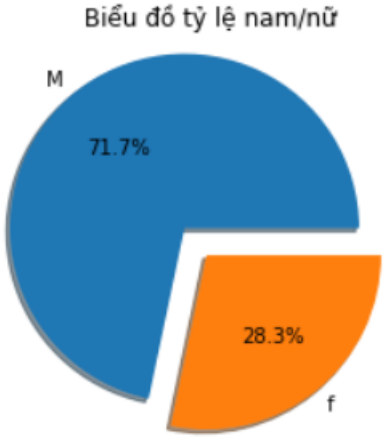
Ở khía cạnh bộ phim thì phim **American Beauty 1999** được đánh giá nhiều nhất với 3428 lần đánh giá, tiếp theo là series phim **Star Wars**. Ở đây thì em cũng thống kê luôn ra có khoảng 2991 bộ phim chỉ được đánh giá 1 lần cho thấy sự thưa thớt của bộ dữ liệu

```
movies = df_dict["movies"]
movies_rating = ratings.merge(movies)
print(movies_rating["Title"].value_counts())
print("\n")
print(" How many movies was rated 1 star ? ")
print(movies_rating["Title"].value_counts()[1])
```

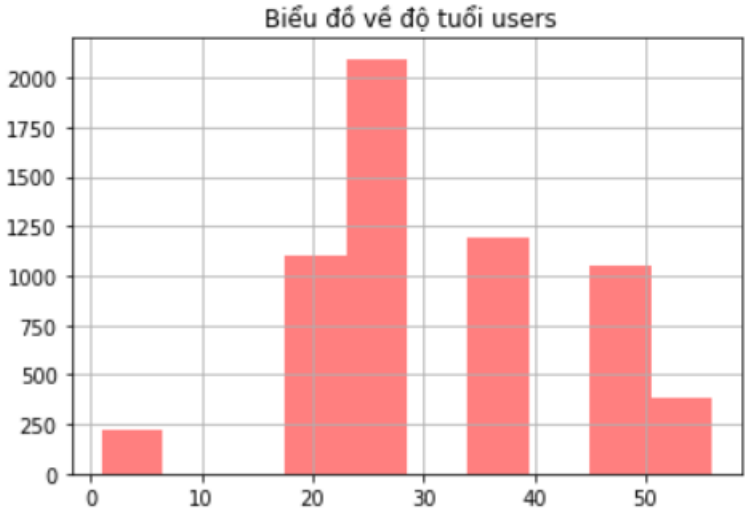
```
American Beauty (1999)                                3428
Star Wars: Episode IV - A New Hope (1977)             2991
Star Wars: Episode V - The Empire Strikes Back (1980) 2990
Star Wars: Episode VI - Return of the Jedi (1983)     2883
Jurassic Park (1993)                                  2672
                                                      ...
Blood and Sand (Sangre y Arena) (1989)                   1
Ring, The (1927)                                         1
Eden (1997)                                              1
Frank and Ollie (1995)                                   1
Five Wives, Three Secretaries and Me (1998)              1
Name: Title, Length: 3706, dtype: int64


 How many movies was rated 1 star ?
2991
```

USERS

Dữ liệu ở bảng này sẽ bao gồm các thông tin như giới tính tuổi và nghề nghiệp, mã vùng. Ở biểu đồ dưới em đã thống kê giới tính của user và thấy có khoảng 4331 là đàn ông chiếm tới 71,7% còn phụ nữ chỉ chiếm khoảng 1709 người tương đương với 28,3%. Và độ tuổi của User cũng chủ yếu từ 18 đến 28, tuy nhiên tỷ lệ những người già trên 50 tuổi xem cũng là tương đối cao.

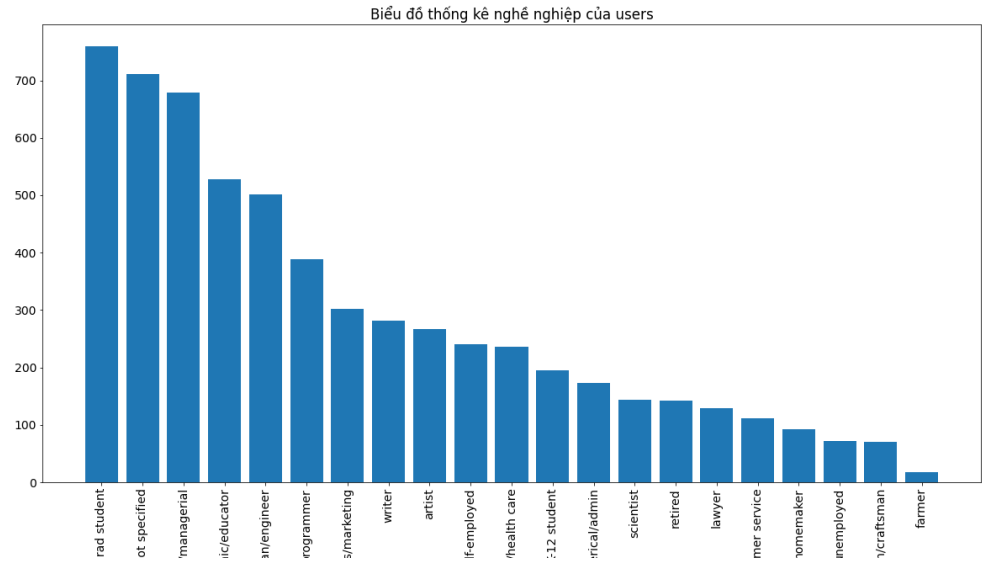


(a)



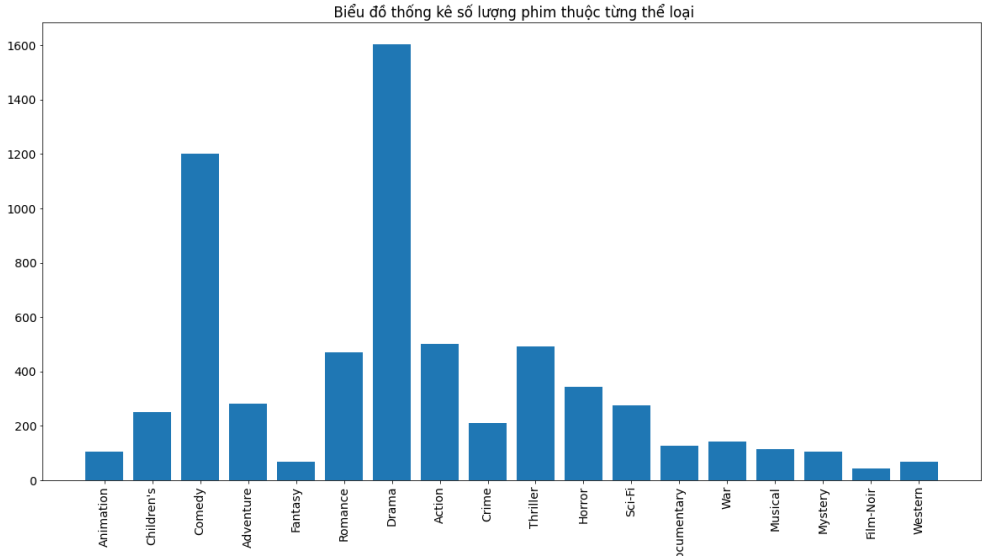
(b)

Không có gì bất ngờ, các bạn sinh viên xuất hiện nhiều trong bộ dữ liệu nhất còn các bác nông dân xuất hiện ít nhất.



MOVIES

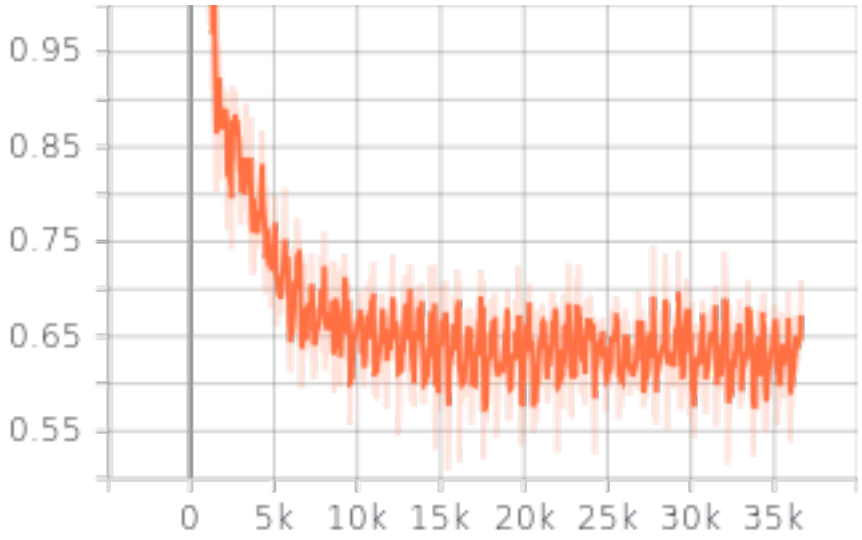
Trả lời cho câu hỏi thứ hai, ta thấy rằng thể loại Drama và Comedy có nhiều bộ phim nhất. Các thể loại Animation, Fantasy, Documentary, War, Mystery, Film-Noir và Western có ít bộ phim nhất với khoảng từ 50 đến 100 bộ phim.



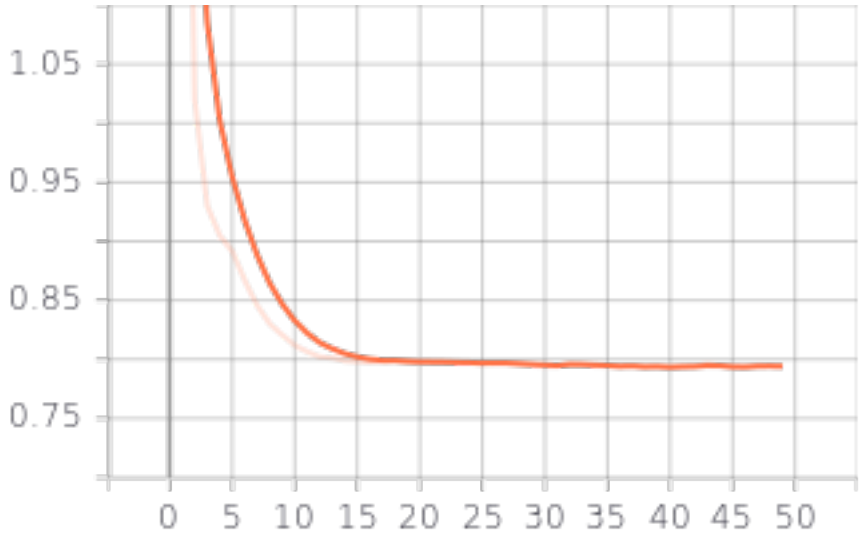
4.3 Kết quả

Sau khi huấn luyện mô hình cho 50 epochs[4] với dữ liệu được chia thành 75/25 cho phần train/test, em đã được RMSE của hàm mất mát trên tập training là RMSE = 0.7088 (Hình 11), và RMSE trên tập test là RMSE = 0.7939 (Hình 12).

[4]Mỗi lần duyệt qua tập dữ liệu thì được gọi là một epoch



Hình 10: RMSE trên tập training biến động theo mỗi vòng lặp nhưng nhìn chung vẫn giảm và cũng có xu hướng hội tụ



Hình 11: Tập test giảm về 0.7939 sau 50 epochs

CHƯƠNG 5: TỔNG KẾT

Kết luận

Phương pháp phân rã ma trận (Matrix Factorization) là một phương pháp khá đơn giản, chi phí dự đoán thấp khi chỉ cần tính tích vô hướng của 2 vecto và đặc biệt hơn khi nó khai thác được sự giống nhau giữa user - user , item - item , user - item một cách hiệu quả. Tuy nhiên cũng có những nhược điểm của phương pháp này đó là nó không sử dụng đến hồ sơ user nên với vấn đề khởi đầu lạnh (cold start problem) thì hệ thống dự đoán có thể không chính xác.

Kết quả đồ án

- Đồ án đã cho thấy được cách mà cách hệ thống gợi ý hoạt động tuy chỉ hình dung mức cơ bản vì thực tế hệ thống này còn phức tạp hơn nhiều.

Kỹ năng đạt được

- Bước đầu biết tìm kiếm, đọc, dịch tài liệu chuyên ngành liên quan đến nội dung đồ án.
- Biết tổng hợp các kiến thức đã học và kiến thức trong tài liệu tham khảo để viết báo cáo đồ án.
- Chế bản đồ án bằng LaTeX.

Hướng phát triển của đồ án trong tương lai

Cải tiến mô hình thành mô hình động bằng cách thêm vào các yếu tố thay đổi theo thời gian. Từ đó có thể tích hợp mô hình vào trong một ứng dụng cụ thể.

References

[1] Bobadilla J., Ortega F., Hernando A., Gutiérrez H. "Recommender systems survey". In: *Knowledge-Based Systems* 46 (July.2013), pp. 109–132.

[2] Jimmy Ba Diederik P. Kingma. "Adam: A Method for Stochastic Optimization". In: *Published as a conference paper at the 3rd International Conference for Learning Representations* 322.10 (1905), pp. 891–921.

[3] Nguyen Thai Nghe. "AN INTRODUCTION TO FACTORIZATION TECHNIQUE FOR BUILDING RECOMMENDATION SYSTEMS". In: *Natural Sciences and Technology* 3.10 (2013), pp. 44–53.

[4] Vu Khac Tiep. "Machine Learning cho dữ liệu dạng bảng". In: Vu Khac Tiep, 2021. Chap. 3.1.

[5] R.Bell Y.Koren and C. Volinsky. "Matrix Factorization Techniques for Recommender Systems". In: *In Computer* 42 (Aug.2009), pp. 30–37.