HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN HẢI ANH

NGHIÊN CỨU VÀ ỨNG DỤNG KỸ THUẬT HỌC SÂU CHO HỆ TƯ VẤN

Chuyên ngành: Khoa Học Máy Tính

Mã số: 8.48.01.018

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

HÀ NỘI - NĂM 2024

Đề án tốt nghiệp được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: TS. Nguyễn Duy Phương

Phản biện 1: ……..…….……

Phản biện 2: ……..……..…….……..

Đề án tốt nghiệp sẽ được bảo vệ trước Hội đồng chấm đề án tốt nghiệp thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng ..... năm .....

Có thể tìm hiểu đề án tốt nghiệp tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Trong cuộc sống ngày nay, chúng ta gặp phải vô vàn tình huống phải đưa ra quyết định. Buổi sáng nên mặc gì cho phù hợp? Lựa chọn thực đơn nào cho gia đình? Nhiệm vụ nào chúng ta nên thực hiện đầu tiên? Nên đăng ký học ở ngôi trường nào? Chúng ta phải trả lời hàng nghìn câu hỏi quan trọng này hàng ngày. Chính vì thế mà hàng loạt các hệ tư vấn đã xuất hiện để đáp ứng nhu cầu này. Hệ tư vấn (Recommender System) là một hệ thống lọc thông tin dùng để dự đoán đánh giá và sở thích của một người dùng về các sản phẩm, từ đó hệ thống có thể đưa ra những tư vấn gợi ý sao cho phù hợp.

Với lý do trên, học viên đã quyết định lựa chọn đề tài "**Nghiên cứu và ứng dụng kỹ thuật học sâu cho hệ tư vấn**" để thực hiện đề án tốt nghiệp thạc sĩ.

2. Đặt vấn đề

Khi người dùng truy cập vào một nền tảng xem phim nào đó thì vấn đề được đặt ra là: "Làm thế nào để nền tảng đó có thể gợi ý cho người dùng những bộ phim mà họ sẽ yêu thích?". Và câu trả lời chính là cần phải xây dựng được một hệ tư vấn đề xuất các bộ phim hiệu quả cho người dùng.

Một hệ tư vấn tốt ảnh hưởng rất lớn đến sự thành bại của các nền tảng và mỗi hệ thống cần tinh chỉnh một hệ tư vấn sao cho phù hợp với dữ liệu mà nền tảng thu thập được. Và trong thực tế, hầu hết các hệ tư vấn đều có thể đạt kết quả rất tốt nếu như sở hữu đủ dữ liệu nhưng sẽ là kém hiệu quả nếu dữ liệu quá ít, điều này khiến những nền tảng vừa và nhỏ sẽ không thể nào tận dụng được những ích lợi mà hệ tư vấn đem lại. Bài toán này là một trong những mục tiêu cần được giải quyết hàng đầu mà nhiều phòng nghiên cứu trên khắp thế giới đang thực hiện.

3. Mục tiêu đề ra

Ngày nay có rất nhiều công trình nghiên cứu về các hệ tư vấn cho người dùng. Nhiều mô hình mới, đa dạng được áp dụng vào thực tế và chất lượng của các mô hình này cũng ngày càng được cải thiện theo thời gian. Tuy nhiên, những phương pháp khác nhau đưa lại những ưu nhược điểm khác nhau. Trong đề án này, học viên sẽ đưa ra hai mục tiêu sau:

1. Nghiên cứu các phương pháp phổ biến đã được xây dựng trước đây và thực hiện cài đặt.
2. Xây dựng mô hình mạng học sâu kết hợp với đồ thị và giải thuật K-means, tiến hành cài đặt và so sánh hiệu suất với các phương pháp phổ biến.

4. Đối tượng và phạm vi nghiên cứu

Trong đề án này, ngoài việc trình bày cơ sở lý thuyết về hệ tư vấn và các phương pháp học máy truyền thống như đề xuất dựa trên nội dung, lọc cộng tác dựa trên người dùng, lọc cộng tác dựa trên sản phẩm kèm với đó các kỹ thuật phổ biến khác như Matrix Factorization, Singular Value Decomposition (SVD), Autoencoder. Đề án sẽ đi sâu về kỹ thuật đồ thị (Graph-Based) kết hợp với Autoencoder và thuật toán phân cụm K-means để xây dựng mô hình GHRS [21].

5. Phương pháp nghiên cứu

Trong quá trình nghiên cứu và thực nghiệm, học viên sẽ kết hợp các công cụ của giải tích, giải thuật phân cụm, lý thuyết đồ thị và kiến trúc mạng cho các phương pháp xây dựng hệ tư vấn kèm với các thư viện của python cho quá trình viết mã.

6. Bố cục của báo cáo

Báo cáo được chia thành ba chương, trong đó:

Chương 1: Tổng quan về hệ tư vấn

Nội dung chính của chương này là trình bày những nghiên cứu cơ bản về hệ tư vấn, các phương pháp tiếp cận phổ biến nhất hiện nay. Trên cơ sở đó trình bày cụ thể một số phương pháp phổ biến hiện nay để có cái nhìn tổng quan khi so sánh với phương pháp được trình bày tại chương 2.

Chương 2: Mô hình dựa trên đồ thị và học sâu

Trình bày cụ thể phương pháp xây dựng mô hình GHRS cũng như cơ sở thực nghiệm sẽ được sử dụng cho việc cài đặt các phương pháp đã trình bày ở cả chương 1 và chương 2.

Chương 3: Kết quả thực nghiệm

Trên cùng một môi trường và tập thử nghiệm, so sánh đầu ra của từng phương pháp kết hợp với kiểm định RMSE và lập bảng so sánh.

Cuối cùng là kết luận và hướng nghiên cứu tiếp theo.

CHƯƠNG I: TỔNG QUAN VỀ HỆ TƯ VẤN

1.1. Khái niệm hệ tư vấn

Hệ tư vấn (hệ thống gợi ý hay còn gọi là hệ thống khuyên dùng), tiếng anh là Recommender System hoặc Recommendation System, là một lớp con của hệ thống lọc thông tin, tìm kiếm dự đoán "đánh giá" hoặc "ưa thích" của người dùng với một sản phẩm hoặc đối tượng nào đó. Dựa theo Ricci và cộng sự [2], Hệ tư vấn là các công cụ và kỹ thuật phần mềm cung cấp đề xuất các đối tượng có thể hữu ích với người dùng. Những đề xuất liên quan đến quyết định của người dùng như: cuốn sách nào nên đọc, bộ phim nào đáng xem, bài hát nào nên nghe hay tin tức nào nên đọc tiếp theo…

1.2. Các lĩnh vực ứng dụng của hệ tư vấn

System	Recommended Items
Amazon.com	Books and other products
Netflix	Streaming Videos
GroupLens	News
GoogleNews	News
Youtube	Online videos
TripAdvisor	Travel products
IMDB	Movies

Hình 1.1: Các hệ thống thực tế của một số nền tảng

Hình 1.2 đưa ra một số ứng dụng phổ biến của hệ tư vấn và mục tiêu của chúng. Nhiều mục tiêu trong số này đều thuộc lĩnh vực thương mại điện tử. Tuy nhiên, hệ tư vấn đã phát triển xa hơn chỉ là trong lĩnh vực gợi ý sản phẩm cụ thể. Để thúc đẩy sự phát triển của mạng xã hội, các nền tảng mạng xã hội trực tuyến thường đề xuất các liên kết với khách hàng của họ.

1.3. Phát biểu bài toán cho hệ tư vấn

Trước khi trình bày về các quy trình và hướng tiếp cận, cần làm rõ 2 thuật ngữ sẽ được sử dụng: Người dùng (user) và sản phẩm (item). Thứ nhất, khái niệm người dùng ở đây là người sử dụng hệ thống để thực hiện các thao tác xem, đánh giá, bình luận, … Thứ hai, khái niệm sản phẩm là mặt hàng như các video, bộ phim, bản nhạc, bài báo, … riêng trong đề án này thì item là các bộ phim. Trong hầu hết các hệ tư vấn, dữ liệu được cung cấp dưới dạng đánh giá của người dùng về sản phẩm.

1.4. Quy trình xây dựng hệ tư vấn

- **Bước 1: Thu thập dữ liệu**

Tại giai đoạn đầu tiên, những thông tin mà các hệ thống hay thu thập như: Sản phẩm (Item), Người dùng (User), Đánh giá (Rating).

- **Bước 2: Xây dựng mô hình**

Bước này có thể thực hiện bằng nhiều hướng khác nhau nhằm đánh giá mối liên hệ giữa các thông tin thu thập được ở Bước 1.

- **Bước 3: Đưa ra dự đoán**

Kết quả đầu ra của Bước 3 sẽ được dùng để dự đoán các đánh giá xếp loại của người dùng với sản phẩm chưa có đánh giá trước đó và chọn ra ***z*** sản phẩm mới phù hợp nhất đối với người dùng hiện thời để đưa ra gợi ý cho họ.

1.5. Các hướng tiếp cận xây dựng hệ tư vấn

Có nhiều cách phân loại các phương pháp xây dựng hệ tư vấn tùy theo quan điểm của mỗi nhà nghiên cứu. Dựa theo bài báo của Jiliang Tang và cộng sự [4] cùng với nhiều nghiên cứu khác sau này [3], việc phân nhóm được đưa ra có sự chồng chéo lẫn nhau nhưng tổng thể được gom lại thành một số loại như là :

1.5.1. Content-based Filtering

Các hệ tư vấn dựa trên nội dung bắt đầu từ việc nghiên cứu truy xuất thông tin và lọc thông tin [5]. Các hệ tư vấn này sẽ tư vấn các mục tương tự như mục mà người dùng đã thích trong quá khứ. Các hệ tư vấn dựa trên nội dung chủ yếu tập trung vào tư vấn các mục có thông tin văn bản như sách, phim và tài liệu. Nội dung trong các hệ thống này được mô tả bằng các sản phẩm và mức độ tin cậy của các sản phẩm đó đối với người dùng thường được đo bằng trọng số TF-IDF. Các phương pháp tiếp cận cho lọc theo nội dung được chia thành hai nhóm chính: Lọc nội dung dựa vào bộ nhớ (Memory-based) và Lọc nội dung dựa vào mô hình (Model-based).

1.5.2. Collaborative Filtering

Lọc cộng tác (CF) là một kỹ thuật phổ biến nhất để xây dựng hệ tư vấn, khai thác những khía cạnh liên quan đến thói quen sử dụng sản phẩm của

cộng đồn người dùng có cùng sở thích trong quá khứ để đưa ra dự đoán các sản phẩm phù hợp nhất. Giả định rằng nếu người dùng đã đồng tình với nhau trong quá khứ thì họ có nhiều khả năng sẽ đồng tình trong tương lai hơn là đồng tình với những người dùng thuộc nhóm khác. Các phương pháp tiếp cận cho CF nói chung cũng chia thành hai nhóm giống như lọc nội dung: CF dựa vào bộ nhớ và CF dựa vào mô hình.

1.5.3. Hybrid Filtering

Lọc kết hợp hay còn lại hệ thống lai là phương pháp kết hợp giữa lọc nội dung và lọc cộng tác nhằm tận dụng những ưu điểm của cả hai phương pháp này. Với lọc nội dung là việc khai thác các khía cạnh liên quan tới đặc điểm trong thông tin đi kèm với từng đối tượng mà không quan tâm tới những người dùng khác. Ngược lại, lọc cộng tác quan tâm đến thói quen người dùng của mỗi khách hàng và độ tương đồng của họ. Mỗi phương pháp đều có những ưu và nhược riêng đã thúc đẩy các nhà nghiên cứu tìm kiếm các phương pháp tận dụng được các ưu điểm đó.

1.5.4. Other Approaches

Ngoài các phương pháp được đề cập ở trong Phần 1.5.1, 1.5.2 và 1.5.3, còn có một số phương pháp khác được phát triển và đã đạt được nhiều kết quả khả quan như: Knowledge-based, Context-aware, Time-sensitive, Location-based, Social-based [7], Demography-based [8].

1.6. Phương pháp đánh giá hệ tư vấn

1.6.1. Mean squared error

Sai số bình phương trung bình (MSE).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1.1}$$

trong đó: n là tổng số mẫu trong tập kiểm tra; y_i là giá trị thực tế tại mẫu i; $\hat{y}_i$ là giá trị dự đoán tại mẫu i.

1.6.2. Root mean squared error

Căn bậc hai của sai số bình phương trung bình (RMSE) hay đơn giản chỉ là MSE lấy căn bậc hai.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (1.2)$$

1.7. Cơ sở lý thuyết cho các phương pháp phổ biến

1.7.1. Hệ tư vấn sử dụng lọc nội dung

Ý tưởng chính của phương pháp này là gợi ý sản phẩm mới căn cứ theo những sản phẩm mà người dùng đã thích trước đó trong quá khứ. Sự tương đồng giữa sản phẩm được gợi ý và sản phẩm đã được người dùng yêu thích trước đó không nhất thiết phải có mối tương quan trực tiếp mà là dựa trên thuộc tính của các sản phẩm đó. Không giống như các hệ thống CF tận dụng các đánh giá của những người dùng khác, các hệ thống lọc nội dung chủ yếu tập trung vào đánh giá xếp hạng của chính người dùng mục tiêu. Do đó, những người dùng khác có độ quan trọng thấp [6].

- Hệ thống này được xây dựng dựa trên 3 bước [10] chính như sau:
- **Bước 1: Xây dựng thông tin sản phẩm**
- **Bước 2: Xây dựng hồ sơ người dùng**
- **Bước 3: Tính giá trị chưa biết cho ma trận tiện ích**

1.7.2. Hệ tư vấn sử dụng lọc cộng tác

1.7.2.1. Lọc cộng tác theo người dùng

Thuật toán cốt lõi của User - CF là tìm những người dùng có hành vi đánh giá trong quá khứ tương tự với người dùng cần dự đoán và sử dụng đánh giá của những người dùng tương tự đó để dự đoán cái mà người dùng cần dự đoán sẽ thích. Việc cần làm là xác định độ tương tự (similarity) giữa hai người dùng. Giả sử thông tin duy nhất ta có là ma trận tiện ích ***Y*** mà không dùng dữ

liệu bên ngoài. Độ tương tự sẽ được xác định dựa trên các cột tương ứng của họ trong ma trận.

Trải qua nhiều thập niên nghiên cứu và phát triển, đã có rất nhiều công thức tính độ tương tự được đề xuất và một vài trong số đó đã được thử nghiệm thực tế và tổng hợp bởi Fethi Fkih [13]. Tuy nhiên, do giới hạn của đề án nên chỉ sử dụng công thức tính độ tương tự thông dụng nhất là cosine với $u_{\alpha,\beta}$ là các vectơ người dùng (vectơ cột) tương ứng trong ma trận tiện ích chuẩn hóa $\overline{Y}$.

$$\text{cosine}(u_\alpha, u_\beta) = \frac{\mathbf{u}_\alpha^T \cdot \mathbf{u}_\beta}{\|\mathbf{u}_\alpha\|_2 \|\mathbf{u}_\beta\|_2} = \frac{\sum_{i=0}^{M-1} \bar{y}_{i,\alpha} \bar{y}_{i,\beta}}{\sqrt{\sum_{i=0}^{M-1} \bar{y}_{i,\alpha}^2} \sqrt{\sum_{i=0}^{M-1} \bar{y}_{i,\beta}^2}} \quad (1.12)$$

Để có thể đo được độ tương tự giữa hai người dùng, cách thường làm là xây dựng vectơ đặc trưng cho mỗi người dùng rồi áp dụng độ tương tự cosine giữa hai vectơ. Các vectơ đặc trưng được xây dựng dựa trên ma trận tiện ích $\boldsymbol{Y}$, tuy nhiên khó khăn đặt ra vì ma trận này thường là một ma trận thưa (sparse matrix) bao gồm nhiều giá trị bị khuyết vì người dùng thường chỉ đánh giá một lượng rất nhỏ các sản phẩm. Giải pháp đơn giản nhất là điền vào những phần trống này một giá trị ước lượng. Những giá trị này chỉ phục vụ cho phần tính độ tương tự chứ không phải kết quả cuối cùng mà hệ thống cần dự đoán.

1.7.2.2. Lọc cộng tác theo sản phẩm

User – CF đạt được nhiều thành công trong quá khứ nhưng cũng gặp phải một số hạn chế khi được sử dụng rộng rãi như: *Sự thưa thớt*: Thực tế ngay cả với người dùng tích cực nhất cũng chỉ có thể mua được số sản phẩm chiếm tỷ lệ rất thấp trong tổng số sản phẩm. Do đó, hệ tư vấn User – CF có thể không đưa ra bất kỳ gợi ý nào;

Một cách tiếp cận khác là lọc cộng tác sản phẩm (Item - CF), được đề xuất bởi Sarwar cùng cộng sự [14] và được Amazon sử dụng cho hệ tư vấn

của họ [15]. Cách thức tính toán thay vì tìm sự tương tự giữa các người dùng, ta có thể tìm sự tương tự giữa các sản phẩm. Từ đó nếu một người dùng thích một sản phẩm thì hệ thống nên gợi ý các sản phẩm tương tự với sản phẩm đó. Và nếu lượng sản phẩm nhỏ hơn số lượng người dùng, mô hình này sẽ có những ưu điểm như tính toán ít hơn do ma trận tiện ích có số hàng ít hơn số cột nên ảnh hưởng bởi đánh giá của một người dùng sẽ ít ảnh hưởng đến giá trị trung bình của tổng các đánh giá của mọi người dùng tới sản phẩm đó. Như vậy ma trận tương tự sản phẩm **S** sẽ không cần cập nhật quá thường xuyên. Thêm nữa là ma trận tương tự sản phẩm **S** có kích thước nhỏ hơn với số hàng bằng số sản phẩm M nên giúp lưu trữ và tính toán ở những bước sau hiệu quả hơn.

1.7.2.3. Lọc cộng tác phân tích ma trận

Khác với hai phương pháp trên còn được biết đền là lọc cộng tác lân cận, một phương pháp được gọi là matrix factorization (MF) cho CF được biết đến lần đầu tiên do Simon Funk đăng trên blog năm 2006 [16] kể về lý do đằng sau việc anh ấy và đồng nghiệp giành được vị trí thứ 3 trong giải thưởng Netflix. Thay vì áp dụng mô hình Singular Value Decomposition (SVD), giải pháp của Simon Funk là phân tích ma trận tiện ích thành tích hai ma trận có số chiều thấp hơn, ma trận thứ nhất có hàng cho mỗi người dùng và ma trận thứ hai có cột tương ứng với mỗi sản phẩm. Hàng và cột này được liên kết với nhau được gọi là latent feature.

Với phương pháp trên kết hợp với nghiên cứu của Koren và cộng sự [17]. Nhiệm vụ hàng đầu là cần cố gắng tính xấp xỉ ma trận tiện ích $\boldsymbol{Y} \in \mathbb{R}^{M \times N}$ bằng tích hai ma trận: ma trận thông tin sản phẩm $\mathbf{X} \in \mathbb{R}^{K \times M}$ và ma trận mô hình người dùng $\mathbf{W} \in \mathbb{R}^{K \times N}$. Giá trị K ở đây chính là tính chất tiềm ẩn và thường nhỏ hơn so với M và N, khi đó cả hai ma trận **X** và **W** đều có hạng (rank) không vượt quá K.

- **Bước 1: Xây dựng hàm mất mát**
- **Bước 2: Tối ưu hàm mất mát**

1.7.2.4. Lọc cộng tác dựa trên bộ tự mã hóa

Trong những năm gần đây, mạng nơ-ron học sâu hay tên gọi tiếng anh là Deep neural networks đã được sử dụng phổ biến trong các hệ thống đề xuất. Một số công trình đã xuất hiện để cải thiện hệ thống đề xuất với kỹ thuật điển hình là Autoencoder [20]

Tổng kết lại có 2 cách chính để áp dụng Autoencoder vào hệ thống tư vấn: Trực tiếp ước tính các giá trị bị thiếu trong ma trận tiện ích bằng cách sử dụng lớp tái xây dựng (reconstruction layer); Hay là sử dụng để biểu diễn đặc trưng kết hợp giảm số chiều của chúng thông qua lớp bottleneck và tái tạo đầu ra với số chiều nhỏ hơn đầu vào.

❖ Cách thức hoạt động:

Cả bộ mã hóa và bộ giải mã đều là các mạng nơ ron chuyển tiếp được kết nối đầy đủ, về cơ bản là các ANN. Bottleneck là một lớp duy nhất với kích thước khá nhỏ. Số nút trong các lớp (kích thước Encoder và Decoder) là một siêu tham số được đặt trước khi huấn luyện cho mô hình Autoencoder. Cụ thể hơn, đầu tiên đầu vào đi qua bộ mã hóa, là một ANN được kết nối đầy đủ dùng để tạo mã. Bộ giải mã, có cấu trúc ANN tương tự, sau đó tạo đầu ra chỉ bằng cách đảo ngược bộ mã hóa. Mục tiêu là để có được một đầu ra giống với đầu vào. Nhưng tùy bài toán cụ thể mà input và output sẽ giống về nội dung chứ giá trị thì không hẳn, ví dụ như làm mờ ảnh.

Tập dữ liệu không gán nhãn $D = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$, giả sử rằng chiều của bottleneck m là đủ nhỏ so với số chiều của đầu vào và đầu ra p ($m < p$). Và nếu autoencoder đã được đào tạo mang lại $\mathbf{x} = \hat{\mathbf{x}}$ thì điều đó có nghĩa là ta cũng đã có hàm giảm chiều dữ liệu. Ví dụ trong thực nghiệm của đề án này ta có $p = 1682$ và $m = 20$. Với encoder được huấn luyện, ta có thể chuyển

đổi mỗi user với vectơ chứa 1682 đặc trưng thành vectơ nhỏ hơn nhiều với kích thước 20. Với decoder sau khi huấn luyện, chỉ cần chuyển đổi ngược lại và nhận được giá trị gần đúng với giá trị gốc. Lựa chọn $m = 20$ ngụ ý rằng hệ số nén vào khoảng 84.

Có thể coi autoencoder như một hàm $f_\theta: \mathbb{R}^p \to \mathbb{R}^p$, trong đó θ là tham số có thể huấn luyện. Các tham số θ có ảnh hưởng tới hoạt động của hàm như sau: $f_\theta(\mathbf{x}^*) \approx \mathbf{x}^*$ trong đó $\mathbf{x}^*$ là một dữ liệu bất kỳ như dữ liệu trong tập huấn luyện (seen data) hoặc dữ liệu trong tập test (unseen data).

Xây dựng hàm mất mát dựa trên bình phương khoảng cách Euclid:

$$\text{Vì} \quad L_i(\theta) = \left\|\mathbf{x}^{(i)} - f_\theta(\mathbf{x}^{(i)})\right\|_2^2 \Longrightarrow L(\theta; D) = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{x}^{(i)} - f_\theta(\mathbf{x}^{(i)})\right\|_2^2 \qquad (1.31)$$

trong đó $L_i(\theta)$ là thước đo sai số của đầu vào $\mathbf{x}^{(i)}$ và đầu ra $\hat{\mathbf{x}}^{(i)} = f_\theta(\mathbf{x}^{(i)})$.

Có thể liên hệ với học có giám sát trong đó chúng ta có thể so sánh nhãn huấn luyện với nhãn dự đoán, thì autoencoder tuy thuộc nhóm học không giám sát nhưng ta có thể coi đầu vào như là nhãn $y^{(i)}$ của học có giám sát.

$f_\theta(\cdot)$: hàm encoder được ký hiệu là $f^{[1]}_{\theta^{[1]}}(\cdot)$ và hàm decoder được ký hiệu là $f^{[2]}_{\theta^{[2]}}(\cdot)$, trong đó $\theta^{[1]}$ là các tham số của encoder và $\theta^{[2]}$ là các tham số của decoder:

$$\hat{\mathbf{x}} = f_\theta(\mathbf{x}) = \left(f^{[1]}_{\theta^{[1]}} \circ f^{[2]}_{\theta^{[2]}}\right)(\mathbf{x}) = f^{[2]}_{\theta^{[2]}}\left(f^{[1]}_{\theta^{[1]}}(\mathbf{x})\right) \qquad (1.32)$$

Trong trường hợp bài toán chỉ với một lớp ẩn như hình 1.5 ta có thể xác định công thức cụ thể như sau:

$$\begin{aligned} f^{[1]}_{\theta^{[1]}}(\mathbf{u}) &= \mathbf{S}^{[1]}\left(\mathbf{b}^{[1]} + \mathbf{W}^{[1]}\mathbf{u}\right) \quad \text{với} \quad \mathbf{u} \in \mathbb{R}^p \quad \text{(Encoder)} \\ f^{[2]}_{\theta^{[2]}}(\mathbf{u}) &= \mathbf{S}^{[2]}\left(\mathbf{b}^{[2]} + \mathbf{W}^{[2]}\mathbf{u}\right) \quad \text{với} \quad \mathbf{u} \in \mathbb{R}^m \quad \text{(Decoder)} \end{aligned} \qquad (1.33)$$

trong đó: Nhóm encoder có các tham số $\theta^{[1]}$ đã bao gồm vectơ bias $\mathbf{b}^{[1]} \in \mathbb{R}^m$ và ma trận trọng số $\mathbf{W}^{[1]} \in \mathbb{R}^{m \times p}$, $\mathbf{S}^{[1]}(\cdot)$ là vectơ các hàm kích hoạt với $\mathbf{S}^{[1]}: \mathbb{R}^m \to \mathbb{R}^m$. Nhóm decoder có các tham số $\theta^{[2]}$ đã bao gồm vectơ bias

$\mathbf{b}^{[2]} \in \mathbb{R}^p$ và ma trận trọng số $\mathbf{W}^{[2]} \in \mathbb{R}^{p\times m}$, $\mathbf{S}^{[2]}(\cdot)$ là vectơ các hàm kích hoạt với $\mathbf{S}^{[2]}: \mathbb{R}^p \to \mathbb{R}^p$. Vì vậy, danh sách toàn bộ các tham số cho mạng autoencoder như sau:

$$\theta = \left(\mathbf{b}^{[1]}, \mathbf{W}^{[1]}, \mathbf{b}^{[2]}, \mathbf{W}^{[2]}\right) \tag{1.34}$$

Làm rõ vectơ các hàm kích hoạt $\mathbf{S}^{[1]}(\cdot)$ và $\mathbf{S}^{[2]}(\cdot)$, ta khái quát thành $\mathbf{S}^{[l]}(\cdot)$ được xây dựng bởi các hàm kích hoạt vô hướng $\sigma^{[l]}: \mathbb{R} \to \mathbb{R}$ với $l = 1,2$ chẳng hạn như hàm sigmoid, ReLU. Có $\mathbf{S}^{[l]}(\mathbf{z})$ được xác định bởi $\sigma^{[l]}(\cdot)$ trên mỗi phần tử của $\mathbf{z}$ bằng phép toán element-wise:

$$\mathbf{S}^{[l]}(\mathbf{z}) = \begin{bmatrix} \sigma^{[l]}(z_1) \\ \vdots \\ \sigma^{[l]}(z_r) \end{bmatrix} \tag{1.35}$$

Tính giá trị cho các nơ-ron bottleneck $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_m$ và nơ-ron đầu ra $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_p$ với $\mathbf{a}_k = \hat{\mathbf{x}}_k$, $\mathbf{a}$ đại diện cho giá trị đã có hàm kích hoạt.

$$\begin{aligned} \tilde{\mathbf{x}}_i &= \sigma^{[1]}\left(b_i^{[1]} + \sum_{k=1}^{p} \mathbf{W}_{ik}^{[1]} \mathbf{x}_k\right) \quad \text{với} \quad i = 1, \dots, m \\ \hat{\mathbf{x}}_j &= \sigma^{[2]}\left(b_j^{[2]} + \sum_{k=1}^{m} \mathbf{W}_{jk}^{[2]} \mathbf{a}_k\right) \quad \text{với} \quad j = 1, \dots, p \end{aligned} \tag{1.36}$$

Viết lại hàm kích hoạt (1.31):

$$L(\theta; D) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}^{(i)} - \underbrace{\mathbf{S}^{[2]}(W^{[2]}\mathbf{S}^{[1]}(\mathbf{W}^{[1]}\mathbf{x}^{(i)} + \mathbf{b}^{[1]}) + \mathbf{b}^{[2]})}_{f_\theta(\mathbf{x}^{(i)})}\|_2^2 \tag{1.37}$$

1.7.2.5. Lọc cộng tác phân tích giá trị suy biến

Trong phần này, hệ tư vấn được xây dựng dựa trên phương pháp phân tích giá trị suy biến [11] còn được viết tắt là SVD thuộc nhóm các phương pháp matrix factorization. Đây là một phương pháp cổ điển từ đại số tuyến tính đã trở nên phổ biến trong lĩnh vực khoa học dữ liệu và học máy. SVD là một kỹ thuật phân tích ma trận thành tích của nhiều ma trận giúp giảm chiều dữ liệu mà không yêu cầu là ma trận vuông như lý thuyết chéo hóa ma trận (Diagonalizable matrix) đã đề cập.

Kết luận: Chương này đã trình bày tổng quát về định nghĩa hệ tư vấn, phân loại các cách tiếp cận phổ biến và mô tả thêm một số phương pháp phổ biến hay được áp dụng trong thực tế. Hơn nữa, một số ví dụ cụ thể nhằm mô tả hoạt động của các phương pháp cũng được trình bày một cách kỹ lưỡng nhằm biểu diễn tốt cách hoạt động của chúng.

CHƯƠNG II: MÔ HÌNH DỰA TRÊN ĐỒ THỊ VÀ HỌC SÂU

2.1. Cơ sở lý thuyết cho mô hình GHRS

2.1.1. Lựa chọn đặc trưng dựa trên đồ thị

2.1.1.1. PageRank

Thuật toán PageRank (PR) được phát minh bởi Serge Brin và Larry Page là công thức toán học đánh giá sự ảnh hưởng của một nút thông qua việc xem xét số lượng của các nút liên kết với nó. Được sử dụng lần đầu tiên để đánh giá tầm quan trọng tương đối của website trong toàn bộ hệ thống World Wide Web hay đơn giản là xếp hạng các trang web nhằm hỗ trợ tác vụ tìm kiếm của Google, thuật toán đã giúp nâng cao chất lượng kết quả tìm kiếm cao hơn rất nhiều so với các công cụ tìm kiếm khác.

Dựa trên bài báo của Zahra và Valipour [21], các tác giả đã xếp hạng các nút bằng việc lặp đi lặp lại việc phân phối giá trị mỗi nút theo tỷ lệ với các nút kết nối với nó. Các nút khác sau đó nhận được giá trị từ nút gốc và quá trình phân phối xếp hạng sẽ tiếp tục diễn ra cho đến khi hội tụ. Phương pháp này cho ra kết quả tương tự như kỹ thuật Random Walk.

2.1.1.2. Degree Centrality

Hệ số trung tâm trực tiếp (Degree Centrality - DC) giúp đo lường được số lượng các mối quan hệ trực tiếp của một nút nào đó với các nút khác trong đồ thị. Giá trị của hệ số này chạy từ 0 đến 1 và khi giá trị tiến gần đến 1 thì

tính trung tâm của nút càng lớn, tức là nút này càng có nhiều kết nối với các nút khác trong đồ thị, điều này đồng nghĩa với việc nút đó nắm giữ nhiều thông tin nhất. Ngược lại khi tiến dần về 0 thì nút trở nên phụ thuộc vào các nút khác, mang ít giá trị khai thác hơn.

2.1.1.3. Closeness Centrality

Hệ số trung tâm lân cận (Closeness Centrality - CC) là hệ số tìm các nút có mối liên hệ "gần" nhất với một nút nào đó trong mạng đồ thị hay cụ thể hơn là tìm những nút có thể chia sẻ thông tin hiệu quả. Một nút như vậy là quan trọng khi nó nằm trên đường đi ngắn nhất tới tất cả các nút còn lại. Hệ số này khắc phục điểm yếu của hệ số DC khi chỉ xét đến mối quan hệ trực tiếp mà bỏ qua tiếp xúc gián tiếp giữa các nút, một nút có hệ số DC cao không có nghĩa là nút đó là nút "gần" nhất với mọi thành viên trong mạng đồ thị.

2.1.1.4. Betweenness Centrality

Định nghĩa chính thức về hệ số trung tâm trung gian (Betweenness Centrality - BC) được đề xuất đầu tiên bởi Linton Freeman, là một trong những thước đo tính trung tâm (Centrality) quan trọng trong mạng đồ thị. Hệ số mô tả một nút bất kỳ có thể mang giá trị C_D và C_C thấp nhưng lại có ý nghĩa cầu nối giữa các nút khác trong đồ thị hay đơn giản là thông tin trao đổi cần đi qua nó.

2.1.1.5. Load Centrality

Một thước đo khác về tính trung tâm được gọi là Load Centrality (LC), một hệ số khá tương tự như hệ số BC. Tuy nhiên, thay vì dựa vào số lượng đoạn đường ngắn nhất, LC ước tính lưu lượng trên một nút, giả sử rằng lưu lượng được phân bố công bằng giữa tất cả các đoạn đường ngắn nhất của đồ thị. Hệ số LC hội tụ về hệ số BC trong nhiều trường hợp thực tế.

2.1.1.6. Average Neighbor Degree

Giá trị của mỗi nút có thể tính bằng cách lấy trung bình của tổng bậc các nút hàng xóm của nút đó.

2.1.2. Autoencoder

2.1.2.1. Autoencoder denoising

Khác với mô hình cơ bản trong hình 1.5, mô hình này học cách loại bỏ nhiễu trong lớp tái xây dựng từ dữ liệu đầu vào bị nhiễu. Nó nhận đầu vào bị hỏng một phần và học cách khôi phục đầu vào đã khử nhiễu ban đầu. Về mặt kiến trúc, autoencoder denoising có kiến trúc tương tự autoencoder cổ điển, tuy nhiên điểm khác biệt chính là trong quá trình huấn luyện, mô hình được đào tạo trên mẫu bị nhiễu và hàm mất mát đã được sửa tương ứng.

Đầu tiên, đầu vào ban đầu $\mathbf{x}$ bị biến đổi thành $\tilde{\mathbf{x}}$ thông qua một số phép ánh xạ ngẫu nhiên ví dụ như Gussian noise, masking noise hay salt-and-pepper noise. Vì vậy công thức (1.32) được biến đổi thành:

$$\hat{\tilde{\mathbf{x}}} = f_\theta(\tilde{\mathbf{x}}) = \left(f_{\theta^{[1]}}^{[1]} \circ f_{\theta^{[2]}}^{[2]}\right)(\tilde{\mathbf{x}}) = f_{\theta^{[2]}}^{[2]}\left(f_{\theta^{[1]}}^{[1]}(\tilde{\mathbf{x}})\right) \tag{2.20}$$

Tiếp theo cần sửa hàm mất mát (1.31), ta so sánh giữa $\mathbf{x}$ với $\hat{\tilde{\mathbf{x}}}$ thay vì $\tilde{\mathbf{x}}$ với $\hat{\tilde{\mathbf{x}}}$:

$$L(\theta; D) = \frac{1}{n}\sum_{i=1}^{n} \left\|\mathbf{x}^{(i)} - f_\theta(\tilde{\mathbf{x}}^{(i)})\right\|_2^2 \tag{2.21}$$

Vì vậy trong quá trình huấn luyện, ta tìm được các tham số θ để cố gắng loại bỏ nhiễu càng nhiều càng tốt.

2.1.2.2. Hồi quy ElasticNet

Một mô hình hồi quy kết hợp của hồi quy Lasso và hồi quy Ridge giúp cân bằng trong việc lựa chọn các đặc trưng (điều này giải quyết vấn đề đa cộng tuyến) nhưng cũng không làm mất tính ổn định của mô hình. Kết hợp với (2.21), ta có:

$$L(\theta; D) = \frac{1}{n}\sum_{i=1}^{n} \left\|\mathbf{x}^{(i)} - f_\theta(\tilde{\mathbf{x}}^{(i)})\right\|_2^2 + \alpha_1 \sum_{i=1}^{n} |\theta_i| \quad (2.22)$$

trong đó α_1 và α_2 là hệ số chính quy hóa lần lượt của hồi quy Lasso và Ridge.

2.1.3. Phân cụm người dùng

Dựa theo đề xuất của Zahra và Valipour [21], mỗi người dùng sẽ thuộc một cụm nào đó và mỗi phân cụm được coi là xếp hạng ước tính cho nhóm người dùng đó. Căn cứ vào kết quả thu được sau quá trình huấn luyện thông qua mạng học sâu Autoencoder để thực hiện phân loại bằng thuật toán phân cụm K-means. Một vấn đề nữa được đặt ra là làm sao tìm được số cụm k thích hợp, ở đây hai tác giả đã sử dụng phương pháp Elbow và Average Silhouette để giải quyết vấn đề nay.

2.1.3.1. K-means

Thuật toán phân cụm K-means (K-means clustering) là một giải thuật của học không giám sát. Trong đó ta không biết được nhãn của từng điểm dữ liệu. Mục đích là làm sao để chia dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có những tính chất giống nhau, hiểu theo một cách hình học là khoảng cách của các thành viên trong cụm tới tâm cụm (centroid) của chúng phải là gần hơn so với tâm của các cụm khác, kiểu phân chia này trong toán học được gọi là sơ đồ Voronoi.

2.1.3.2. Phương pháp Elbow

Tính đến hiện tại đã có nhiều biện pháp để xác định xem số cụm thích hợp để tách nhóm dữ liêu. Sự phân tách đo mức độ khác biệt của một cụm với một cụm khác. Sai số Cluster cohesion có thể được đo bằng Within Cluster Sum of Square (WCSS) và Cluster separation với Between Cluster Sum Squares (BCSS)

$$\text{WCSS} = \sum_{j}^{N_C} \sum_{\mathbf{x} \in C_j} \left\|\mathbf{x} - \mathbf{m}_j\right\|_2^2 \quad (2.27)$$

trong đó N_C là tập hợp các cụm, C_j là phân cụm thứ j thuộc tập N_C.

$$\text{BCSS} = \sum_{j}^{N_C} |C_j| \left\| \bar{\mathbf{m}} - \mathbf{m}_j \right\|_2^2 \quad (2.28)$$

trong đó $|C_j|$ là số cụm trong N_C và $\bar{\mathbf{m}}$ là tâm cụm của toàn bộ tập dữ liệu.

2.1.3.3. Phương pháp Silhouette

Phương pháp này sử dụng hệ số silhouette là sự kết hợp của Cluster cohesion và Cluster separation. Kết quả mong muốn khi hệ số silhouette trả về càng cao càng tốt.

$$\text{WCSS} < \text{BCSS}: \quad s = 1 - \frac{\text{WCSS}}{\text{BCSS}}, \quad \text{ngược lại:} \quad s = \frac{\text{BCSS}}{\text{WCSS}} - 1 \quad (2.29)$$

2.2. Cơ sở thực nghiệm

Tập dữ liệu thực nghiệm được sử dụng trong đề án là MovieLens-100k [1] đã được giới thiệu tại mục mở đầu. Bộ dữ liệu này được công bố năm 1998 bởi GroupLens. Nó bao gồm 100.000 đánh giá từ 943 người dùng cho 1682 bộ phim và hơn nữa đã được chia sẵn thành các tập huấn luyện và tập kiểm tra cho việc kiểm định mô hình.

2.3. Xây dựng mô hình GHRS

Bước 1: Xây dựng đồ thị với những người dùng như các nút. Hai người dùng sẽ được kết nối dựa trên những đặc điểm giống nhau theo mục 2.1.1. Cạnh kết nối là những người có sự tương đồng về xếp hạng.

Bước 2: Trích xuất thông tin của người dùng từ đồ thị.

Bước 3: Tiến hành kết hợp thông tin phụ như giới tính và độ tuổi với các đặc trưng dựa trên đồ thị ở bước 2 làm đầu vào cho giai đoạn Autoencoder.

Bước 4: Áp dụng kĩ thuật Autoencoder được mô tả trong mục 1.7.2.4 và 2.1.2 để trích xuất đặc trưng mới và giảm kích thước dữ liệu.

Bước 5: Sử dụng các đặc trưng mới được mã hóa bởi Autoencoder để phân cụm người dùng, sử dụng thuật toán K-means đã trình bày trong mục 2.1.3 để tạo ra một số lượng các nhóm người dùng có sự tương đồng.

Bước 6: Phân bố người dùng mới vào cụm thích hợp dựa trên các đặc trưng được mã hóa và dự đoán xếp hạng các mục mới mà người dùng đó chưa xếp hạng.

Bước 7: Dự đoán xếp hạng của người dùng cho tất cả các mục theo xếp hạng trung bình của cụm và tiến hành đề xuất cho người dùng.

Kết luận: Chương này đã trình bày một cách cụ thể quá trình xây dựng mô hình GHRS. Sáu kỹ thuật được sử dụng trong quá trình xây dựng đồ thị, phương pháp Autoencoder với nhiễu đầu nhằm tăng cường khả năng học các đặc trưng tiềm ẩn và hai phương pháp phổ biến cho việc tìm hệ số phân cụm tối ưu.

CHƯƠNG III: KẾT QUẢ THỰC NGHIỆM

3.1 Môi trường thực nghiệm

3.1.1 Môi trường thực nghiệm

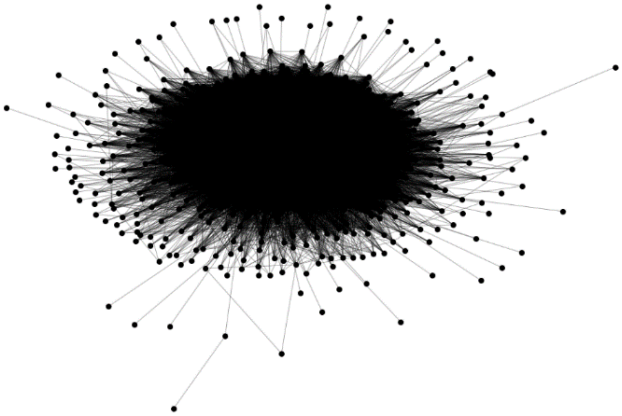
Toàn bộ quá trình cài đặt huấn luyện và kiểm thử được cài đặt trên 2 môi trường: JupyterLab với Intel(R) Core i5-4460 CPU @ 3.20GHz, bộ nhớ RAM 8GB; Google Colab với Intel(R) Xeon(R) CPU @ 2.20GHz, bộ nhớ RAM 12.7 GB.

3.1.2 Ngôn ngữ và thư viện lập trình

Xuyên suốt quá trình thực hiện đề án, tôi đã sử dụng Python 3.7.4 trên JupyterLab cho tất cả các mô hình ngoại trừ mạng Autoencoder và mô hình GHRS được triển khai trên Google Colab sử dụng Python 3.10.12. Các thư viện đi kèm bao gồm: numpy, pandas, matplotlib, sklearn, scipy, pytorch, networkx và một số thư viện hỗ trợ đọc file, tìm đường dẫn.

3.2 Thực hiện các bước xây dựng mô hình GHRS

Bước 1,2: Dựa trên lý thuyết đã trình bày tại mục 2.1.1, cũng như mô tả trong nghiên cứu [21] và hệ số $\alpha = 0.01$ được lựa chọn cho thực nghiệm này. Kết quả đồ thị tương tự (Similarity Graph) thu được như trong hình 3.1 mô tả mối liên hệ giữa 943 người dùng dựa trên đánh giá từ 1682 bộ phim:



Hình 3.1: Đồ thị tương tự của 943 người dùng

Bước 3,4: Dựa trên mô tả trong nghiên cứu [21] về thông tin người dùng kết hợp với kết quả thu được từ bước 2, ta thu được bảng mô tả đặc trưng các user được mô tả trong hình 3.2.

	age1	age2	age3	age4	age5	age6	gender1	...	PR	CD	CC	CB	LC	AND
0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.460577	0.673966	0.817829	0.043582	0.043606	0.479257
1	0.0	0.0	0.0	0.0	0.0	1.0	1.0	...	0.044933	0.014599	0.568223	0.000000	0.000000	0.660178
2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.0	0.0	0.0	1.0	0.0	0.0	1.0	...	0.189245	0.255474	0.653926	0.000660	0.000662	0.610428
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
938	0.0	0.0	1.0	0.0	0.0	0.0	1.0	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
939	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.107218	0.116788	0.618164	0.000025	0.000026	0.705393
940	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
941	0.0	0.0	0.0	0.0	1.0	0.0	1.0	...	0.057353	0.036496	0.574410	0.000000	0.000000	0.709976
942	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.260841	0.369830	0.697905	0.002970	0.002976	0.581300

943 rows × 35 columns

Hình 3.2: Các đặc trưng đầu vào cho Autoencoder

Sau khi chuẩn bị xong dữ liệu, ta xây dựng mô hình Autoencoder như hình 3.3 nhằm giảm chiều dữ liệu để có thể dễ dàng phân cụm người dùng tại bước 5.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                [-1, 16]             576
           Sigmoid-2                [-1, 16]               0
            Linear-3                 [-1, 8]             136
            Linear-4                [-1, 16]             144
           Sigmoid-5                [-1, 16]               0
            Linear-6                [-1, 35]             595
================================================================
Total params: 1,451
Trainable params: 1,451
Non-trainable params: 0
----------------------------------------------------------------
```

Hình 3.3: Thông tin mạng Autoencoder

Bước 5,6,7: Kết thúc bước 4 ta thu được ma trận đặc trưng. Và để áp dụng giải thuật K-means, lúc này cần xác định hệ số k thông qua phương pháp Elbow và Silhouette được trình bày trong mục 2.1.3. Kết quả thu được cho biết số cụm tốt nhất là 3 với số lượng người dùng của từng cụm được mô tả như hình 3.4.

```
Cluster 0: 405 users
Cluster 1: 342 users
Cluster 2: 196 users
```

Hình 3.4: Số người dùng trong mỗi cụm

Hình 3.5 dưới đây mô tả cụ thể user thuộc nhóm nào với 1 đại diện cho user đang được phân vào cụm (cột) tương ứng còn 0 là user đó không thuộc cụm (cột) người dùng đó.

	cluster1	cluster2	cluster3
1	1.0	0.0	0.0
2	1.0	0.0	0.0
3	1.0	0.0	0.0
...	...	...	...
941	0.0	0.0	1.0
942	0.0	1.0	0.0
943	1.0	0.0	0.0

943 rows × 3 columns

Hình 3.5: Ma trận phân cụm người dùng

Thực hiện tương tự cho ma trận phân cụm bộ phim, ta thu được kết quả như hình 3.6:

	1	2	3	4	5	6	...	1681	1682
cluster1	3.960591	3.202899	3.071429	3.555556	3.25	2.875	...	3.0	3.0
cluster2	3.769231	3.136364	2.769231	3.381818	3.411765	3.777778	...	3.262132	3.702399
cluster3	3.729412	3.233333	3.166667	3.727273	3.277778	3.666667	...	3.297181	3.695487

3 rows × 1682 columns

Hình 3.6: Ma trận phân cụm bộ phim

Cuối cùng ta nhân ma trận phân cụm người dùng với ma trận phân cụm bộ phim để có ma trận dự đoán của người dùng với bộ phim được mô tả trong hình 3.7.

	1	2	3	4	5	6	...	1681	1682
1	3.960591	3.202899	3.071429	3.555556	3.25	2.875	...	3.0	3.0
2	3.960591	3.202899	3.071429	3.555556	3.25	2.875	...	3.0	3.0
3	3.960591	3.202899	3.071429	3.555556	3.25	2.875	...	3.0	3.0
4	3.960591	3.202899	3.071429	3.555556	3.25	2.875	...	3.0	3.0
5	3.960591	3.202899	3.071429	3.555556	3.25	2.875	...	3.0	3.0
...	...	...	...	...	...	...	...	...	...
939	3.960591	3.202899	3.071429	3.555556	3.25	2.875	...	3.0	3.0
940	3.729412	3.233333	3.166667	3.727273	3.277778	3.666667	...	3.297181	3.695487
941	3.729412	3.233333	3.166667	3.727273	3.277778	3.666667	...	3.297181	3.695487
942	3.769231	3.136364	2.769231	3.381818	3.411765	3.777778	...	3.262132	3.702399
943	3.960591	3.202899	3.071429	3.555556	3.25	2.875	...	3.0	3.0

943 rows × 1682 columns

Hình 3.7: Ma trận dự đoán người dùng – bộ phim

3.3 Kết quả mô hình và so sánh

Trong đề án này, chúng tôi tiến hành đánh giá hiệu suất của một loạt các phương pháp dự đoán dựa trên cùng một tập huấn luyện **ua.base** và cùng một tập kiểm tra **ua.test**. Các phương pháp đã được trình bày tại chương I và chương II sẽ được đưa vào thực nghiệm bao gồm Content-based, User-based CF, Item-based CF, User-based Matrix Factorization, Item-based MF, Autoencoder-based CF, User-based SVD, Item-based SVD và GHRS.

Bảng 3.1: So sánh độ chính xác giữa các mô hình

Method	RMSE
Content-based	1.2703
User - CF	0.9767
Item - CF	0.9678
User - MF	1.0431
Item - MF	1.0421
Autoencoder - CF	0.9678
User - SVD	1.0031
Item - SVD	1.0050

GHRS	1.0518

Trong số các phương pháp này, User-based CF, Item-based CF và Autoencoder-based CF đã thể hiện sự hiệu quả cao nhất với các giá trị RMSE thấp nhất. Đặc biệt, Autoencoder-based CF cho thấy sự tiến bộ đáng kể trong việc học biểu diễn dữ liệu và dự đoán chính xác hơn về sở thích của người dùng.

Nhận xét: Từ bảng 3.1 ta thấy được mô hình Lọc nội dung có kết quả kém nhất với RMSE ≈ 1.2703, trong khi đó ba mô hình Item - CF, User - CF và Autoencoder - CF cho kết quả khả quan hơn rất nhiều với RMSE đều xấp xỉ 0.97. Mô hình GHRS đã không cho kết quả như kỳ vọng như theo bài báo [21] với RMSE ≈ 1.0518 gần như tương đương với RMSE thu được từ các phương pháp MF. Phương pháp sử dụng kỹ thuật SVD cho kết quả khá tốt với RMSE lần lượt là 1.0031 và 1.0050.

Qua nhận xét về kết quả thu được trong bảng 3.1 phía trên, ta có thể nhận thấy rằng việc thay đổi trong cách xây dựng tập huấn luyện và kiểm tra đã dẫn tới phương pháp được chứng minh có hiệu suất cao như GHRS lại cho ra kết quả kém hơn so với một vài phương pháp phổ biến khác. Việc tác giả bỏ qua thực nghiệm kiểm tra trên tập dữ liệu huấn luyện có phân phối không đồng nhất là một thiếu sót khiến người đọc không nhận thức được rõ ràng phạm vi mà mô hình GHRS có thể đạt được hiệu quả cao và những trường hợp GHRS không thể cho ra kết quả tốt hơn so với các phương pháp phổ biến khác.

KẾT LUẬN CHUNG

I. Kết quả đạt được

Đề án hướng tới một chủ đề quan trọng cả về mặt lý thuyết và thực tiễn trong khoa học máy tính, nó đã và đang được nhiều nhà nghiên cứu quan tâm, đó là nghiên cứu một số phương pháp xây dựng hệ tư vấn.

❖ Về mặt lý thuyết

Đề án đã trình bày tổng quan về bài toán xây dựng hệ tư vấn cũng như vai trò của bài toán trong xã hội hiện nay kèm theo là xu hướng phát triển. Những lý thuyết này về cơ bản là những lý thuyết nền tảng của học máy thống kê và học sâu.

❖ Về mặt thực tiễn

Đề án đã triển khai cài đặt được một số phương pháp nổi bật nhất trong lĩnh vực xây dựng hệ tư vấn, kết quả thực nghiệm của từng phương pháp, từ đó đánh giá được ưu nhược điểm của các phương pháp đó dựa trên tập dữ liệu Movielens-100k.

II. Hạn chế và hướng phát triển

❖ Hạn chế

Do hạn chế về mặt phạm vi và thời gian, kèm theo đó là sức mạnh phần cứng đã cản trở rất nhiều tới hiệu suất thực hiện việc cài đặt các mô hình. Phần cứng yếu đã khiến việc triển khai trên một tập dữ liệu lớn hơn như MovieLens 10M 20M [1] là không khả thi. Các vấn đề khác chưa giải quyết được như khởi động nguội hay việc dữ liệu liên tục cập nhật.

❖ Hướng phát triển

Định hướng tiếp theo của tác giả là tìm hiểu và xây dựng phương pháp mới cho hệ tư vấn trong thời điểm các mô hình ngôn ngữ lớn đã và đang trở nên phổ biến như hiện nay.