ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN

TRẦN NGUYÊN LỘC

KHẢO SÁT VỀ HỆ THỐNG KHUYẾN NGHỊ TRÊN MỘT SỐ TRANG WEB XEM PHIM

ĐỒ ÁN CHUYÊN NGÀNH
NGÀNH: CÔNG NGHỆ THÔNG TIN

Thành phố Hồ Chí Minh, tháng 12 năm 2023

ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN

TRẦN NGUYÊN LỘC

KHẢO SÁT VỀ HỆ THỐNG KHUYẾN NGHỊ TRÊN MỘT SỐ TRANG WEB XEM PHIM

ĐỒ ÁN CHUYÊN NGÀNH
NGÀNH: CÔNG NGHỆ THÔNG TIN

Giảng viên phụ trách
TS. PHAN TẤN QUỐC

Thành phố Hồ Chí Minh, tháng 12 năm 2023

MỤC LỤC

LỜI CẢM ƠN

Trước hết em xin gửi đến lời cảm ơn chân thành và sâu sắc nhất đến thầy TS. Phan Tấn Quốc, người trực tiếp hướng dẫn và tận tình chỉ bảo cho em cho tới khi em hoàn thành đồ án của mình.

Tiếp đến em dành lời cảm ơn đến quý thầy cô khoa Công nghệ thông tin – trường Đại học Sài Gòn đã truyền đạt cho em những kiến thức vô cùng quý báu và bổ ích trong suốt quá trình nghiên cứu và học tập tại trường.

Xin chân thành cảm ơn tới những người bạn đã luôn sát cánh cùng em, những lời động viên, những lần hỗ trợ những lúc cần thiết đã phần nào giúp em hoàn thành đồ án này.

Cuối cùng, em xin cảm ơn đến ba mẹ và người thân trong gia đình đã hỗ trợ và tạo điều kiện thuận lợi cho em trong suốt thời gian học tập và nghiên cứu tại Đại học Sài Gòn.

LỜI CAM ĐOAN

Tôi xin cam đoan rằng đồ án chuyên ngành "Khảo sát về hệ thống khuyến nghị trên một số trang web xem phim" là công trình nghiên cứu của riêng tôi, không sao chép lại của người khác. Trong toàn bộ nội dung của luận văn, những điều đã được trình bày hoặc là của chính cá nhân tôi được tổng hợp từ nhiều nguồn tài liệu. Tất cả tài liệu tôi kham khảo được đều có xuất xứ rõ ràng và hợp pháp.

Tôi hoàn toàn cam chịu trách nhiệm và chịu mọi hình thức kỉ luật theo quy định nếu tái phạm.

Thành phố Hồ Chí Minh, Tháng 12 Năm 2023

Ký tên

Trần Nguyên Lộc

DANH MỤC HÌNH ẢNH

LỜI MỞ ĐẦU

Lý do chọn đề tài

Với sự phát triển của xã hội hiện nay, việc nhu cầu giải trí của con người ngày càng tăng cao trong số đó phải kể đến là nhu cầu giải trí thông qua phim ảnh. Với lịch sử lâu đời của ngành điện ảnh cùng với kho tài liệu phim ảnh khổng lồ đã được sản xuất qua năm tháng, cùng với đó là sự phát triển của công nghệ hiện đại ngày nay đã cho phép các bộ phim có thể được lưu trữ trong các cơ sở dữ liệu lớn và được sử dụng để phục vụ cho lượng lớn khán giả trong hầu hết các trang web xem phim trực tuyến. Nhưng với số lượng phim khổng lồ và hàng ngàn thể loại phim khác nhau như thế sẽ là một vấn đề đối với các nhà sản xuất phim ảnh, họ có thể sẽ không thể thu lại lợi nhuận từ phim của họ nếu bộ phim mà họ sản xuất không được đến được tay khán giả hoặc không được nhiều người chú ý đến. Do đó, Hệ thống Khuyến Nghị là một trong những giải pháp ứng dụng phù hợp và tốt nhất để giải quyết cho vấn đề trên, nhằm thu hút nhiều khán giả đến với các sản phẩm phim hơn, đồng thời cũng là cầu nối cho khán giả đến với các bộ phim.

Và ngày nay, hệ thống Khuyến Nghị (Recommender System) là một trong những lớp ứng dụng thành công và phổ biến nhất của trí tuệ nhân tạo. Với các nền tảng dịch vụ trực tuyến đang ngày càng phát triển mạnh mẽ trong đời sống hiện nay thì Hệ thống Khuyến Nghị đóng một vai trò rất lớn trong việc ứng dụng vào các ngành Dịch vụ của con người, Ví dụ: Thương mại điện tử, mua bán các sản phẩm dịch vụ trực tuyến, ứng dụng trực tuyến, xem phim/video trực tuyến ..v..vv.

Trên các trang web xem phim, hệ thống Khuyến nghị đóng một vai trò chính trong việc giới thiệu các bộ phim đến với các khán giả. Nó đóng vai trò phân tích và tìm hiểu khối dữ liệu cá nhân của người dùng và từ đó đưa ra những dự đoán, gợi ý, đề xuất phù hợp với sở thích của khán giả. Các trang web xem phim lớn hiện nay ứng dụng thành công hệ thống Khuyến nghị như Netflix, BiliBili, FPT Play …

Mục đích nghiên cứu

Đề tài của đồ án chuyên ngành tập trung phân tích và làm rõ cách thức hoạt động của hệ thống khuyến nghị một số trang web xem phim, đồng thời phân tích chi tiết các thuật toán được ứng dụng để giải quyết bài toán khuyến nghị phim.

Nhiệm vụ nghiên cứu

Khái quát và tổng quan về Hệ thống Khuyến nghị.

Phân tích về các thuật toán được sử dụng để giải quyết bài toán khuyến nghị phim.

Khảo sát về Hệ thống khuyến nghị trên các trang web xem phim.

Đối tượng nghiên cứu và phạm vi nghiên cứu

Đối tượng nghiên cứu: các thuật toán được sử dụng để giải quyết cho bài toán khuyến nghị phim trên một số trang web xem phim.

Phạm vi nghiên cứu: Trang web xem phim như Netflix, BiliBili, FPT Play.

Phương pháp nghiên cứu

Phương pháp quan sát: Quan sát hành vi thu thập thông tin người dùng của một số hệ thống Khuyến nghị nhằm phân tích bài toán khuyến nghị phim.

Phương pháp điều tra: Tìm hiểu cụ thể đặc điểm và tính chất của bài toán Khuyến nghị phim trên các trang web xem phim nhằm đưa ra một bài viết dễ hình dung cho người đọc và có độ chính xác về mặt nội dung cao.

Cấu trúc đồ án chuyên ngành

Cấu trúc của đồ án chuyên ngành gồm 3 phần chính:

Chương 1. Tổng quan về hệ thống khuyến nghị

Chương 1 trong đồ án chuyên ngành sẽ giới thiệu tổng quan về hệ thống khuyến nghị, lý thuyết của bài toán khuyến nghị và các phương pháp tiếp cận của hệ thống khuyến nghị và các khảo sát nghiên cứu có liên quan với mỗi phương pháp. Sau đó, giới thiệu một số phương pháp/thuật toán được sử dụng để giải quyết bài toán khuyến nghị phim.

Chương 2. Một số phương pháp giải quyết bài toán khuyến nghị phim

Chương 2 trong đồ án chuyên ngành sẽ tập trung khảo sát và phân tích vào các thuật toán, phương pháp, hệ thống được sử dụng phổ biến trong bài toán khuyến nghị phim, trình bày chi tiết các phương pháp như PMF (Probabilistic Matrix Factorization), BPMF (Bayesian Probabilistic Matrix Factorization), ALS (Alternating Least Squares).

Chương 3. Khảo sát hệ thống khuyến nghị phim trên một số trang web xem phim

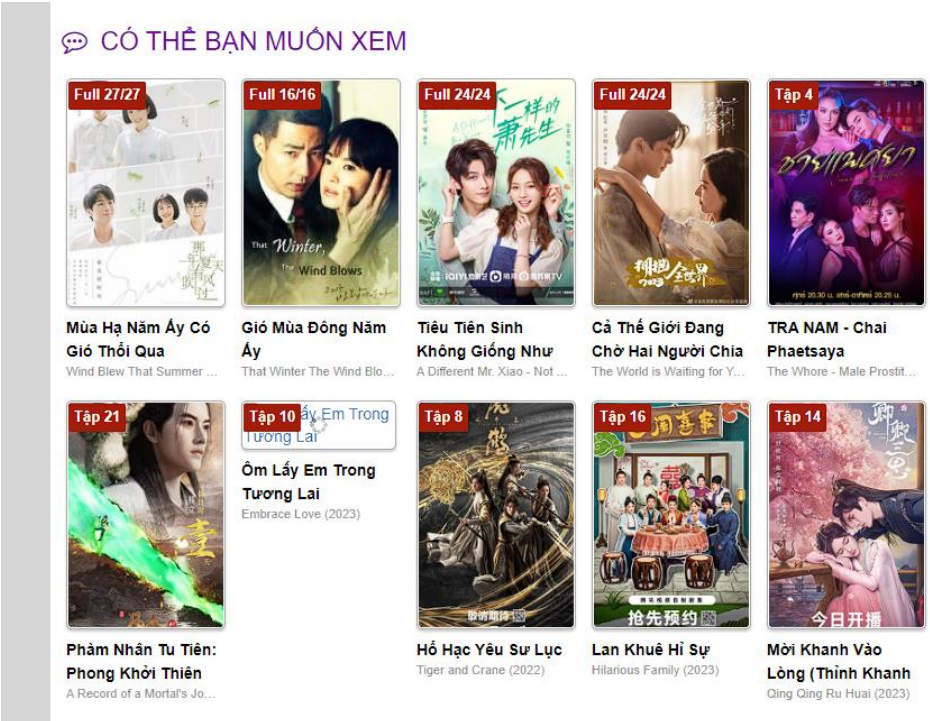
Chương 3 trong đồ án chuyên ngành sẽ tập trung vào việc khảo sát cách thức của một hệ thống khuyến nghị hoạt động trên các trang web xem phim. Các trang web được khảo sát trong đồ án chuyên ngành gồm Netflix, BiliBili và FPT Play. Sau đó ta so sánh cách thức xử lý bài toán khuyến nghị phim giữa 3 trang web xem phim trên.

CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG KHUYẾN NGHỊ

1.1. Giới thiệu chung

Hệ thống khuyến nghị (có tên gọi tiếng anh là Recommender System) hay còn được gọi là hệ thống tư vấn là một hệ thống có nhiệm vụ chọn lọc thông tin nhằm dự đoán sở thích, mức độ phù hợp, mối quan tâm và nhu cầu của người dùng để đưa ra một hoặc nhiều mục, sản phẩm, dịch vụ mà người dùng sẽ quan tâm với xác suất lớn nhất [1].

Và hiện nay, mọi hệ thống, ứng dụng có hiển thị quảng cáo trên internet đều sử dụng hệ thống khuyến nghị để đưa ra quảng cáo, đề xuất tốt nhất có thể đến cho người dùng… Một vài ví dụ phổ biến và dễ gặp nhất đó là gợi ý sản phẩm hoặc dịch vụ có liên quan trên các trang web, ứng dụng phổ biến.



Hình 1. 1. Ví dụ về hệ thống khuyến nghị của một trang web xem phim

Và để làm được điều đó, hệ thống khuyến nghị đã sử dụng những thuật toán để phân tích và dự đoán đựa trên dữ liệu hành vi người dùng được lưu lại. Nhờ đó,

những quảng cáo mang tính cá nhân hóa được đưa đến cho người dùng. Hệ thống sẽ biết chính xác từng người dùng sử dụng có nhu cầu gì, muốn gì để từ đó đưa ra khuyến nghị.

Trong thực tế, ý tưởng để những người lập trình xây dựng một hệ thống khuyến nghị không đâu xa lạ chính là xuất phát từ hành vi của người mua hàng và người bán hàng: Khi một người mua hàng có nhu cầu mua một sản phẩm, họ thường sẽ có hành vi hỏi người bán hàng để tư vấn cho họ về sản phẩm mà họ có ý định mua. Người bán hàng sẽ tiến hành thu thập thông tin từ người mua bao gồm: nhu cầu sử dụng, đặc điểm, mức độ phù hợp, chức năng, màu sắc, … đồng thời kết hợp với kiến thức hiểu biết của mình về sản phẩm để đưa ra đề xuất, lời khuyên sản phẩm phù hợp nhất cho người mua. Và ở một mức độ cao hơn, người bán sẽ liên hệ, liên tưởng những người đã từng mua sản phẩm mà có đặc điểm tương đồng vời người mua hiện tại, từ đó họ dự đoán người mua hiện tại có khả năng thích sản phẩm nào nhất để đưa ra khuyến nghị sản phẩm phù hợp nhất cho người mua.

1.2. Phát biểu về bài toán khuyến nghị

Để có thể xây dựng được một hệ thống khuyến nghị hoàn chỉnh cho từng lĩnh vực cụ thể, các nghiên cứu trước đã phát biểu được lời giải chung cho bài toán khuyến nghị như sau:

Định nghĩa 1: Không gian người dùng [2]

Không gian người dùng là tập tất cả những người dùng mà hệ thống quan sát được, để thực hiện phân tích, khuyến nghị. Ký hiệu là *U*, $U = \{u_1, u_2, u_3, \dots u_i\}$.

Định nghĩa 2: Không gian đối tượng khuyến nghị [2]

Không gian đối tượng khuyến nghị là tập tất cả những đối tượng sẽ được khuyến nghị cho người dùng. Tùy vào ứng dụng cụ thể, đối tượng khuyến nghị có thể là sản phẩm, dịch vụ hoặc con người … Ký hiệu là *P*, $P = \{p_1, p_2, p_3, \dots p_j\}$.

Định nghĩa 3: Hàm phù hợp [3]

Hàm phù hợp *F* là ánh xạ $F : U \times P \rightarrow R$, dùng để ước lượng độ phù hợp của p ∈ P với u ∈ U. Với R là một ma trận có thứ tự các số nguyên hoặc thực trong một khoảng nhất định.

Phát biểu bài toán khuyến nghị [1]**:**

Đầu vào (Input):

+ Tập người dùng *U*, mỗi người dùng u_i thuộc *U* và có các đặc điểm I = $\{i_1, i_2, i_3, \dots\ i_k\}$.

+ Một tập sản phẩm, dịch vụ (ở đây ta gọi chung là sản phẩm) P, mỗi sản phẩm p_i có các đặc điểm đặc trưng J = $\{j_1, j_2, j_3, \dots\ j_k\}$.

+ Một ma trận đánh giá tổng quát R = (r_{ij}) với i = 1, … N và j = 1…M, thể hiện mối quan hệ giữa tập người dùng U đối với tập sản phẩm P. Trong đó r_{ij} là đánh giá của người dùng u_i cho sản phẩm p_i, N và M là lần lượt số người dùng và số sản phẩm.

Người dùng \ Sản phẩm	1	2	…	*i*	…	*M*
1	5	3	0	1	2	0
2	0	2	0	0	0	4
:	0	0	5	0	0	0
u	3	4	0	2	1	0
:	0	0	0	0	4	0
N	0	0	3	2	0	0
a	3	5	0	?	1	0

Hình 1. 2. Ví dụ ma trận đánh giá tổng quát Rij

Đầu ra (Output):

Danh sách các sản phẩm p_i thuộc *P* có độ phù hợp với người dùng u_i thuộc *U* nhất.

Để giải bài toán này chúng ta cần xây dựng hàm F(u_i, p_i) để đo độ phù hợp của sản phẩm p_i với người dùng u_i, từ đó ta sẽ lấy được danh sách các sản phẩm, dịch vụ phù hợp (là các sản phẩm, dịch vụ mà có khả năng được người dùng chọn) nhất.

Và cũng tùy thuộc vào phương pháp sử dụng mà ta có nhiều các xây dựng hàm F khác nhau, các cách xây dựng hàm F phụ thuộc chủ yếu bởi các yếu tố sau [1]:

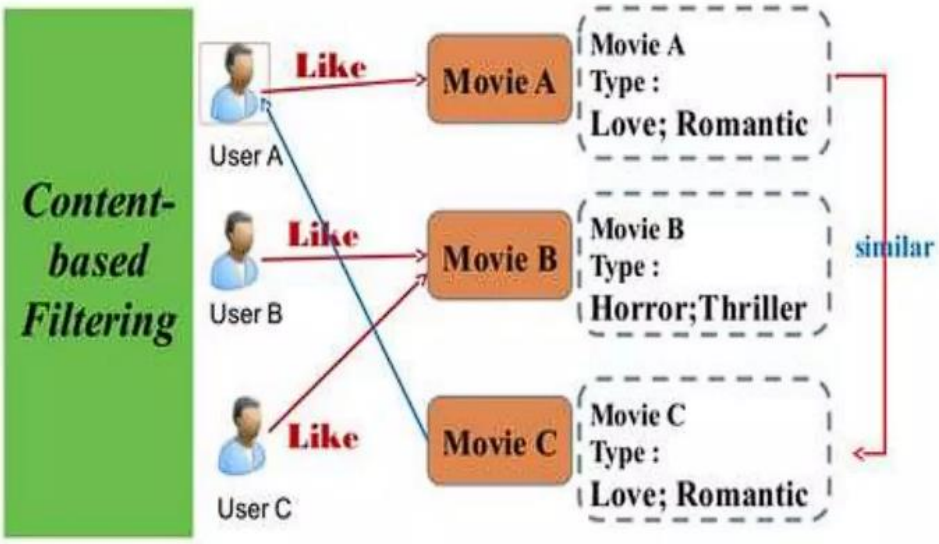
- Đặc điểm của người dùng u_i (lọc theo nội dung người dùng): đặc điểm này được đánh giá chủ quan bởi các quy luật tự nhiên hoặc các quy tắc cơ bản. Ví dụ u_i là nữ thì sẽ có xu hướng chọn mua các sản phẩm của nữ hơn là các sản phẩm của nam [1].
- Đặc điểm của sản phẩm p_i (lọc theo nội dung sản phẩm): cũng giống như lọc theo nội dung người dùng, các sản phẩm có đặc điểm giống nhau thì cũng có khả năng được người dùng đánh giá như nhau. Ví dụ đặc điểm của các món đồ công nghệ có thể là đặc điểm, tính năng, nhu cầu sử dụng … [1]
- Lịch sử giao dịch của người dùng u_i: từ lịch sử giao dịch cũng có thể suy ra sản phẩm mà người dùng u_i quan tâm đến, do đó các sản phẩm cùng thể loại, lĩnh vực sẽ có độ liên quan cao hơn. Ví dụ một người đã từng mua áo, giày bóng đá thì có thể dự đoán được được người là một người đam mê bóng đá, thích thể thao. Từ đó suy ra được người này sẽ có khả năng sử dụng dịch vụ hoặc mua các sản phẩm thể thao cao hơn các dịch vụ, sản phẩm khác [1].
- Những người dùng u_j khác có cùng đặc điểm giống với người dùng u_i: với quan niệm rằng những người dùng giống nhau sẽ thích, đánh giá những sản phẩm giống nhau. Các đặc điểm u_j bao gồm tập đặc điểm *I* ban đầu, kết hợp với các đặc điểm cộng tác như cùng mua mặt hàng nào đó hoặc có các hành vi mua hàng giống nhau… Việc tìm hiểu những mặt hàng, dịch vụ mà u_j đã từng quan tâm sẽ đưa ra được những gợi ý phù hợp cho người dùng u_i [1].

1.3. Các hướng tiếp cận của bài toán khuyến nghị

1.3.1. Phương pháp lọc dựa trên nội dung

1.3.1.1. Định nghĩa

Lọc dựa trên nội dung (tên tiếng anh là Content-Base Filtering) là phương pháp thực hiện dựa trên việc so sánh nội dung thông tin hay mô tả hàng hóa, để tìm ra những sản phẩm tương tự với những gì mà người dùng đã từng quan tâm để giới thiệu cho họ những sản phẩm này [1].



Hình 1. 3. Ví dụ mô hình kỹ thuật lọc dựa trên nội dung

Trong hình 1.3 ta có thể thấy mô hình minh họa cho việc lọc theo nội dung như sau: User A đánh giá thích Movie A và Movie A có thể loại Love, Romantic. Do đó, phương pháp lọc theo nội dung sẽ dựa theo Type (Thể loại) của Movie A và từ đó khuyến nghị Movie C có cùng Type (Thể loại) với Movie A cho User A.

1.3.1.2. Khái quát bài toán

Để thực hiện việc ước lượng xem có hay không một người dùng *u* có thích đối tượng sản phẩm *p*. Ta xây dựng một hàm phù hợp *f(u,p)* của các sản phẩm khuyến nghị *p* với người dùng *u* và ước lượng giá trị phù hợp này. Các phương pháp tiếp cận nội dung thường sẽ thực hiện các bước sau đây [4]:

- **Bước 1:** Biểu diện nội dung đối tượng khuyến nghị $p \in P$, ký hiệu *Content(p)*.
- **Bước 2:** Mô hình hóa sở thích người dùng $u \in U$, gọi tắt là hồ sơ người dùng (User's Profile), ký hiệu *UserProfile(u)*.
- **Bước 3:** Ước lượng giá trị phù hợp dựa trên độ tương tự nội dung của sản phẩm khuyến nghị *p* với hồ sơ người dùng *u*. Hệ thống sẽ ưu tiên khuyến nghị những đối tượng sản phẩm *p* có nội dung tương tự cao so với hồ sơ người dùng *p*.

Pasquale Lops và cộng sự đã tiến hành khảo sát, phân tích các hệ khuyến nghị dựa trên tiếp cận nội dung [5]. Theo Lops, hệ khuyến nghị dựa trên tiếp cận nội dung thường sẽ thực hiện ba việc chính và được đảm nhận bởi ba thành phần tương ứng:

- Phân tích nội dung (Content Analyzer): Thành phần này có nhiệm vụ phân tích mô hình háo nội dung của các đối tượng khuyến nghị. Tuỳ vào bài toán, các phương pháp rút trích đặc trưng sẽ được dùng để chuyển đổi nội dung đối tượng khuyến nghị từ định dạng gốc sang không gian đặc trưng.
- Mô hình hoá hồ sơ người dùng (Profile Learner): Các nghiên cứu thường dùng các phương pháp học máy giám sát để học hồ sơ người dùng dựa trên đặc trưng của các đối tượng mà người dùng thích hay không thích trong quá khứ. Qua thời gian sở thích người dùng có thể thay đổi, hệ thống thường sẽ định kì học và cập nhật lại hồ sơ người dùng.
- Lọc nội dung (Filtering Component): Thành phần này có nhiệm vụ so khớp hồ sơ người dùng với nội dung của các đối tượng để thực hiện khuyến nghị những đối tượng phù hợp với sở thích người dùng

1.3.1.3. Phân loại các cách tiếp cận lọc dựa trên nội dung

Phương pháp lọc dựa trên nội dung có thể được chia ra làm 2 nhóm chính:

1. Phương pháp lọc dựa trên bộ nhớ, thực hiện tính toán độ tương tự giữa *Content(p)* và *UserProfile(u)* dùng các độ do lường tương tự Consine, Euclide [6].

2. Phương pháp lọc dựa trên mô hình, với mô hình được học từ dữ liệu dùng các kỹ thuật học máy giám sát để phân các sản phẩm khuyến nghị thành những sản phẩm được người dùng quan tâm hoặc không quan tâm như: phân lớp SVM [7], phân lớp Bayesian [8] và các phương pháp xác xuất như Pazzani và Billsus [9], Mooney và Roy [10], Gemmis và đồng nghiệp [5].

a) Tiếp cận nội dung dựa trên bộ nhớ

Tiếp cận nội dung dựa trên bộ nhớ (hay còn gọi là phương pháp dựa trên bộ nhớ) là phương pháp thường được thực hiện với việc ước lượng mức độ phù hợp của đối tượng huyến nghị $p \in P$ với người dùng $u \in U$ (tức giá trị hàm phù hợp *f(u,p)*) dựa trên việc tổng hợp mức độ quan tâm u đối với tập k đối tượng có nội dung tương tự với p, ký hiệu là $P_k = \{p_k\}$, $P_k \subseteq P$, hoặc tổng hợp mức độ quan tâm từ tập k những người dùng có sở thích tương tự u, $U_k = \{u_k\}$, $U_k \subseteq U$. Tùy thuộc vào cách biểu diễn nội dung đối tượng dữ liệu và hồ sơ người dùng, chúng ta sẽ có một hàm phù hợp để tính độ tương tự và xác định tập P_k cũng như U_k. Thông thường, các nghiên cứu dùng mô hình không gian vector độ đo Consine để biểu diễn nội dung và tính độ tương tự giữa các đối tượng [4].

Phương pháp dựa trên bộ nhớ có những ưu điểm và nhược điểm như sau [4]:

❖ **Ưu điểm:**

- Đơn giản, dễ thực hiện.
- Chất lượng khuyến nghị thường tốt hơn do tính toán trên cả tập dữ liệu khi thực hiện khuyến nghị.

❖ **Nhược điểm:**

- Tốn bộ nhớ và tốc độ xử lý chậm do phải tính toán, trên cả tập dữ liệu thực khi thực hiện khuyến nghị.
- Không thể tổng quát hóa tập dữ liệu.

b) Tiếp cận nội dung dựa trên mô hình

Với phương pháp dựa trên bộ nhớ, hệ thống thường sẽ tính giá trị hàm phù hợp dựa trên các độ do như Cosine, Euclide. Đối với các phương pháp dựa trên mô

hình, một mô hình sẽ được huấn luyện từ dữ liệu để phân các đối tượng khuyến nghị thành những đối tượng được người dùng quan tâm hay không quan tâm và quan tâm nhiều hay ít dùng các phương pháp học máy giám sát: phân lớp SVM [7], phân lớp Bayesian [8] và một số phương pháp xác xuất khác. Nói cách khác, mô hình huấn luyện giúp tiên đoán giá trị hàm phù hợp *f(u,p)* của đối tượng khuyến nghị $p \in P$ đối với người dùng $u \in U$ [4]. Chẳng hạn, phân lớp Bayesian là một phương pháp dựa trên mô hình khá phổ biến, được dùng trong khai thác dữ liệu, phân lớp Bayesian có thể dùng để ước lượng xác xuất đối tượng khuyến nghị *p* phù hợp với *u* như thế nào. Hay nói cách khác, *p* được *u* quan tâm không hay quan tâm nhiều hay ít [8].

Ví dụ, xác xuất một tài liệu *p* được một người dùng *u* nào đó quan tâm là bao nhiêu? Tức là, giá trị hàm phù hợp *f*(*u*,*p*) khi đó được tính dựa trên việc ước lượng xác xuất *p* thuộc lớp C_1(u) và C_0(u) (u quan tâm và không quan tâm đến p) là bao nhiêu, khi cho trước một tập các từ khóa mô tả tài liệu p là $\{k_{1,p}, k_{2,p}, k_{3,p}, \dots\ k_{n,p}\}$. Giá trị hàm phù hợp *f*(*u*,*p*) khi đó được tính như sau [4]:

$$f(u,p) = \text{P}(\text{p} \in C_1(\text{u})) = \text{P}(C_1(u)|k_{1,p}, k_{2,p}, k_{3,p}, \dots\ k_{n,p})$$

Giả sử các từ khóa mô tả tài liệu là độc lập, khi đó xác xuất P(p ∈ C_1(u)) sẽ là [4]:

$$\text{P}(\text{p} \in C_1(\text{u})) = \text{P}(C_1(u)|k_{1,p}, k_{2,p}, k_{3,p}, \dots\ k_{n,p}) = \text{P}(C_1(u))\prod_{i=1}^{n} \text{P}(k_{i,p}|C_1(u))$$

Nhìn chung phương pháp dựa trên mô hình có ưu điểm và khuyến điểm như sau [4]:

- **Ưu điểm**
  - Khả năng đáp ứng tốt khi tập dữ liệu được gia tăng.
  - Một mô hình biểu diễn tốt thế giới thực sẽ giúp tránh được vấn đề khớp (overfitting) so với phương pháp dựa trên bộ nhớ.
  - Nhanh hơn so với phương pháp dựa trên bộ nhớ do không phải tính trên cả tập dữ liệu mà chỉ dựa vào mô hình đã xây dựng để khuyến nghị.

❖ **Khuyết điểm**

- Phải xây dựng và cập nhật lại mô hình khi có sự thay đổi. Đây là quá trình gây tốn tài nguyên.
- Chất lượng tiên đoán thấp hơn so với các phương pháp dựa trên bộ nhớ vì không được tính toán trên cả tập dữ liệu. Tuy nhiên, nò tùy thuộc vào chất lượng của mô hìnnh được được xây dựng có phản ánh tốt thế giới thực hay không, tức là có đúng với thực tế hay không.

1.3.1.4. Ưu điểm và khuyết điểm của lọc dựa trên nội dung

❖ **Ưu điểm:**

- Là phương pháp trực quan, dễ dàng hiểu và giải thích được [11].
- Không bị ảnh hưởng bởi khởi đầu lạnh (cold start) [11].
- Không bị ảnh hưởng bởi vấn đề vấn đề thưa thớt dữ liệu.
- Có thể khuyến nghị những sản phẩm mới hoặc sản phẩm không phổ biến.
- Có thể khuyến nghị cho những người dùng có sở thích riêng.

❖ **Khuyết điểm:**

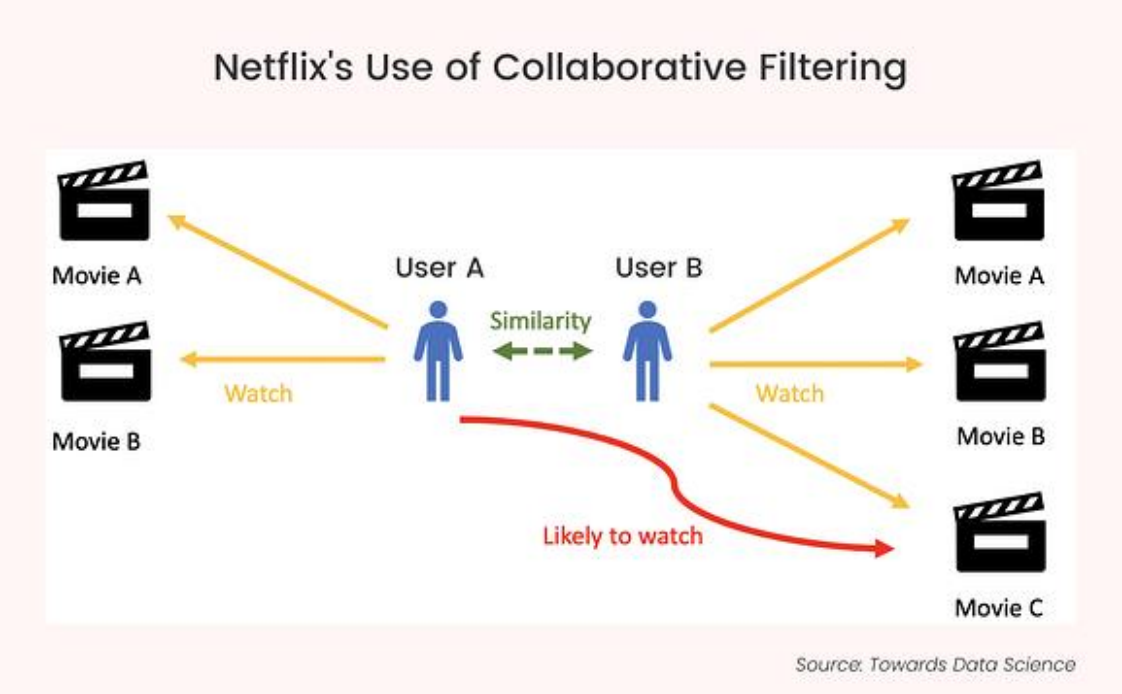
- Thường gặp các khó khăn liên quan đến phân tích nội dung.
- Gặp các vấn đề khi khuyến nghị cho người dùng mới (Khởi động lạnh).
- Không thể đa dạng trong khuyến nghị (bao gồm các đối tượng khuyến nghị ngoài lĩnh vực quan sát).

1.3.2. Phương pháp lọc cộng tác

1.3.2.1. Định nghĩa

Lọc cộng tác (tên tiếng anh là Collaborative-Filtering) là phương pháp khai thác những khía cạnh liên quan đến thói quen sử dụng sản phẩm của một nhóm người dùng có cùng sở thích trong quá khứ để đưa ra dự đoán các sản phẩm mới phù hợp với người dùng hiện tại (có thể hình dung rằng là lọc cộng tác giả định rằng

những người đồng ý trong quá khứ sẽ đồng ý trong tương lai rằng họ sẽ thích các mặt hàng tương tự như các mặt hàng mà họ đã thích trong quá khứ) [12].



Hình 1. 4. Ví dụ mô hình kỹ thuật lọc cộng tác

Trong hình minh họa 1.3.2.1 ta có thể thấy phương pháp lọc cộng tác hoạt động như thế nào trên hệ thống Netflix. User A đã xem Movie A và Movie B, User B thì đã xem Movie A, Movie B và Movie C và giữa User A và User B đều có sự tương đồng về sở thích ở một mức độ giống nhau. Suy ra phương pháp lọc cộng tác sẽ cho rằng User A cũng sẽ có thể thích xem Movie C (Movie C là movie mà User B đã xem).

1.3.2.2. Khái quát bài toán

Cũng giống như lọc dựa trên nội dung, phương pháp lọc cộng tác cũng xây dựng một ma trận đánh giá gồm danh sách các người dùng $U = \{u_1, u_2, u_3, \dots u_i\}$ và danh sách các sản phẩm $P = \{p_1, p_2, p_3, \dots p_j\}$ nhằm tìm kiếm những giá trị tiên đoán độ phù hợp giữa sản phẩm p và người dùng u được gọi là ma trận A(U,P). Ma trận A có kích thước là i x j và chứa các giá trị đánh giá a_{ij} với i ∈ 1…N và j ∈ 1…M. Những giá trị a_{ij} này thể hiện mức độ phù hợp của đối tượng p_j với người dùng u_i. Giá trị a_{ij} có thể là giá trị nguyên hay thực trong 1 khoảng tùy vào bài

toán. Thông thường, giá trị đánh giá mức độ phù hợp a_{ij} trong hầu hết hệ thống ứng dụng phổ biến nhận giá trị từ 1 (không phù hợp) đến 5 (rất phù hợp). Nếu người dùng u_i chưa thể hiện đánh giá với đối tượng p_j thì giá trị $a_{ij} = \emptyset$ và cần được thu thập hoặc tính toán [4] (Ví dụ minh họa hình 1.5).

	p_1	p_2	p_3	p_4	p_5	...	p_m
u_1	1	?	5	?	4	?	?
u_2	?	?	4	?	5	?	?
u_3	?	4	?	5	?	?	?
u_4	?	?	?	4	?	?	?
u_5	?	?	?	5	?	?	?
...	?	?	?	?	?	?	?
u_n	?	3	?	?	?	?	5

Hình 1. 5. Dấu ? là những giá trị cần tiên đoán trong ma trận đánh giá

Ý tưởng chung của phương pháp lọc cộng tác là khai thác thông tin, hành vi quá khứ của người dùng dựa trên các đánh giá sẵn có từ ma trận đánh giá để tiên đoán, lượng hóa mức độ phù hợp của các đối tượng sản phẩm khuyến nghị mà người dùng chưa biết [4].

1.3.2.3. Phân loại các cách tiếp cận lọc cộng tác

Các phương pháp lọc cộng tác được phân thành hai nhóm chính [4]:

1. Lọc cộng tác với cách tiếp cận dựa trên bộ nhớ (Memory-Based) như các thuật toán tính toán lân cận, tương tự.
2. Lọc cộng tác với cách tiếp cận dựa trên mô hình (Model-Based) như các thuật toán gom cụm, phân lớp giám sát, thừa số hóa ma trận.

a) Tiếp cận lọc cộng tác dựa trên bộ nhớ

Các hệ thống lọc cộng tác dựa trên bộ nhớ thường dùng các kỹ thuật thống kê để tìm kiếm những người dùng, hoặc các đối tượng khuyến nghị tương tự nhau dựa trên thông tin đánh giá, hành vi quá khứ của người dùng từ ma trận đánh giá. Tiếp cận lọc cộng tác dựa trên bộ nhớ tìm cách ước lượng giá trị hàm phù hợp *f(u,p)* của đối tượng khuyến nghịp với người dùng *u* dựa trên những đánh giá của những người đồng sở thích của *u* đối với *p* (lọc dựa trên người dùng), hoặc dựa trên những đánh giá của *u* với các đối tượng khuyến nghị *p'* tương tự với *p* (lọc dựa trên đối tượng khuyến nghị). Về cơ bản, thì các thuật toán, kỹ thuật tỉnh toán cho lọc cộng tác dựa trên người dùng và lọc dựa trên đối tượng khuyến nghị từ ma trận đánh giá là tương tự nhau. Có khác chăng là kích thước của không gian người dùng và không gian đối tượng khuyến nghị sẽ ảnh hưởng đến tốc độ tính toán khi xác định nhóm các đối tượng tương tự. Phương pháp lọc cộng tác với cách tiếp cận dựa trên bộ nhớ có đặc trưng cơ bản là thường sử dụng toàn bộ dữ liệu đã có để dự đoán đánh giá của một người dùng nào đó về sản phẩm mới [4]. Cách tiếp cận dựa trên bộ nhớ thường được chia làm 2 loại: dựa trên người dùng và dựa trên sản phẩm:

- **Dựa trên người dùng**

  Phương pháp này gồm 2 bước như sau:

  - Bước 1: Tìm kiếm những người dùng có đánh giá tương tự với người dùng cần được dự đoán.
  - Bước 2: Sử dụng đánh giá từ những người dùng được tìm thấy ở bước 1 để tính toán dự đoán cho người cần được dự đoán.

- **Dựa trên sản phẩm**

  Phương pháp này gồm 2 bước như sau:

  - Bước 1: Xây dựng một ma trận để xác định mối quan hệ giữa các cặp sản phẩm với nhau.
  - Bước 2: Kiểm tra thị hiếu của người dùng cần dự đoán bằng cách kiểm tra ma trận và kết hợp dữ liệu của người dùng đó.

b) Tiếp cận lọc cộng tác dựa trên mô hình

Phương pháp lọc cộng tác với cách tiếp cận dựa trên mô hình chủ yếu phát triển các mô hình bằng cách sử dụng các khai phá dữ liệu khác nhau, các thuật toán học máy để dự đoán đánh giá của người dùng về các mặt hàng chưa được đánh giá [4]. Theo quan điểm xác xuất, thì các thuật toán lọc cộng tác dựa trên mô hình cần tính toán xác suất mà người dùng u đánh giá $a_{u,p}$ cho một đối tượng khuyến nghị p, P($a_{u,p}|u,p$). Quá trình đó có thể xem như việc tính toán giá trị kỳ vọng cho đánh giá của người dùng u với đối tượng khuyến nghị p [13].

Khác với lọc cộng tác dựa trên bộ nhớ, các thuật toán lọc cộng tác dựa trên mô hình dùng tập các đánh giá có sẵn trong ma trận A để học một mô hình đánh giá cho mỗi người dùng. Sau đó, mô hình học được sẽ dùng để tiên đoán các đánh giá khác [4]. Một số thuật toán lọc cộng tác dựa trên mô hình được sử dụng phổ biến như Thuật toán lọc cộng tác gom cụm, Thuật toán lọc cộng tác dựa trên xác xuất Bayes [13], Thừa số hóa ma trận (Matrix Factorization)…

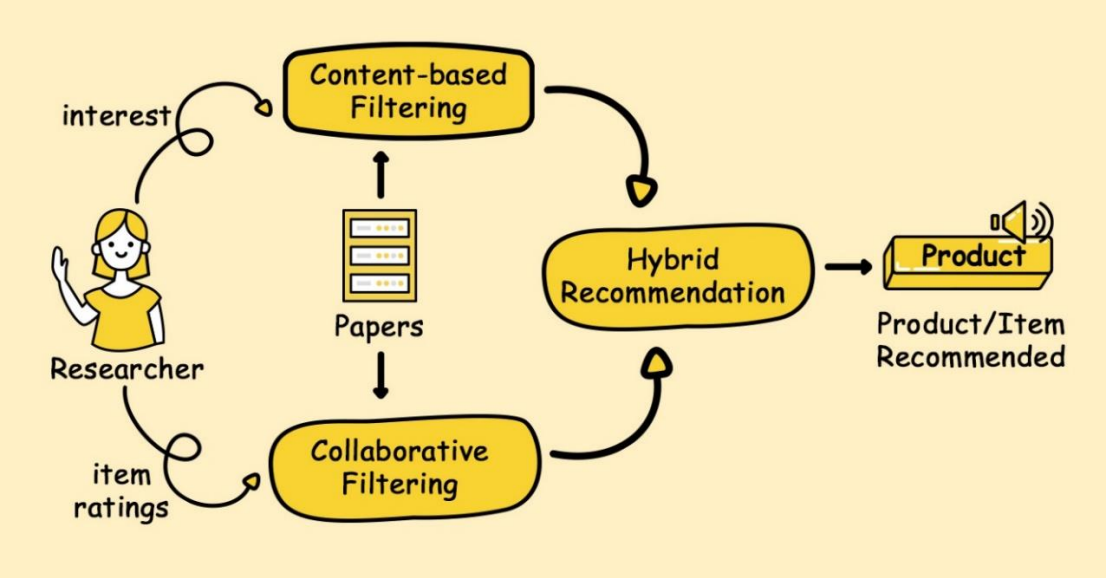
1.3.2.4. Ưu điểm và khuyết điểm của lọc cộng tác

- **Ưu điểm:** Theo các tác giả Su và Khoshgoftaar, tiếp cận lọc cộng tác được xem là một trong những cách tiếp cận thành công nhất để thiết kế các thuật toán và xây dựng các hệ khuyến nghị [14]. Các ưu điểm của tiếp cận này là:
  - Có khả năng dự doán sở thích và nhu cầu của người dùng để đưa ra các gợi ý sản phẩm phù hợp với từng khách hàng mà không cần hiểu sản phẩm.
  - Phù hợp với những hệ thống lớn có nhiều đánh giá từ phía người dùng.
- **Khuyết điểm:** Tuy nhiên, theo nghiên cứu tổng quan của các tác giả Adomavicius và Tuzhilin [3]; Tác giả Bobadilla và cộng sự [15] thì tiếp cận lọc cộng tác cũng có một số hạn chế như sau**:**
  - Ma trận đánh giá còn thưa.
  - Giống như lọc dựa trên nội dung, Lọc cộng tác vẫn gặp các vấn đề khi khuyến nghị cho người dùng mới (Khởi động lạnh).

- Phương pháp này cũng không thể gợi ý được các sản phẩm mới và các sản phẩm chưa được người dùng đánh giá.
- Độ chính xác sẽ kém nếu như sở thích của người dùng thay đổi.

1.3.3. Phương pháp tiếp cận lai

Phương pháp tiếp cận lai (tên tiếng anh là Hybird Filtering) là phương pháp kết hợp các kỹ thuật khuyến nghị khác nhau [12]. Hầu hết các hương pháp tiếp cận lai đều đưa ra các dự đoán dựa trên nội dung và dựa trên cộng tác một cách riêng biệt và sau đó kết hợp chúng lại với nhau. Bằng cách thêm các tính năng của lọc dựa trên nội dung vào lọc cộng tác (có thể làm ngược lại) ta có thể đưa ra các dữ liệu thực nghiệm từ một số nghiên cứu thực nghiệm đã được chứng minh rằng hiệu suất của phương pháp tiếp cận lai có thể đưa ra các kết quả khuyến nghị chính xác hơn các phương pháp tiếp cận thuần túy [1].



Hình 1. 6. Ví dụ minh họa cho phương pháp tiếp cận lai

Trong tiếp cận lai ta có một số cách kết hợp các phương pháp như sau [1]:

- Sử dụng cả hai phương pháp lọc dựa trên nội dung và phương pháp lọc cộng tác, sau đó dùng hai kết quả thu được để quyết định:
  - Sử dụng kết quả của phương pháp nào tốt hơn (tùy thuộc vào từng thời điểm).

- Dùng cả hai kết quả để đánh giá.

- Xây dựng hệ thống lọc dựa trên nội dung sử dụng các đặc trưng của lọc cộng tác.
- Xây dựng hệ thống lọc cộng tác sử dụng các đặc trưng của lọc dựa trên nội dung.
- Xây dựng hệ thống kết hợp cả lọc dựa trên nội dung và lọc cộng tác (Hoạt động chia làm nhiều pha, mỗi pha mỗi phương pháp hoạt động độc lập với nhau).

Ngoài ra, trong thực tế các phương pháp tiếp cận lai thường được sử dụng rất đa dạng, các phương pháp lai cho hệ khuyến nghị gồm có 7 phương pháp tiến cận lai phổ biến [16]: Lai có trọng số (Weighted Hybrid); Lai chuyển đổi (Switching Hybrid); Lai trộn (Mixed Hybrid); Lai kết hợp đặc trưng (Feature Combination Hybrid); Lai theo đợt (Cascade Hybrid); Lai tăng cường đặc trưng (Feature Augmentation Hybrid); Lai meta (Meta-Level Hybrid). Phần tiếp theo trong đồ án chuyên ngành sẽ trình bày sơ lược về các phương pháp lai đã được mêu tả trên.

1.3.3.1. Lai có trọng số

Lai có trong số (Weight Hybird) đặc tả như sau: Mỗi phương pháp khuyến nghị phải đi tìm và xác định giá trị hàm phù hợp *f(u,p)* của một đối tượng $p \in P$ với người dùng $u \in U$. Tiếp cận lai có trọng số sẽ tính toán giá trị của hàm phù hợp $f_{hybird}(u,p)$ dựa trên kết quả của tất cả $f(u,p)$ của các phương pháp khuyến nghị khác tồn tại trong hệ thống. Thông thường thì hình thức lai có trọng số đơn giản nhất là kết hợp tuyến tính các giá trị phù hợp tính được từ các phương pháp khác nhau trong hệ thống [4]. Tác giả Claypool và cộng sự [17] đã xây dựng hệ thống lọc tin tức trực tuyến sử dụng khuyến nghị dựa trên nội dung và khuyến nghị dựa trên cộng tác. Kết quả cho ra trọng số ngang nhau, sau đó hệ thống sẽ dần hiệu chỉnh trọng số khi nhận được những đánh giá phản hồi từ người dùng. Tuy nhiên trọng số sẽ được điều chỉnh theo kết quả khuyến nghị của bộ phận nào đưa ra là chính xác hoặc không chính xác.

Nhìn chung ưu điểm và nhược điểm của lai có trọng số là như sau:

- **Ưu điểm:** Tất cả khả năng, phương pháp khác nhau của hệ thống được tham gia vào quá trình khuyến nghị một cách minh bạch, tự nhiên và dễ dàng thực hiện, dễ dàng điều chỉnh.
- **Nhược điểm:** Việc ước lượng trọng số lớn hay nhỏ cho phù hợp với những phương pháp khác nhau.

1.3.3.2. Lai chuyển đổi

Lai chuyển đổi (Switching Hybird) là các hệ thống khuyến nghị thuộc nhóm lai chuyển đổi thường sử dụng một số điều kiện để chuyển đổi qua lại giữa các phương pháp khuyến nghị khác nhau. Ví dụ như hệ thống DailyLearner [18] của Billsus và Pazzani. Đây là một hệ thống gợi ý tin tức cho người dùng là một hệ thống khuyến nghị sử dụng phương pháp lai chuyển đổi giữa tiếp cận nội dung và lọc cộng tác. Các tác giả đã áp dụng phương pháp lọc nội dung trước và sau đó những trường hợp là tiếp cận nội dung không thực hiện được (không thể tiếp tục thực hiện khuyến nghị và đưa ra giá trị phù hợp thấp) thì tiếp cận lọc cộng tác sẽ được áp dụng vào thay [4].

Lọc cộng tác trong phương pháp lai chuyển đổi sẽ giúp hệ thống có thể khuyến nghị được các đối tượng có nội dung, ngữ nghĩa khác với các đối tượng đã được đánh giá cao. Nói cách khác, một đối tượng có thể không được khuyến nghị với cách tiếp cận nội dung nhưng sau khi áp dụng lọc cộng tác thì đối tượng đó có thể được ưu tiên khuyến nghị [4].

Nhìn chung ưu điểm và nhược điểm của lai chuyển đổi là như sau:

- **Ưu điểm:** Phương pháp tiếp cận này rất "nhạy" với các điểm mạnh và điểm yếu của các phương pháp khác nhau.
- **Nhược điểm:** Tuy "nhạy" với điểm mạnh và điểm yếu của các phương pháp khác nhau, nhưng lai chuyển đổi yêu cầu cần phải xác định điều kiện để chuyển đổi giữa các phương pháp. Điều đó làm quá trình chuyển đổi trở nên phức tạp hơn.

1.3.3.3. Lai trộn

Tiếp cận lai trộn (Mixed Hybird) là phương pháp thực hiện các phương pháp khuyến nghị khác nhau một cách độc lập và kết hợp các kết quả từ phương pháp sẽ được chuyển thành danh sách đề xuất và được chuyển đến cho người dùng. Tiếp cận lai trộn **có** thể tránh được vấn đề người dùng mới (Khởi động lạnh – Cold Start). Giống như trường hợp trên, lọc dựa trên nội dung trong tiếp cận lai trộn cũng giúp đề xuất các đối tượng khuyến nghị mới (là các đối tượng vừa được khởi tạo, chưa có một đánh giá từ phía người dùng) trong danh sách sau cùng dựa trên mô tả nội dung của đối tượng này, trong khi đó phương pháp lọc cộng tác thông thường không thể làm được. Bù lại, lọc cộng tác trong lai trộn chỉ giúp đề xuất các đối tượgn khuyến nghị tiềm năng nhưng lại không tương tự về nội dung [4]. Các tác giả Smyth và Cotter [19] dùng cách tiếp cận này để phát triển một hệ thống khuyến nghị chương trình truyền hình phù hợp với sở thích cá nhân của người dùng, được gọi là hệ thống PTV. Với PTV, những người dùng đăng ký vào hệ thống sẽ nhận được các khuyến nghị chương truyền hình mỗi ngày thông qua Internet. PVT xây dựng hỗ trợ hồ sơ người dùng bằng cách cho người dùng tự cập nhật thông tin sở thích. Bên cạnh đó, hệ thống cũng ghi nhận lại phản hồi ý kiến từ người dùng thông qua kết quả khuyến nghị, Kết quả khuyến nghị được tập hợp, trộn kết quả từ 2 phương pháp lọc nội dung và lọc cộng tác. Chất lượng, độ chính xác khuyến nghị của hệ thống PTV được đánh giá thông qua khảo sát ý kiến người dùng.

Nhìn chung ưu điểm và nhược điểm của lai trộn là như sau:

- **Ưu điểm:** Có thể giúp đề xuất các đối tượng tiềm năng mà bản thân một phương pháp riêng biệt không thể xác định được. Trộn lọc nội dung và lọc cộng tác có thể giúp giải quyết được vấn đề khởi động lạnh (Cold Start) và cũng có thể đa dạng hóa khuyến nghị.
- **Nhược điểm:** Vì tiếp cận này sử dụng nhiều đề xuất từ các phương pháp khác nhau. Do đó hệ thống cần được xử lý, lọc các đề xuất đụng độ, trùng lắp từ các phương pháp khác nhau.

1.3.3.4. Lai kết hợp đặc trưng

Lai kết hợp đặc trưng (Feature Combination Hybird) là tiếp cận phát triển phương pháp khuyến nghị bằng cách sử dụng kết hợp thông tin đánh giá của người dùng với nội dung của đối tượng khuyến nghị [4]. Tác giả Basu và đồng nghiệp đã đề xuất tiếp cận học thuật dựa trên việc kết hợp đặc trưng để thực hiện khuyến nghị [20]. Họ đã thử nghiệm trên tập dữ liệu hơn 45000 phim và 250 người dùng. Mỗi cặp (người dùng, phim) được mã hóa thành các tập các đặc trưng bao gồm đặc trưng cộng tác (rút ra từ ma trận đánh giá) và đặc trưng của nội dung phim mô tả. Kết quả thực nghiệm cho thấy: Việc sử dụng tất cả đặc trưng về nội dung cải tiến do bao phủ (recall), nhưng không cải tiến độ chính xác (Presison), việc kết hợp đặc trưng đã cải tiến đáng kể độ chính xác và độ bao phủ so với không kết hợp đặc trưng.

Ưu điểm và nhược điểm của lai kết hợp đặc trưng là như sau:

- **Ưu điểm:** Lai kết hợp đặc trưng cho phép hệ thống xem xét dữ liệu cộng tác, nhưng không chỉ phụ thuộc duy nhất vào dữ liệu cộng tác trong ma trận đánh giá. Ngược lại, hệ thống cũng có được thông tin về sự tương tự vốn có giữa các đối tượng khuyến nghị (dựa trên đặc trưng nội dung) mà không bị ảnh hưởng bởi dữ liệu cộng tác.
- **Nhược điểm:** Khó khăn trong việc xác định các đặc trưng cộng tác và đặc trưng nội dung phù hợp.

1.3.3.5. Lai theo đợt

Lai theo đợt (Cascade Hybrid) là tiếp cận mà các phương pháp khuyến nghị khác nhau được lần lượt áp dụng theo một thứ tự ưu tiên được xác định trước tùy vào mỗi ứng dụng cụ thể.Ví dụ, phương pháp khuyến nghị thứ nhất sinh ra một danh sách xếp hạng các Ứng viên (danh sách thô). Tiếp đó, những phương pháp khác với độ ưu tiên thấp hơn sẽ được áp dụng để lọc lại danh sách thô này. Lai theo đợt giúp phương pháp thứ hai tránh những đối tượng có thể không bao giờ cần khuyến nghị vì những đối tượng này đã được lọc qua phương pháp thứ nhất. Đồng thời, các đối tượng được ưu tiên chọn với phương pháp thứ nhất sẽ được tinh lọc,

chứ không bị loại bỏ thông qua phương pháp thứ hai [4]. Ví dụ hệ thống khuyến nghị nhà hàng Entree Chicago [16] dựa trên hệ khuyến nghị tri thức FindMe để hiệu chỉnh kết quả đưa ra bởi thuật toán lọc cộng tác. Entree dùng kỹ thuật suy luận dựa trên trường hợp (case-based reasoning) để chọn và xếp hạng những nhà hàng hỗ trợ những người tham gia một hội nghị Chicago năm 1996. EntreeC là một cải tiến của Entree, áp dụng tiếp cận lai theo đợt bằng cách bổ sung thêm phương pháp lọc cộng tác để thực hiện việc tinh lọc ở đợt 2 so với đợt lọc đầu tiên dựa trên tri thức của Entree [16].

Ưu điểm và nhược điểm của lai theo đợt là như sau:

- **Ưu điểm:** So với tiếp cận lại có trọng số (Weighted Hybrid) và một số tiếp cận lai khác thì việc lọc lại danh sách thô làm cho tiếp cận này hiệu quả hơn bởi vì các phương pháp tiếp theo chỉ thực hiện lọc trên một không gian nhỏ hơn, thay vì trên cả không gian tất cả các đối tượng khuyến nghị.
- **Nhược điểm:** Khó khăn trong việc xác định độ ưu tiên giữa các phương pháp khác nhau cho mỗi ứng dụng cụ thể.

1.3.3.6. Lai tăng cường đặc trưng

Với tiếp cận lai tăng cường đặc trưng (Feature Augmentation Hybird), phương pháp đầu sẽ học một mô hình để sinh ra đặc trưng tăng cường cho đầu vào của phương pháp tiếp theo (Nhìn chung nó khá giống lai theo đợt). Nhìn chung, lai theo đợt và lai tăng cường đặc trưng đều là những tiếp cận mà hai phương pháp khác nhau sẽ được thực hiện một cách trình tự, tức là kết quả từ phương pháp thứ nhất sẽ ảnh hưởng đến phương pháp thứ hai. Tuy nhiên, về cơ bản thì hai tiếp cận lai này là hoàn toàn khác nhau. Với lai tăng cường đặc trưng thì những đặc trưng được dùng trong phương pháp thứ hai bao gồm những đặc trưng sinh ra bởi phương pháp thứ nhất, còn đối với lai theo đợt thì phương pháp thứ hai được dùng với độ ưu tiên thấp hơn phương pháp thứ nhất, nhằm lọc lại danh sách ứng viên mà phương pháp thứ nhất đã sinh ra [4]. Tác giả Mooney và Roy đã giới thiệu một hệ thống thử nghiệm LIBRA (Learning Intelligent Book Recommending Agent) dùng cơ sở dữ liệu thông tin về sách được rút trích từ trang Amazon.com cho bài toán khuyến nghị

sách [10]. Hệ thống này khai thác thông tin về những tác giả có liên quan, tiêu đề liên quan mà Amazon đã tạo ra dựa trên phương pháp lọc cộng tác. Sau đó, những thông tin này được dùng như những đặc trưng bổ sung thêm vào những đặc trưng nội dung để học hồ sơ sở thích người dùng

Ưu điểm và nhược điểm của lai tăng cường đặc trưng là như sau:

- **Ưu điểm**: Việc tăng cường đặc trưng dùng các phương pháp khác giúp hệ thống có thể cải tiến độ chính xác khuyến nghị mà không thay đổi, ảnh hưởng đến phương pháp khuyến nghị chính.
- **Nhược điểm:** Khó khăn trong việc xác định đặc trưng tăng cường phù hợp.

1.3.3.7. Lai meta

Lai meta (Meta-Level Hybird) dùng mô hình được tạo ra bởi phương pháp trước làm đầu vào cho phương pháp sau. Với lại tăng cường đặc trưng (Feature Augmentation Hybird) thì phương pháp đầu sẽ học một mô hình để sinh ra đặc trưng làm đầu vào cho phương pháp tiếp theo. Trong khi lai meta thì cả mô hình của phương pháp thứ nhất sẽ làm đầu vào cho phương pháp thứ hai. Lai meta giữa lọc nội dung và lọc cộng tác phần nào giải quyết được vẫn đề ma trận thưa trong tiếp cận lọc cộng tác bởi vì lai meta sẽ tìm kiếm những người dùng tương tự dựa trên các đặc trưng nội dung trước khi áp dụng phương pháp lọc cộng tác. Đối với những người dùng có quá ít đánh giá thì việc xác định nhóm những người đồng sở thích thông qua lọc cộng khác sẽ không được chính xác [4]. Tác giả Pazzani đã áp dụng tiếp cận lai meta để đề xuất phương pháp lọc hồ sơ người dùng và thực hiện khuyến nghị các trang web hay bài báo về lĩnh vực nhà hàng [21]. Đầu tiên, hồ sơ người dùng được học từ nhiều nguồn thông tin khác như: Thông tin cá nhân, nội dung trang web mà họ quan tâm, đánh giá và biểu diễn dưới dạng vector trọng số. Sau đó, phương pháp lọc cộng tác sẽ áp dụng để tổng hợp đánh giá từ những người dùng đồng sở thích đã xác định bởi phương pháp lọc dựa trên nội dung trước đó.

Ưu điểm và nhược điểm của lai meta là như sau:

- **Ưu điểm**: Với lai meta giữa lọc nội dung và cộng tác, phương pháp lọc cộng tác sẽ dễ dàng thực hiện tính toán trên "Dữ liệu dày" hơn so với dữ liệu thô trong ma trận đánh giá.
- **Nhược điểm:** Khó khăn trong việc chọn phương pháp để thực hiện trước. Mỗi phương pháp được chọn vẫn phải gặp những hạn chế vốn có của nó.

Tóm lại, mỗi phương pháp tiếp cận lai đều có những ưu và nhược điểm vốn có của nó. Tiếp cận lai sẽ giúp giảm bớt phần nào hạn chế của các phương pháp khác nhau và đồng thời cũng được sử dụng cho nhiều trường hợp, mục đích khác nhau.

1.4. Giới thiệu bài toán khuyến nghị phim trên các trang web xem phim

Bài toán khuyến nghị phim là một bài toán gồm các thuộc tính là sản phẩm và người dùng. Sản phẩm ở đây chính là các bộ phim và người dùng ở đây chính là những khán giả có nhu cầu thưởng thức bộ phim, và trong mỗi bộ phim đều có các thuộc tính như thể loại, thời lượng, loại phim, đánh giá, tác giả… và đối với mỗi đối tượng khán giả đều có các thuộc tính riêng biệt nhau về sở thích và nhu cầu xem phim. Bài toán khuyến nghị phim không hệ xa lạ nó đã tồn tại rất lâu trong các hệ thống xem phim nhằm hỗ trợ cho khách hàng tìm kiếm thông tin về một bộ phim mà họ yêu thích. Trong đồ án chuyên ngành này sẽ tập trung phân tích và khảo sát các phương pháp giải quyết bài toán khuyến nghị phim này, các phương pháp đó được biết đến như một thuật toán, một chuỗi phương trình nhằm giải quyết bài toán khuyến nghị phim. Và các phương pháp đó sẽ được trình bày riêng trong chương tiếp theo trong đồ án chuyên ngành này.

1.5. Kết chương 1

Chương này đã trình bày tổng quan và bài toán chung của hệ thống khuyến nghị, đồng thời trình bày chi tiết các phương pháp và các hướng tiếp cận cụ thể của từng phương pháp và cả các khảo sát có liên quan đối với từng phương pháp (thông

tin được trình bày được tham khảo từ một số tài liệu công trình luận văn thạc sĩ, tiến sĩ trong nước).

CHƯƠNG 2. MỘT SỐ PHƯƠNG PHÁP TRONG VIỆC GIẢI QUYẾT BÀI TOÁN KHUYẾN NGHỊ PHIM

2.1. PMF

2.1.1. Tổng quan

PMF (tên tiếng anh là Probabilistic Matrix Factorization) là một phương pháp dựa trên một mô hình tuyến tính với xác xuất nhiễu Gaussian [11]. PMF sử dụng một phương pháp lọc cộng tác dựa trên xác suất để lập mô hình xếp hạng của người dùng cho các sản phẩm, trong đồ án chuyên ngành này thì sản phẩm được đề cập là phim. PMF giả định rằng ma trận xếp hạng có thể được phân tích thành hai ma trận cấp thấp hơn, một cho các đặc trưng của người dùng và một cho các đặc trưng của sản phẩm. PMF sử dụng các phân phối xác suất để biểu diễn các ma trận đặc trưng và các xếp hạng quan sát được, và sử dụng các kỹ thuật học máy để ước lượng các tham số của mô hình. PMF có thể mở rộng tốt cho các tập dữ liệu lớn và thưa thớt, và có thể cải thiện hiệu suất dự đoán bằng cách sử dụng các ưu tiên thích ứng hoặc các ràng buộc về sự tương đồng của người dùng.

Trong thực tế các phương pháp tiếp cận lọc cộng tác được sử dụng trong thực tế đều thường gặp phải vấn đề không thể xử lý các tập dữ liệu rất lớn cũng như không thể dễ dàng xử lý những người dùng có rất ít xếp hạng [22]. Trong một số nghiên cứu gần đây của tác giả Ruslan Salakhutdinov, Andriy Mnih [22] đã chỉ ra rằng phương pháp PMF hoạt động tỷ lệ tuyến tính với số lượng quan sát và quan trọng hơn là hoạt động tốt trên tập dữ liệu Netflix Prize. Tập dữ liệu này được mô tả là rất lớn, thưa thớt và rất mất cân bằng [22]. Nhiều thuật toán lọc cộng tác hiện tại đã được áp dụng để lập mô hình xếp hạng của người dùng trên tập dữ liệu Netflix Prize bao gồm 480.189 người dùng, 17.770 phim và hơn 100 triệu lượt quan sát (bộ ba người dùng/phim/xếp hạng) [22]. Tuy nhiên, không có phương pháp nào trong số này tỏ ra đặc biệt thành công vì hai lý do: Đầu tiên, không có cách tiếp cận nào hiện

tại có thể lập chính xác được mô hình xếp hạng người dùng, ngoại trừ các cách tiếp cận dựa trên hệ số ma trận vì chúng có khả năng mở rộng tốt cho các tập dữ liệu lớn; Thứ hai, hầu hết các thuật toán hiện tại đều gặp khó khăn trong việc đưa ra dự đoán chính xác cho những người dùng có rất ít xếp hạng [22]. Thực tiễn phổ biến trong hầu hết phương pháp lọc cộng tác là xóa tất cả người dùng có ít hơn một số xếp hạng tối thiểu, do đó kết quả được báo cáo trên các bộ dữ liệu tiêu chuẩn chẳng hạn như MovieLens và EachMovie trông có vẻ ấn tượng vì những trường hợp khó nhất đã được loại bỏ. Ví dụ: tập dữ liệu Netflix rất mất cân bằng, với người dùng "không thường xuyên" xếp hạng dưới 5 phim, trong khi người dùng "thường xuyên" xếp hạng trên 10.000 phim [22].

2.1.2. Mô hình bài toán

2.1.2.1. PMF không ràng buộc

Giả sử chúng ta có *M* bộ phim, *N* người dùng và giá trị xếp hạng bằng số nguyên dương trong khoảng 1 đến *K*. Hãy cho rằng R_{ij} là giá trị biểu thị xếp hạng của người dùng *i* đối với phim *j*, $U \in R^{DxN}$ và $V \in R^{DxM}$ là ma trận vector đặc trưng giữa phim và người dùng, với từng cột trong vector U_i và V_i lần lượt biểu thị là vector đặc trưng của người dùng cụ thể *i* và phim cụ thể *j* [22]. Hiệu suất của mô hình PMF được đo bằng cách tính sai số bình phương trung bình gốc (gọi tắt là RMSE) trên tập kiểm tra, một số nghiên cứu đi trước [22] đã áp dụng mô hình tuyến tính xác xuất với nhiễu quan sát Gaussian. Công thức định nghĩa sự phân bố có điều kiện trên các xếp hạng được quan sát:

$$p(R \mid U, V, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\mathcal{N}\left(R_{ij} \mid U_i^T V_j, \sigma^2\right)\right]^{I_{ij}} \qquad (2.1.1)$$

Trong đó $\mathcal{N}(x|\mu, \sigma^2)$ là hàm mật độ xác xuất của phân bố Gaussian với giá trị trung bình μ với phương sai σ^2 và và I_{ij} là hàm chỉ thị bằng 1 nếu người dùng *i* xếp hạng phim *j* và bằng 0 nếu ngược lại [22]. Một số công thức đặc trưng vector ở người dùng và phim:

$$p(U \mid \sigma_U^2) = \prod_{i=1}^{N} \mathcal{N}(U_i \mid 0, \sigma_U^2 \mathbf{I}) \tag{2.1.2}$$

$$p(V \mid \sigma_V^2) = \prod_{j=1}^{M} \mathcal{N}\left(V_j \mid 0, \sigma_V^2 \mathbf{I}\right) \tag{2.1.3}$$

Công thức tính log phân phối sau đối với đặc trưng của người dùng và phim:

$$\ln p(U, V \mid R, \sigma^2, \sigma_V^2, \sigma_U^2) = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} \left(R_{ij} - U_i^T V_j\right)^2 - \frac{1}{2\sigma_U^2} \sum_{i=1}^{N} U_i^T U_i$$
$$-\frac{1}{2\sigma_V^2} \sum_{j=1}^{M} V_j^T V_j - \frac{1}{2}\left(\left(\sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij}\right) \ln \sigma^2 + \mathrm{ND} \ln \sigma_U^2 + \mathrm{MD} \ln \sigma_V^2\right) + C \tag{2.1.4}$$

Trong đó C là hằng số không phụ thuộc vào các tham số. Và cực đại hóa log-posterior (Log-posterior là một khái niệm trong thống kê Bayes, nó là hàm logarit của xác suất hậu nghiệm. Xác suất hậu nghiệm là xác suất của một sự kiện sau khi đã có thêm thông tin mới.) trên các đặc trưng phim và người dùng với các siêu tham số (tức là phương sai nhiễu quan sát và phương sai tiên nghiệm) được giữ cố định tương đương với việc tối thiểu hóa hàm mục tiêu tổng bình phương sai số với các điều kiện chuẩn hóa bậc hai như sau [22]:

$$E = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} \left(R_{ij} - U_i^T V_j\right)^2 + \frac{\lambda_U}{2} \sum_{i=1}^{N} \|U_i\|_{Fro}^2 + \frac{\lambda_V}{2} \sum_{j=1}^{M} \|V_j\|_{Fro}^2 \tag{2.1.5}$$

Trong đó $\lambda_U = \sigma^2/\sigma_U^2, \lambda_V = \sigma^2/\sigma_V^2$ và $\|\cdot\|_{Fro}^2$ biểu thị chuẩn Frobenius (Chuẩn Frobenius của một ma trận là căn bậc hai của tổng các bình phương của tất cả các phần tử trong ma trận đó). Bằng cách sử dụng phương pháp Gradient, một giá trị cực tiểu được sinh ra bởi công thức trên có thể được tìm thấy bằng cách giảm giá trị của *U* và *V* [22]. Trong một số nghiên cứu gần đây về về PMF, tác giả Ruslan Salakhutdinov và Andriy Mnih [22] đã đưa ra nhận định rằng thay vì sử dụng một mô hình tuyến tính Gaussian đơn giản, ta có thể tạo ra các dự đoán ngoài phạm vi

của các giá trị đánh giá hợp lệ bằng cách tích vô hướng giữa các vector đặc trưng cụ thể cho người dùng và phim được truyền qua hàm logistic $g(x) = 1/(1 + \exp(-x))$, giới hạn phạm vi của các dự đoán được thể hiện qua công thức sau:

$$p(R \mid U, V, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\mathcal{N}\left(R_{ij} \mid g\left(U_i^T V_j\right), \sigma^2\right)\right]^{I_{ij}} \quad (2.1.6)$$

Công thức ánh xạ các đánh giá 1…K vào khoảng [0,1] bằng cách sử dụng hàm $t(x) = (x - 1)/(K - 1)$, để phạm vi của các giá trị đánh giá hợp lệ khớp với phạm vi của các dự đoán mà mô hình 2 tác giả trên tạo ra. Việc giảm thiểu hàm mục tiêu được đưa ra ở trên bằng cách sử dụng phương pháp giảm dốc cần có thời gian tuyến tính theo số lượng quan sát. Bằng cách này việc triển khai một mô hình huấn luyện trên matlab cho tập dữ liệu của netflix chỉ tốn 1 giờ cho việc huấn luyện 30 mô hình [22].

2.1.2.2. PMF ràng buộc

PMF ràng buộc là một cách bổ sung để ràng buộc các vector đặc trưng của người dùng, đặc biệt có ảnh hưởng mạnh với người dùng có ít đánh giá. Cách này giả định rằng những người dùng có rất ít đánh giá sẽ có các vector đặc trưng gần với giá trị trung bình tiên nghiệm (hay còn gọi là người dùng trung bình), do đó các đánh giá dự đoán cho những người dùng này sẽ gần với đánh giá trung bình của phim. Ta gọi $W \in R^{D \times M}$ là ma trận ràng buộc tương tự tiềm ẩn. Ta định nghĩa vector đặc trưng cho người dùng *i* qua công thức sau [22]:

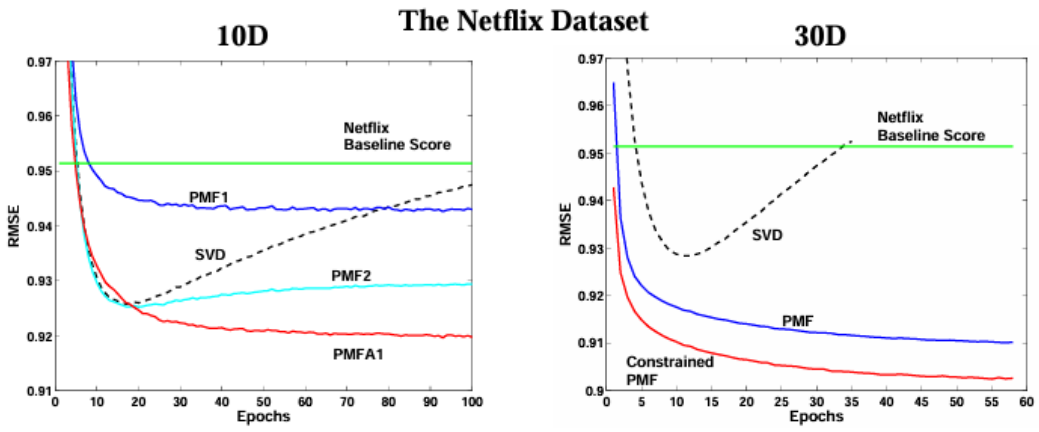
$$U_i = Y_i + \frac{\sum_{k=1}^{M} I_{ik} W_k}{\sum_{k=1}^{M} I_{ik}} \quad (2.1.7)$$

Trong đó *I* là ma trận chỉ số quan sát với I_{ij} có giá trị bằng 1 nếu người dùng *i* đánh giá phim *j* và bằng 0 nếu ngược lại. Theo một cách trực quan nhất, cột thứ i của ma trận W thể hiện ảnh hưởng của việc một người dùng đã đánh giá một phim cụ thể lên giá trị trung bình tiên nghiệm của vector đặc trưng của người dùng đó. Kết quả là, những người dùng đã xem các phim giống nhau (hoặc tương tự) sẽ có các phân phối tiên nghiệm tương tự cho các vector đặc trưng của họ. Lưu ý rằng Y_i

có thể được coi là độ lệch được cộng vào giá trị trung bình của phân phối tiên nghiệm để thu được vector đặc trưng U_i cho người dùng i. Trong mô hình PMF không ràng buộc, U_i và Y_i là bằng nhau vì giá trị trung bình tiên nghiệm được cố định bằng 0. Công thức xác định phân phối điều kiện các đánh giá như sau [22]:

$$p(R \mid Y, V, W, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\mathcal{N}\left(R_{ij} \mid g\left(\left[Y_i + \frac{\sum_{k=1}^{M} I_{ik} W_k}{\sum_{k=1}^{M} I_{ik}} \right]^T V_j \right), \sigma^2 \right) \right]^{I_{ij}} \quad (2.1.8)$$

Một số kết quả thực nghiệm của 2 tác giả Ruslan Salakhutdinov và Andriy Mnih [22] ở hình dưới đã cho thấy rằng PMF ràng buộc hoạt động tốt hơn nhiều so với PMF không ràng buộc, đặc biệt là trên những người dùng có ít đánh giá.



Hình 2. 1. Kết quả thực nghiệm thu được bằng cách sử dụng PMF (Bản số 1)

Mô tả hình 2.1: Bảng bên trái: Hiệu suất của SVD, PMF không ràng buộc sử dụng các vector đặc trưng 10D trên toàn bộ dữ liệu kiểm tra của Netflix; Bảng bên phải: Hiệu suất của SVD, PMF ràng buộc sử dụng các vector đặc trưng 30D trên dữ liệu kiểm tra của Netflix. Trục y hiển thị RMSE (sai số bình phương trung bình căn), và trục x cho biết số lần lặp, hay lượt, qua toàn bộ tập dữ liệu huấn luyện [22].

2.1.3. Thực nghiệm và đánh giá

2.1.3.1. Mô tả tập dữ liệu thực nghiệm

Tập dữ liệu thực nghiệm trong đồ án chuyên ngành được tham khảo từ tập thử nghiệm trong đề tài [22] của tác giả Ruslan Salakhutdinov và Andriy Mnih nhằm mô tả tính thực nghiệm của phương pháp PMF trong dữ liệu thực. Theo tập dữ liệu đến từ Netflix, dữ liệu được thu thập từ tháng 10 năm 1998 đến tháng 12 năm 2005 và đại diện cho phân bố của tất cả các đánh giá mà Netflix nhận được trong khoảng thời gian này. Tập dữ liệu huấn luyện bao gồm 100.480.507 đánh giá từ 480.189 người dùng được chọn ngẫu nhiên và vô danh trên 17.770 tựa phim. Đồng thời, Netflix cũng cung cấp dữ liệu kiểm tra chứa khoảng 1.408.395 đánh giá. Ngoài ra, Netflix cũng cung cấp một tập dữ liệu thử nghiệm chứa 2.817.131 cặp người dùng/phim với các đánh giá bị giấu đi. Các cặp này được chọn từ các đánh giá gần đây nhất cho một tập hợp con của các người dùng trong tập dữ liệu huấn luyện. Để giảm tối thiểu trình trạng quá tải hiệu suất ngoài ý muốn cho các bộ thử nghiệm trên học máy, Netflix đã hỗ trợ sai số bình phương trung bình căn (RMSE) trên một nửa không biết của tập dữ liệu thử nghiệm. Do đó để làm mốc cho đánh giá thực nghiệm, Netflix đã cung cấp điểm số thử nghiệm của hệ thống của riêng họ được huấn luyện trên cùng một dữ liệu, là 0,9514 [22].

Và để cung cấp thêm thông tin về hiệu suất của các thuật toán khác nhau, Ruslan Salakhutdinov và Andriy Mnih đã tạo ra một tập dữ liệu nhỏ hơn và khó hơn nhiều từ dữ liệu của Netflix bằng cách chọn ngẫu nhiên 50.000 người dùng và 1.850 phim. Tập dữ liệu thực nghiệm chứa 1.082.982 cặp người dùng/phim huấn luyện và 2.462 cặp kiểm tra. Bên cạnh đó hơn 50% người dùng trong tập dữ liệu huấn luyện có ít hơn 10 đánh giá [22].

Để tăng tốc độ huấn luyện mô hình thay vì thực hiện học theo lô, Ruslan Salakhutdinov và Andriy Mnih đã chia nhỏ dữ liệu của Netflix thành các lô nhỏ có kích thước 100.000 (cặp người dùng/phim/đánh giá) và cập nhật các vector đặc trưng sau mỗi 5 lô nhỏ. Sau khi thử nghiệm nhiều giá trị khác nhau cho tốc độ học

và động lượng, Ruslan Salakhutdinov và Andriy Mnih chọn sử dụng tốc độ học là 0,005 và động lượng là 0,9, vì họ cho rằng cài đặt này hoạt động tốt cho tất cả các giá trị của *D* mà họ đã thử nghiệm [22].

2.1.3.2. Kết quả thực nghiệm của PMF không ràng buộc

Để đánh giá hiệu suất của mô hình PMF không ràng buộc, Ruslan Salakhutdinov và Andriy Mnih [22] sử dụng các mô hình có 10D đặc trưng. Số chiều này được chọn để minh họa rằng ngay cả khi số chiều của các đặc trưng khá thấp thì các mô hình giống SVD vẫn có thể quá khớp và có thể cải thiện hiệu suất bằng cách điều chuẩn các mô hình một cách tự động. Ruslan Salakhutdinov và Andriy Mnih đã so sánh tập các mô hình gồm một mô hình SVD, hai mô hình PMF tiên nghiệm cố định, và hai mô hình PMF có tiên nghiệm thích nghi. Mô hình SVD được huấn luyện để tối thiểu hóa khoảng cách bình phương chỉ đối với các phần tử quan sát được của ma trận mục tiêu. Các vector đặc trưng của mô hình SVD không được điều chuẩn theo bất kỳ cách nào. Hai mô hình PMF không ràng buộc khác nhau về các tham số điều chuẩn: một (PMF1) có λ_U = 0,01 và λ_V= 0,001, trong khi đó (PMF2) có λ_U = 0,001 và λ_V = 0,0001. Mô hình PMF đầu tiên với tiên nghiệm thích nghi (PMFA1) có tiên nghiệm Gaussian với các ma trận hiệp phương sai cầu trên các vector đặc trưng của người dùng và phim, trong khi mô hình thứ hai (PMFA2) có các ma trận hiệp phương sai chéo. Trong cả hai trường hợp, các tiên nghiệm thích nghi có các giá trị trung bình có thể điều chỉnh. Các tham số tiên nghiệm và phương sai nhiễu được cập nhật sau mỗi 10 và 100 lần cập nhật ma trận đặc trưng tương ứng. Các mô hình được so sánh dựa trên RMSE trên tập dữ liệu kiểm tra [22].

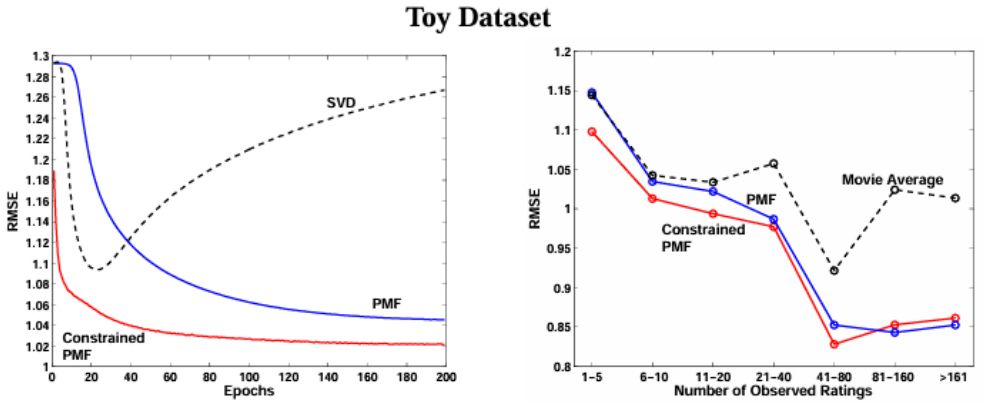
Kết quả của việc so sánh được hiển thị trên **Hình 2.1** (bảng bên trái). Lưu ý rằng đường cong cho mô hình PMF với các ma trận hiệp phương sai cầu không được hiển thị vì nó gần như giống với đường cong cho mô hình với các ma trận hiệp phương sai chéo. So sánh các mô hình dựa trên RMSE thấp nhất đạt được trong quá trình huấn luyện, Ruslan Salakhutdinov và Andriy Mnih thấy rằng mô hình SVD hoạt động gần như tốt bằng mô hình PMF được điều chuẩn vừa phải (PMF2)

(0,9258 so với 0,9253) trước khi quá khớp xấu vào cuối quá trình huấn luyện. Trong khi PMF1 không quá khớp, nó rõ ràng là thiếu khớp vì nó chỉ đạt RMSE là 0,9430. Các mô hình với tiên nghiệm thích nghi rõ ràng vượt trội so với các mô hình cạnh tranh, đạt RMSE là 0,9197 (ma trận hiệp phương sai chéo) và 0,9204 (ma trận hiệp phương sai cầu). Những kết quả này cho thấy việc điều chuẩn tự động thông qua các tiên nghiệm thích nghi hoạt động tốt trong thực tế. Hơn nữa, kết quả sơ bộ thực nghiệm của Ruslan Salakhutdinov và Andriy Mnih cho các mô hình với các vector đặc trưng có số chiều cao hơn cho thấy khoảng cách về hiệu suất do sử dụng các tiên nghiệm thích nghi có thể tăng lên khi số chiều của các vector đặc trưng tăng lên. Trong khi việc sử dụng các ma trận hiệp phương sai chéo không mang lại sự cải thiện đáng kể so với các ma trận hiệp phương sai cầu, các ma trận hiệp phương sai chéo có thể phù hợp để điều chuẩn tự động phiên bản tham lam của thuật toán huấn luyện PMF, nơi các vector đặc trưng được học theo từng chiều [22].

2.1.3.3. Kết quả thực nghiệm của PMF ràng buộc

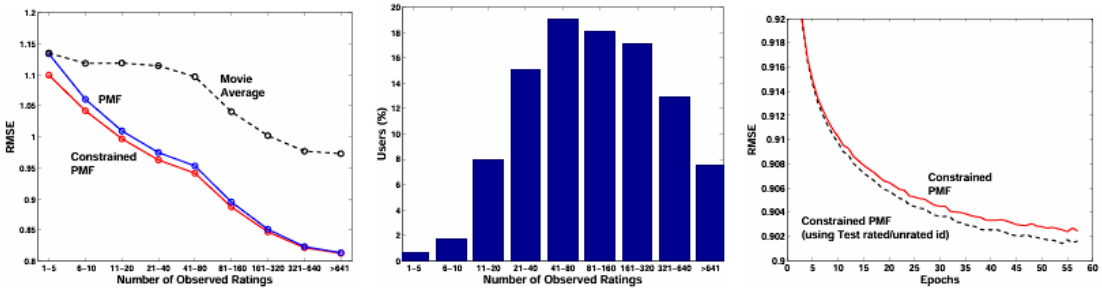
Đối với các thí nghiệm liên quan đến mô hình PMF ràng buộc, Ruslan Salakhutdinov và Andriy Mnih [22] sử dụng các đặc trưng 30D (D = 30), vì lựa chọn này mang lại hiệu suất mô hình tốt nhất trên tập dữ liệu kiểm tra. Các giá trị của D trong khoảng [20, 60] cho kết quả tương tự. Kết quả hiệu suất của SVD, PMF không ràng buộc và PMF ràng buộc trên tập dữ liệu thực nghiệm được hiển thị trên **Hình 2.2**. Các vector đặc trưng được khởi tạo với các giá trị giống nhau trong tất cả ba mô hình. Đối với cả hai mô hình PMF không ràng buộc và PMF ràng buộc, các tham số điều chuẩn được đặt là $\lambda_U = \lambda_Y = \lambda_V = \lambda_W = 0.002$. Rõ ràng là mô hình SVD quá phù hợp. Mô hình PMF ràng buộc hoạt động tốt hơn nhiều và hội tụ nhanh hơn rất nhiều so với mô hình PMF không ràng buộc. **Hình 2.2** (bảng bên phải) cho thấy ảnh hưởng của việc ràng buộc các đặc trưng cụ thể cho người dùng lên các dự đoán cho những người dùng ít đánh giá. Hiệu suất của mô hình PMF cho một nhóm người dùng có ít hơn 5 đánh giá trong các tập dữ liệu huấn luyện gần như giống với thuật toán trung bình phim vì luôn dự đoán điểm số tru ng bình của mỗi phim. Tuy nhiên, mô hình PMF ràng buộc hoạt động tốt hơn đáng kể đối với những

người dùng có ít đánh giá. Khi số lượng đánh giá tăng lên, cả PMF không ràng buộc và PMF ràng buộc đều có hiệu suất tương tự [22].



Hình 2. 2. Kết quả thực nghiệm thu được bằng cách sử dụng PMF (Bản số 2)

Mô tả cho hình 2.2: Bảng bên trái: Hiệu suất của SVD, PMF không ràng buộc và PMF có ràng buộc trên dữ liệu xác thực. Trục y hiển thị RMSE (sai số bình phương trung bình), và trục x cho biết số lần lặp qua toàn bộ tập dữ liệu huấn luyện; Bảng bên phải: Hiệu suất của PMF có ràng buộc, PMF không ràng buộc và thuật toán trung bình phim (Movie Average) luôn dự đoán điểm số trung bình của mỗi phim. Người dùng được nhóm theo số lượng đánh giá được quan sát trong tập dữ liệu huấn luyện [22].



Hình 2. 3. Kết quả thực nghiệm thu được bằng cách sử dụng PMF (Bản số 3)

Mô tả cho hình 2.3: Bảng bên trái: Hiệu suất của PMF có ràng buộc, PMF không ràng buộc và thuật toán trung bình phim (Movie Average) luôn dự đoán điểm số trung bình của mỗi phim. Người dùng được nhóm theo số lượng đánh giá được quan sát trong tập dữ liệu huấn luyện, với trục x cho biết các nhóm đó, và trục y cho biết RMSE trên tập dữ liệu xác thực Netflix đầy đủ cho mỗi nhóm; Bảng bên giữa:

Phân bố người dùng trong tập dữ liệu huấn luyện; Bảng bên phải: Hiệu suất của PMF có ràng buộc và PMF có ràng buộc sử dụng thông tin đánh giá/không đánh giá bổ sung thu được từ tập dữ liệu kiểm tra [22].

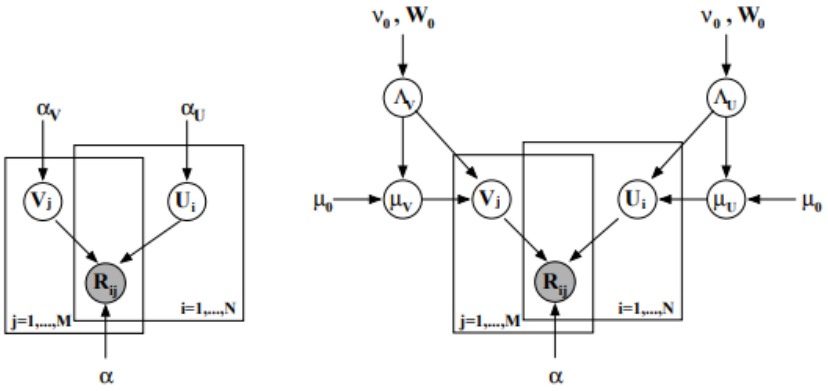
Kết quả thực nghiệm trên toàn bộ dữ liệu Netflix tương tự như kết quả trên tập dữ liệu trên hình 2.2. Đối với cả hai mô hình PMF không ràng buộc và PMF có ràng buộc, các tham số điều chuẩn được đặt là $\lambda_U = \lambda_Y = \lambda_V = \lambda_W = 0.001$. Hình 2.1 (bảng bên phải) cho thấy PMF có ràng buộc vượt trội hơn nhiều so với mô hình PMF không có ràng buộc, đạt được RMSE là 0.9016. Một mô hình SVD đơn giản đạt được RMSE khoảng 0.9280 và sau khoảng 10 lần lặp bắt đầu quá khớp. Hình 2.3 (bảng bên trái) cho thấy mô hình PMF có ràng buộc có khả năng tổng quát hóa tốt hơn nhiều cho những người dùng có ít đánh giá. Lưu ý rằng hơn 10% người dùng trong tập dữ liệu huấn luyện có ít hơn 20 đánh giá. Bên cạnh đó netflix cho chúng ta biết trước những cặp người dùng/phim xuất hiện trong tập kiểm tra, vì vậy chúng ta có một danh mục bổ sung: những bộ phim đã được xem nhưng không biết điểm số. Đây là một nguồn thông tin quý giá về những người dùng xuất hiện nhiều lần trong tập kiểm tra, đặc biệt là nếu họ chỉ có một số lượng nhỏ đánh giá trong tập huấn luyện. Mô hình PMF có ràng buộc có thể dễ dàng sử dụng thông tin này. Hình 2.3 (bảng bên phải) cho thấy nguồn thông tin bổ sung này cải thiện thêm hiệu suất của mô hình [22].

2.2. BPMF

2.2.1. Tổng quan

Bayesian Probabilistic Matrix Factorization (BPMF) là một phương pháp tiếp cận Bayes hoàn toàn cho bài toán phân tích nhân tố ma trận xác suất (PMF). PMF là một mô hình hóa sở thích của người dùng dựa trên tích vô hướng của hai vector đặc trưng người dùng và sản phẩm. BPMF khác với PMF ở chỗ nó không chỉ tối đa hóa hậu nghiệm trên các vector đặc trưng với các siêu tham số cố định mà còn đặt các tiên nghiệm liên hợp cho các vector đặc trưng và các siêu tham số và sử

dụng phương pháp Markov chain Monte Carlo (MCMC) để tính toán xấp xỉ phân phối hậu nghiệm. BPMF có thể kiểm soát tự động khả năng của mô hình và cung cấp phân phối dự đoán cho các giá trị đánh giá mới. BPMF được cho là có hiệu quả dự đoán cao hơn so với PMF được huấn luyện bằng ước lượng điểm tối đa hậu nghiệm (MAP).



Hình 2. 4. Hình bên trái là mộ hình đồ thị PMF, Hình bên phải là mô hình đồ thị BPMF.

2.2.2. Mô hình bài toán

2.2.2.1. Lý thuyết

Mô hình đồ thị biểu diễn BPMF được thể hiện trong hình 2.4 (Hình bên phải). Cũng giống như PMF, khả năng đánh giá quan sát được biểu diễn bởi công thức bên dưới [23]:

$$p(R \mid U, V, \sigma) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\mathcal{N}\left(R_{ij} \mid g\left(U_i^T V_j\right), \sigma^{-1}\right)\right]^{I_{ij}} \quad (2.2.1)$$

Các phân phối tiên nghiệm trên vector đặc trưng của người dùng và phim được cho bởi 2 công thức sau [23]:

$$p(U \mid \mu_U, \Lambda_U) = \prod_{i=1}^{N} \mathcal{N}(U_i \mid \mu_U, \Lambda_U^{-1}) \quad (2.2.2)$$

$$p(V \mid \mu_V, \Lambda_V) = \prod_{i=1}^{M} \mathcal{N}(V_i \mid \mu_V, \Lambda_V^{-1}) \quad (2.2.3)$$

Các tiên nghiệm Gaussian-Wishart cũng được định nghĩa trên các siêu tham số $\Theta_U = \{\mu_U \mid \Lambda_U\}$ và $\Theta_V = \{\mu_V \mid \Lambda_V\}$ và được củng cố bằng 2 công thức bên dưới [23]:

$$\begin{aligned} p(\Theta_U \mid \Theta_0) &= p(\mu_U \mid \Lambda_U)p(\Lambda_U) \\ &= \mathcal{N}(\mu_U \mid \mu_0, (\beta_0\Lambda_U)^{-1})\mathcal{W}(\Lambda_U \mid W_0, \nu_0) \end{aligned} \tag{2.2.4}$$

$$\begin{aligned} p(\Theta_V \mid \Theta_0) &= p(\mu_V \mid \Lambda_U)p(\Lambda_V) \\ &= \mathcal{N}(\mu_V \mid \mu_0, (\beta_0\Lambda_V)^{-1})\mathcal{W}(\Lambda_V \mid W_0, \nu_0) \end{aligned} \tag{2.2.5}$$

Trong đó, W nghĩa là Wishart với v_0 bậc tự do và ma trận tỷ lệ D × D là W_0 [23]:

$$\mathcal{W}(\Lambda \mid W_0, \nu_0) = \frac{1}{C}|\Lambda|^{\frac{(\nu_0 - D - 1)}{2}} \exp\left(-\frac{1}{2}Tr(W_0^{-1}\Lambda)\right) \tag{2.2.6}$$

Trong đó, C là hằng số chuẩn hóa. Để thuận tiện cho việc nghiên cứu tác giả của nghiên cứu tác giả của công thức trên đã định nghĩa $\Theta_U = \{\mu_0, v_0, W_0\}$. Họ cũng đã đặt $v_0 = D$ và W_0 là ma trận đơn vị cho cả hai siêu tham số người dùng và phim và đồng thời chọn $\mu_0 = 0$ theo tính đối xứng [23].

Giả định: Ta có phân phối dự đoán của giá trị đánh giá R_{ij} cho người dùng i và phim j được thu được bằng cách tích phân theo các tham số mô hình và siêu tham số theo công thức như sau [23]:

$$\begin{aligned} p\left(R_{ij}^* \mid R, \Theta_0\right) = \iint p\left(R_{ij}^* \mid U_i, V_j\right)p(U, V \mid R, \Theta_U, \Theta_V) \\ p(\Theta_U, \Theta_V \mid \Theta_0)\mathrm{d}\{U, V\}\mathrm{d}\{\Theta_U, \Theta_V\}. \end{aligned} \tag{2.2.7}$$

Tuy nhiên,việc đánh giá chính xác phân phối dự đoán này là không khả thi phân tích do sự phức tạp của hậu nghiệm, chúng ta cần phải sử dụng suy luận xấp xỉ. Một lựa chọn là sử dụng các **phương pháp biến thiên** để cung cấp các kịch bản xấp xỉ xác định cho các hậu nghiệm. Cụ thể, chúng ta có thể xấp xỉ hậu nghiệm thực $p(U, V, \Theta_U, \Theta_V \mid R)$ bằng một phân phối có tính chất phân tích, với mỗi yếu tố có một dạng tham số cụ thể như phân phối Gaussian. Hậu nghiệm xấp xỉ này sẽ cho phép chúng ta xấp xỉ các tích phân trong công thức số 7 trên. Các phương pháp biến thiên đã trở thành phương pháp lựa chọn, vì chúng thường có thể mở rộng tốt cho

các ứng dụng quy mô lớn. Tuy nhiên, chúng có thể tạo ra kết quả không chính xác vì chúng có xu hướng liên quan đến các xấp xỉ quá đơn giản cho hậu nghiệm [23].

Các phương pháp dựa trên MCMC (Markov Chain Monte Carlo) thì ngược lại, chúng sử dụng phép tính gần đúng Monte Carlo cho phân phối dự đoán của công thức số 7 như sau [23]:

$$p\left(R_{ij}^{*} \mid R, \Theta_0\right) \approx \frac{1}{K} \sum_{k=1}^{K} p\left(R_{ij}^{*} \mid U_i^{(k)}, V_j^{(k)}\right) \tag{2.2.8}$$

Các mẫu $\{U_i^{(k)}, V_j^{(k)}\}$ được sinh ra bằng cách chạy một đoạn Markov có phân phối cân bằng là phân phối hậu nghiệm trên các tham số và siêu tham số mô hình U, V, Θ_U, Θ_V . Lợi thế của các phương pháp dựa trên Monte Carlo là chúng cho kết quả chính xác khi tiến tới giới hạn. Tuy nhiên, trong thực tế, các phương pháp MCMC thường được coi là rất tốn thời gian tính toán nên việc sử dụng chúng bị giới hạn ở các vấn đề quy mô nhỏ [23].

2.2.2.2. Giải thuật và mã giả

Một trong những thuật toán MCMC (Markov Chain Monte Carlo) đơn giản nhất là thuật toán lấy mẫu Gibbs (Gibbs Sampling Algorithm), nó hoạt động bằng cách lặp lại qua các biến tiềm ẩn, lấy mẫu từng biến từ phân phối có điều kiện dựa trên các giá trị hiện tại của tất cả các biến khác. Thuật toán lấy mẫu Gibbs thường được sử dụng khi các phân phối có điều kiện này có thể được lấy mẫu một cách dễ dàng [23].

Do sử dụng các tiên nghiệm liên hợp cho các tham số và siêu tham số trong mô hình BPMF, các phân phối có điều kiện được suy ra từ phân phối hậu nghiệm rất dễ để lấy mẫu. Cụ thể hơn, phân phối có điều kiện trên vector đặc trưng người dùng U_i, có điều kiện trên các đặc trưng phim, ma trận đánh giá người dùng quan sát R, và các giá trị của siêu tham số là Gaussian [23]:

$$\begin{aligned} p(U_i \mid R, V, \Theta_U, \alpha) &= \mathcal{N}(U_i \mid \mu_i^{*}, [\Lambda_i^{*}]^{-1}) \\ &\sim \prod_{j=1}^{M} \left[\mathcal{N}\left(R_{ij} \mid U_i^T V_j, \alpha^{-1}\right)\right]^{I_{ij}} p(U_i \mid \mu_U, \Lambda_U), \end{aligned} \tag{2.2.9}$$

Trong đó, Λ_i^* và μ_i^* được định nghĩa như sau:

$$\Lambda_i^* = \Lambda_U + \alpha \sum_{j=1}^{M} \left[V_j V_j^T\right]^{I_{ij}} \tag{2.2.10}$$

$$\mu_i^* = [\Lambda_i^*]^{-1} \left(\alpha \sum_{j=1}^{M} \left[V_j R_{ij}\right]^{I_{ij}} + \Lambda_U \mu_U \right) \tag{2.2.11}$$

Lưu ý rằng phân phối có điều kiện trên ma trận đặc trưng người dùng tiềm ẩn U được phân tích thành tích của các phân phối có điều kiện trên các vector đặc trưng người dùng riêng lẻ như sau [23]:

$$p(U \mid R, V, \Theta_U) = \prod_{i=1}^{N} p(U_i \mid R, V, \Theta_U) \tag{2.2.12}$$

Do đó, chúng ta có thể dễ dàng tăng tốc bộ lấy mẫu bằng cách lấy mẫu từ các phân phối có điều kiện này một cách song song. Sự tăng tốc có thể rất đáng kể, đặc biệt khi số lượng người dùng lớn [23].

Công thức phân phối có điều kiện trên các siêu tham số người dùng có điều kiện trên ma trận đặc trưng người dùng U được cho bởi phân phối Gaussian-Wishart như sau [23]:

$$\begin{aligned} &p(\mu_U, \Lambda_U \mid U, \Theta_0) = \\ &\mathcal{N}(\mu_U \mid \mu_0^*, (\beta_0^* \Lambda_U)^{-1}) \mathcal{W}(\Lambda_U \mid W_0^*, \nu_0^*) \end{aligned} \tag{2.2.13}$$

Trong đó, các biến được mô tả như sau:

$$\mu_0^* = \frac{\beta_0 \mu_0 + N\overline{U}}{\beta_0 + N}, \quad \beta_0^* = \beta_0 + N, \quad \nu_0^* = \nu_0 + N,$$

$$[W_0^*]^{-1} = W_0^{-1} + N\bar{S} + \frac{\beta_0 N}{\beta_0 + N}(\mu_0 - \overline{U})(\mu_0 - \overline{U})^T$$

$$\overline{U} = \frac{1}{N}\sum_{i=1}^{N} U_i \bar{S} = \frac{1}{N}\sum_{i=1}^{N} U_i U_i^T.$$

Mã giả cho thuật toán lấy mẫu Gibbs được mô tả như sau [23]:

1. Khởi tạo các tham số mô hình $\{U^1, V^1\}$
2. Vòng lặp For: t = 1…, T
   - Lấy mẫu siêu tham số (Công thức 13)
     - $\Theta_U^t \sim p(\Theta_U | U^T, \Theta_0)$
     - $\Theta_V^t \sim p(\Theta_V | V^T, \Theta_0)$
   - Với mỗi i = 1…, N mẫu đặc trưng của người dùng (Công thức 9)
     - $U_i^{t+1} \sim p(U_i | R, V^t, \Theta_U^t)$
   - Với mỗi i = 1…, N mẫu đặc trưng của phim
     - $V_i^{t+1} \sim p(V_i | R, U^{t+1}, \Theta_V^t)$

2.2.3. Quá trình và đánh giá thực nghiệm

2.2.3.1. Mô tả tập dữ liệu thực nghiệm

Tập dữ liệu thực nghiệm trong đồ án chuyên ngành được tham khảo từ tập thử nghiệm trong đề tài [23] của tác giả Ruslan Salakhutdinov và Andriy Mnih nhằm mô tả tính thực nghiệm của phương pháp BPMF trong dữ liệu thực và đồng thời trong phần này cũng so sánh mức độ hiệu quả giữa PMF và BPMF trong thực tế. Bộ dữ liệu được thu thập bởi Netflix, đại diện cho phân bố của tất cả các đánh giá Netflix nhận được từ tháng 10 năm 1998 đến tháng 12 năm 2005. Bộ dữ liệu huấn luyện bao gồm 100.480.507 đánh giá từ 480.189 người dùng được chọn ngẫu nhiên, ẩn danh trên 17.770 tựa phim. Bên cạnh đó, Netflix cũng cung cấp thêm dữ liệu kiểm tra, chứa 1.408.395 đánh giá. Ngoài ra, Netflix cũng cung cấp một tập dữ liệu kiểm tra chứa 2.817.131 cặp người dùng/phim với các đánh giá được giữ lại. Các cặp này được chọn từ các đánh giá gần đây nhất từ một tập hợp con của các người dùng trong tập dữ liệu huấn luyện. Để giảm tối thiểu trình trạng quá tải hiệu suất ngoài ý muốn cho các bộ thử nghiệm trên học máy, Netflix đã hỗ trợ sai số bình phương trung bình căn (RMSE) trên một nửa không biết của tập dữ liệu thử nghiệm. Do đó để làm mốc cho đánh giá thực nghiệm, Netflix đã cung cấp điểm số thử nghiệm của hệ thống của riêng họ được huấn luyện trên cùng một dữ liệu, là 0,9514.

2.2.3.2. Mô tả quá trình huấn luyện mô hình PMF

Để so sánh, Ruslan Salakhutdinov và Andriy Mnih đã huấn luyện nhiều mô hình PMF tuyến tính sử dụng MAP, chọn các tham số điều chuẩn bằng cách sử dụng tập kiểm định. Ngoài các mô hình PMF tuyến tính, Ruslan Salakhutdinov và Andriy Mnih cũng đã huấn luyện các mô hình PMF logistic, trong đó họ truyền tích vô hướng giữa các vector đặc trưng cụ thể cho người dùng và phim qua hàm logistic $\sigma(x) = 1/(1 + \exp(-x))$ để giới hạn phạm vi của các dự đoán:

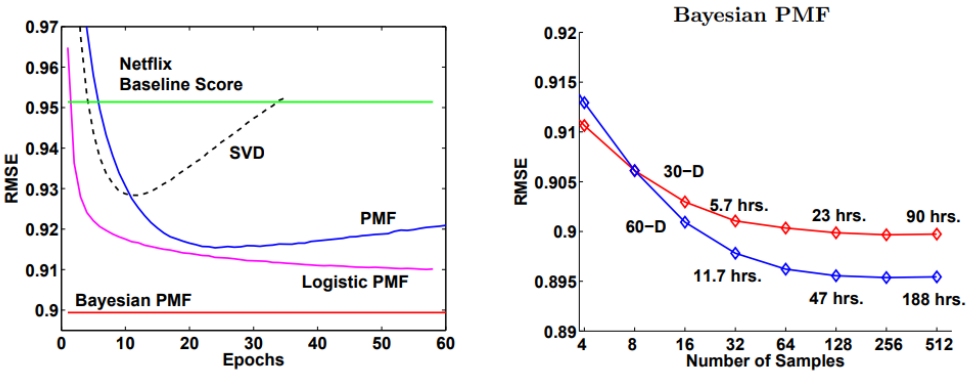
$$p(R \mid U, V, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\mathcal{N}\left(R_{ij} \mid U_i^T V_j, \sigma^2\right)\right]^{I_{ij}} \quad (2.2.14)$$

Trong đó, các xếp hạng từ 1 đến 5 được ánh xạ vào khoảng [0, 1] bằng cách sử dụng hàm $t(x) = (x-1)/4$ và để phạm vi của các giá trị xếp hạng hợp lệ khớp với phạm vi của các dự đoán mô hình của Ruslan Salakhutdinov và Andriy Mnih có thể tạo ra. Các mô hình PMF logistic có thể đôi khi cung cấp kết quả tốt hơn một chút so với các mô hình tuyến tính cùng loại. Để tăng tốc độ huấn luyện, thay vì thực hiện toàn bộ học theo lô, Ruslan Salakhutdinov và Andriy Mnih đã chia dữ liệu Netflix thành các lô nhỏ có kích thước 100.000 (bộ ba người dùng/phim/xếp hạng) và cập nhật các vector đặc trưng sau mỗi lô nhỏ. Họ sử dụng tốc độ học là 0.005 và động lượng là 0.9 để huấn luyện các mô hình PMF tuyến tính cũng như logistic.

2.2.3.3. Mô tả quá trình huấn luyện mô hình BPMF

Trong quá trình thực hiện, Ruslan Salakhutdinov và Andriy Mnih khởi tạo bộ lấy mẫu Gibbs bằng cách đặt các tham số mô hình U và V bằng các ước lượng MAP thu được bằng cách huấn luyện một mô hình PMF tuyến tính. Họ cũng đặt $\mu_0 = 0$, $v_0 = D$, và trong đó W_0 là ma trận đơn vị cho cả siêu tiên nghiệm của người dùng và phim. Độ chính xác của nhiễu quan sát α được cố định là 2. Phân phối dự đoán được tính toán thông qua công thức số 8 bằng cách chạy bộ lấy mẫu Gibbs với các mẫu $\{U_i^{(k)}, V_j^{(k)}\}$ thu thập sau mỗi bước chạy giải thuật này.

2.2.3.4. Kết quả thực nghiệm thu được



Hình 2. 5. Kết quả thực nghiệm thu được bằng cách sử dụng phương pháp BPMF

Mô tả cho hình 2.5: Hình 2. Bảng bên trái: Hiệu suất của SVD, PMF, logistic PMF và Bayesian PMF sử dụng vector đặc trưng 30D, trên dữ liệu kiểm tra của Netflix. Trục tung hiển thị RMSE (sai số bình phương trung bình), và trục hoành cho biết số lần lặp, hay số lần duyệt qua toàn bộ tập huấn luyện; Bảng bên phải: RMSE cho các mô hình Bayesian PMF trên tập dữ liệu kiểm tra theo số lượng mẫu được tạo ra. Hai đường cong là cho các mô hình có vector đặc trưng 30D và 60D.

Trong một số thí nghiệm đầu tiên, Ruslan Salakhutdinov và Andriy Mnih [23] so sánh một mô hình BPMF với một mô hình SVD, một mô hình PMF tuyến tính và một mô hình PMF logistic, tất cả sử dụng vector đặc trưng 30D. Mô hình SVD được huấn luyện để cực tiểu hóa tổng bình phương khoảng cách đến các mục quan sát của ma trận mục tiêu, không áp dụng điều chuẩn cho các vector đặc trưng. Lưu ý rằng mô hình này có thể được coi là một mô hình PMF được huấn luyện bằng phương pháp khả năng tối đa (Maximum Likelihood Method). Đối với các mô hình PMF, các tham số điều chuẩn λ_U và λ_V được đặt là 0.002. Hiệu suất dự đoán của các mô hình này trên tập kiểm định được hiển thị trong Hình 2.5 (bảng bên trái). Giá trị trung bình của phân phối dự đoán của mô hình BPMF đạt RMSE là 0.8994, so với RMSE là 0.9174 của một mô hình PMF tuyến tính có điều chuẩn vừa phải, cải thiện hơn 1.7% [23].

Mô hình PMF logistic có hiệu suất cao hơn mô hình tuyến tính, đạt RMSE là 0.90971. Tuy nhiên, hiệu suất của nó vẫn kém hơn nhiều so với mô hình BPMF.

Một mô hình SVD đơn giản đạt RMSE khoảng 0.9280 và sau khoảng 10 epoch thì bắt đầu xuất hiện hiện tượng overfit nặng. Thí nghiệm này minh chứng rõ ràng rằng các mô hình PMF được huấn luyện bằng SVD và MAP có thể xảy ra hiện tượng overfit và độ chính xác dự đoán có thể được cải thiện bằng cách tích phân các tham số và siêu tham số của mô hình [23].

D	Valid. RMSE		%	Test RMSE		%
	PMF	BPMF	Inc.	PMF	BPMF	Inc.
30	0.9154	0.8994	1.74	0.9188	0.9029	1.73
40	0.9135	0.8968	1.83	0.9170	0.9002	1.83
60	0.9150	0.8954	2.14	0.9185	0.8989	2.13
150	0.9178	0.8931	2.69	0.9211	0.8965	2.67
300	0.9231	0.8920	3.37	0.9265	0.8954	3.36

Hình 2. 6. Bảng dữ liệu thực nghiệm so sánh PMF và BPMF

Ruslan Salakhutdinov và Andriy Mnih [23] sau đó đã huấn luyện các mô hình PMF lớn hơn với D = 40 và D = 60. Việc kiểm soát dung lượng cho những mô hình này trở thành một nhiệm vụ khá thách thức. Ví dụ, một mô hình PMF với D = 60 có khoảng 30 triệu tham số. Do dó, việc tìm kiếm các giá trị điều chuẩn phù hợp trở nên rất tốn kém về tính toán. Bảng 2.4 cũng cho thấy rằng đối với các vector đặc trưng 60 chiều, BPMF vượt trội so với PMF MAP hơn 2%. Ruslan Salakhutdinov và Andriy Mnih đã chỉ ra rằng ngay cả phần mở rộng Bayesian đơn giản nhất cũng có thể có đặc trưng của mô hình PMF, trong đó các ưu tiên Gamma được đặt trên các siêu tham số chính xác là α_U và α_V (xem Hình 2.4, mô hình bên trái), do dó nó cũng có hiệu suất cao hơn nhiều so với các mô hình PMF được huấn luyện bằng MAP, mặc dù nó không hoạt động tốt bằng các mô hình BPMF [23].

Ruslan Salakhutdinov và Andriy Mnih [23] đã nhận định thông qua kết quả thực nghiệm rằng khi kích thước chiều của đặc trưng tăng lên, độ chính xác của mô hình PMF được huấn luyện bằng MAP không cải thiện và việc kiểm soát overfit trở thành một vấn đề quan trọng. Tuy nhiên, độ chính xác dự đoán của mô hình BPMF được cải thiện đều đặn khi độ phức tạp của mô hình tăng lên. Do đó, họ đã thử nghiệm với các mô hình BPMF có D = 150 và D = 300 vector đặc trưng và kết quả RMSE cho hai mô hình này trên tập kiểm định là 0.8931 và 0.8920 (kết quả từ Hình

2.6) và điều đó cho thấy những mô hình này không chỉ vượt trội hơn các mô hình PMF MAP tương ứng mà còn vượt trội hơn các mô hình BPMF có ít tham số hơn. Những kết quả này rõ ràng cho thấy rằng phương pháp Bayes không yêu cầu giới hạn độ phức tạp của mô hình dựa trên số lượng mẫu huấn luyện. Tuy nhiên trong thực tế, chúng ta thường sẽ bị giới hạn bởi tài nguyên máy tính hiện tại [23].

2.3. ALS

2.3.1. Tổng quan và mô hình bài toán

Alternating Least Squares (ALS) là một phương pháp lọc cộng tác, sử dụng kỹ thuật phân tích ma trận để giải quyết vấn đề quá khớp (overfitting) trong dữ liệu thưa thớt và tăng độ chính xác của dự đoán trong hệ thống gợi ý.

Trong một số hệ thống đánh giá hiện tại, dữ liệu đánh giá có thể được mô tả và biểu diễn dưới dạng một ma trận R gồm m × n phần tử, trong đó n là số người dùng và m là số sản phẩm. Phần tử $(u, i)^{th}$ trong ma trận R có nghĩa là λ_{ui}, nó có nghĩa là đánh giá của người dùng u cho sản phẩm i. Do đó, Ma trận R là ma trận thưa, vì vậy các sản phẩm không nhận được nhiều đánh giá từ người dùng. Cho nên, ma trận R có nhiều giá trị bị thiếu nhất. Phân tích ma trận là giải pháp cho vấn đề ma trận thưa này. Có hai vector k chiều được gọi là "nhân tố" trong đó định nghĩa về chúng là [24]:

- x_u là ma trận k chiều tượng trưng cho mọi người dùng u
- y_i là ma trận k chiều tượng trưng cho các sản phẩm i

Một phương trình đã được chứng minh như sau:

$$\text{argmin} \sum_{r_u i} (r_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) \qquad (2.3.1)$$

Trong đó, λ được gọi là hệ số điều chuẩn và nó được sử dụng để giải quyết vấn đề quá khớp (overfitting) và còn được gọi là điều chuẩn trọng số λ. Giá trị của λ

có thể được điều chỉnh để giải quyết vấn đề quá khớp, giá trị mặc định thường là 1 [24].

Giả sử rằng, tập hợp các biến x_u là một hằng số thì hàm mục tiêu của y_i là hàm lồi và tập hợp các biến y_i là hằng số thì hàm mục tiêu của x_u là hàm lồi. Do đó, giá trị tối ưu của x_u và y_i có thể được tìm thấy bằng cách lặp lại phương pháp đã nêu trên cho đến khi hội tụ, nói cho dễ hiểu thì ALS hoạt động bằng cách thay phiên cố định một ma trận nhân tố và giải cho ma trận nhân tố còn lại, cho đến khi hội tụ. Đây còn được gọi là phương pháp lặp xen kẽ bình phương nhỏ nhất (Alternating Least Squares - ALS) [24]

ALS có thể xử lý dữ liệu phản hồi rõ ràng (explicit feedback), nơi người dùng cung cấp đánh giá cho các sản phẩm hoặc dữ liệu phản hồi không rõ ràng (implicit feedback), nơi người dùng chỉ thể hiện sự quan tâm bằng cách xem, mua, hoặc tương tác với các sản phẩm, trong đồ án chuyên ngành này thì sản phẩm mà ta để cập đến là phim.

Mã giả của ALS được mô tả như sau [24]:

Thuật toán: Alternating Least Squares (ALS)
❖ Hàm thủ tục ALS(x_u, y_i) 1. Khởi tạo $x_u \Leftarrow 0$ 2. Khởi tạo ma trận y_i với giá trị ngẫu nhiên 3. Lặp lại cho đến số lần lặp tối đa • Tìm kiếm y_i và giải quyết x_u bằng cách giảm thiểu hàm mục tiêu • Hàm (tổng các sai số bình phương) • Tiếp tục tìm kiếm y_i và giải quyết x_u bằng cách giảm thiểu hàm mục tiêu • Lặp lại hoạt động tương tự 4. Trả về giá trị x_u, y_i 5. Kết thúc

2.3.2. Thực nghiệm và đánh giá

Tập dữ liệu thực nghiệm trong đồ án chuyên ngành được tham khảo từ tập thử nghiệm trong đề tài [24] của nhóm tác giả gồm Subasish Gosh, Nazmun Nahar, Mohammad Abdul Wahab, Munmun Biswas, Mohammad Shahadat Hossain và Karl Andersson. Dữ liệu thực nghiệm đến từ bộ dữ liệu của MovieLens gồm 1 triệu dataset gồm có 71567 người dùng xếp hạng và 10681 bộ phim, trong đó những người dùng trong tập dữ liệu này đã thực hiện xếp hạng cho tối thiểu 20 bộ phim khác nhau. Và có khoảng 10000054 xếp hạng có sẵn trong tập dữ liệu này. Do đó, tập dữ liệu MovieLens được cho là tập dữ liệu có phản hồi rõ ràng từ người dùng. Nhóm tác giả chia bộ dữ liệu thành các tập huấn luyện và tập kiểm tra, sử dụng thuật toán ALS để huấn luyện mô hình gợi ý và dự đoán các đánh giá còn thiếu trong tập kiểm tra. Tác giả sử dụng chỉ số RMSE để đo lường sai số giữa các đánh giá thực tế và dự đoán. Hệ thống thực nghiệm của nhóm tác giả được triển khai trên Apache Spark và cụm Apache Hadoop [24].

Parameters	Value
Lambda	0.01
Iteration	5

Hình 2. 7. Bảng cài đặt tham số tham khảo cho thực nghiệm

Trong quá trình xây dựng hệ thống nhóm tác giả [24] đã chia các tập dữ liệu thành 20 RDD (Viết tắt của Resilient Distributed Dataset – hay còn gọi là Tập dữ liệu phân tán linh hoạt) cho các yếu tố người dùng và các yếu tố sản phẩm, mỗi loại 10 RDD. Do đó, 2 yếu tố này được tổng hợp để tạo ra các gợi ý sản phẩm Tham số ban đầu cho. Tham số cho hệ thống được hiển thị trên hình 2.7 [24].

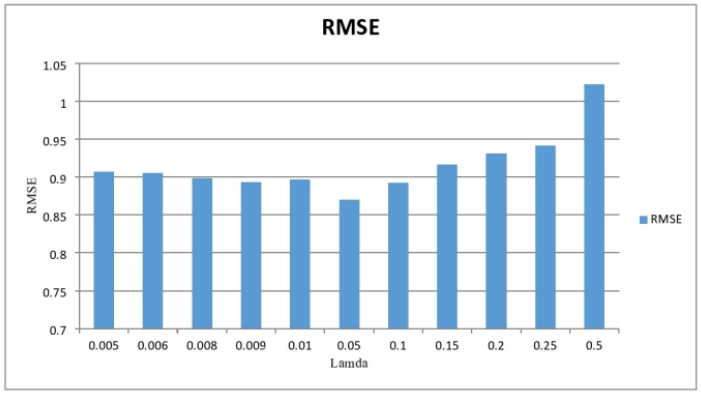
UserID	ItemID	Avg Rating	Prediction
5192	557	5	5.402
5192	864	4	5.284
5192	2512	3.9	5.189
5192	1851	4	5.125
5192	2905	4.6	5.098

Hình 2. 8. Tập khuyến nghị sử dụng thuật toán ALS

Từ hình 2.8 cho thấy 5 gợi ý hàng đầu của sản phẩm cho một người dùng cụ thể cùng với điểm trung bình (Avg Rating) và giá trị dự đoán (Prediction) trong tập dữ liệu và giá trị thực nghiệm RMSE thu được là 0.9 đối với bảng trên [24].

Lamda	Iteration	RMSE
0.005	5	0.907
0.006	5	0.905
0.008	5	0.898
0.009	5	0.893
0.01	5	0.896
0.05	5	0.870
0.1	5	0.892
0.15	5	0.916
0.2	5	0.931
0.25	5	0.941
0.5	5	1.022

Hình 2. 9. Giá trị RMSE tương ứng với số lần lặp trong thực nghiệm

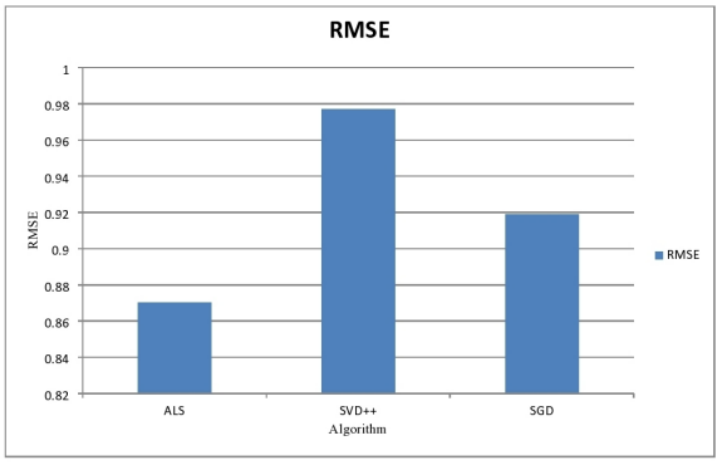


Hình 2. 10. Mô hình biểu diễn giá trị RMSE

Trong hình 2.9, quá trình huấn luyện của nhóm tác giả [24] bao gồm một bước kiểm tra ngưỡng RMSE trong đó giá trị Lamda(λ) được thay đổi để tối ưu hóa mô hình. Trong hình 2.10, bằng cách giữ nguyên số vòng lặp là 5 và thay đổi giá trị lambda trong khoảng từ 0.005 đến 1 do đó giá trị RMSE nhỏ nhất được tìm thấy là 0.870 đối với giá trị lambda là 0.05. Kết quả thực nghiệm từ hình 2.9 đã chỉ ra rằng khi giá trị lambda nằm trong khoảng từ 0.005 đến 0.006, giá trị RMSE lần lượt là 0.907 và 0.905. Và khi giá trị lambda tăng lên thì giá trị RMSE sẽ giảm so với giá trị trước đó. Sau đó, giá trị RMSE sẽ thay đổi khi giá trị lambda tăng lên và giá trị RMSE nhỏ nhất được tìm thấy khi giá trị lambda là 0.05. Giá trị RMSE nhỏ nhất là 0.870 khi số vòng lặp là 5 [24].

Algorithm	RMSE
ALS	0.870
SVD++	0.977
SGD	0.919

Hình 2. 11. So sánh giá trị RMSE giữa các phương pháp ALS, SVD++ và SGD



Hình 2. 12. Mô hình biểu diễn so sánh giá trị RMSE giữa các phương pháp ALS, SVD++ và SGD

Do đó, trong kết quả nghiên cứu của nhóm tác giả [24], họ đã đưa ra một phân tích so sánh được tiến hành để đánh giá hiệu suất của phương pháp ALS. Do đó, một sự so sánh được tiến hành giữa ALS, SVD++ và các phương pháp phân tích ma trận song song SGD. Kết quả cho thấy rằng giá trị RMSE được giảm đáng kể bằng cách sử dụng phương pháp phân tích ma trận ALS cho kết quả giá trị RMSE là 0.870. Do đó, thí nghiệm cho thấy rằng thuật toán ALS phù hợp để huấn luyện tập dữ liệu phản hồi rõ ràng nơi người dùng cung cấp đánh giá cho các sản phẩm [24].

2.4. Kết chương 2

Chương này trong đồ án chuyên ngành đã trình bày chi tiết thông tin, bài toán, công thức và một số thực nghiêm trong thực tế có liên quan đến bài toán khuyến nghị phim của các phương pháp như PMF, BPMF và ALS và chứng minh được độ hiệu quả của các phương pháp vừa kể trên dành cho bài toán khuyến nghị

phim mà đồ án chuyên ngành đã đặt ra khi so sánh với các phương pháp tiếp cận truyền thống khác.

Với các dữ liệu và thông tin đã thu thập được, đồ án chuyên ngành này sẽ tiếp tục khai thác sâu vào tính ứng dụng của bài toán khuyến nghị phim trên các xem phim. Chương tiếp theo sẽ trình bày rõ hơn về cách thức hoạt động, cũng như ứng dụng khuyến nghị trong thực tế của một số trang web xem phim, để rồi từ đó đưa ra nhận định và so sánh một cách khách quan nhất đối với những trang web xem phim đã được chọn trên.

TÀI LIỆU THAM KHẢO

Tiếng việt

[1] C. X. Kiều, "Nghiên cứu và xây dựng Hệ thống Khuyến nghị cho bài toán dịch vụ giá trị gia tăng trong ngành viễn thông," Đại học Công Nghệ, Hà Nội, 2017.

[4] T. N. Huỳnh, "Phát triển một số phương pháp khuyến nghị hỗ trợ tìm kiếm thông tin học thuật dựa trên tiếp cận phân tích mạng xã hội," Trường Đại học Công nghệ thông tin, Hồ Chí Minh, 2016.

[11] M. V. Bùi, "Nghiên cứu, Xây dựng Hệ thống Khuyến nghị phim tự động," Học viện Bưu chính viễn thông, Hà Nội, 2017.

[12] L. T. Đỗ, "Phát triển một số phương pháp xây dựng hệ tư vấn," Học viện Bưu chính viễn thông, 2020, 2020.

Tiếng anh

[2] D. Jannach, M. Zanker, A. Felfernig and G. Friedrich, Recommender System: An Introduction, New York: Cambridge University Press, 2010.

[3] A. Gediminas and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering, 2005.

[5] M. de Gemmis, P. Lops, G. Semeraro and P. Basile, Integrating tags in a semantic content-based recommender, New York: Proceedings of the 2008 ACM conference on Recommender systems, 2008.

[6] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, Boston – USA: Addison Wesley Longman Publishing Co. Inc., 1999.

[7] T. Joachims , Text categorization with Support Vector Machines: Learning with many relevant features, London, UK: In procedings of the 10th Europen Conference on Machine Learning, 1998.

[8] K. Nigam, A. K. Mccallum, S. Thrun and T. Mitchell, Text Classification from Labeled and Unlabeled Documents using EM, Machine Learning, 2000.

[9] M. Pazzani and D. Billsus, Learning and Revising User Profiles: The Identification of Interesting Web Sites, Machine Learning, 1997.

[10] R. J. Mooney and L. Roy, Content-based book recommending using learning for text categorization, New York: In proceedings of the Fifth ACM Conference on Digital Libraries, 2000.

[13] J. S. Breese, D. Heckerman and C. Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Fan Francisco: In Proceeding of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998.

[14] X. Su and T. M. Khoshgoftaar, A survey of collaborative filtering technique, in Artif. Intell, 2009 .

[15] J. Bobadilla, F. Ortega, A. Hernando and A. GutiéRrez, Recommender systems survey, Knowledge-Based Systems, 2013.

[16] R. Burke, Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-adapted Interaction, 2002.

[17] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov and D. Netes, Combining content-based and collaborative filters in an online newspaper, Massachusetts: Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999.

[18] G. Adomavicius and A. Tuzhilin , Context-Aware Recommender Systems, New York: New York University, 2011.

[19] B. Smyth and P. Cotter, A personalized television listings service, Communications of the ACM, 2000.

[20] C. Basu, H. Hirsh and W. Cohen, Recommendation as Classification: Using Social and Content-Based Information in Recommendation, AAAI Press, 2000.

[21] M. J. Pazzani , A Framework for Collaborative, Content-Based and Demographic Filtering, Artificial Intelligence Review, 1999.

[22] R. Salakhutdinov and . A. Mnih, Probabilistic Matrix Factorization, Canada: Department of Computer Science, University of Toronto, 2007.

[23] R. Salakhutdinov and A. Mnih, Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo, Canada: Proceedings of the 25th international conference on Machine learning, 2008.

[24] S. Gosh, N. Nahar, M. A. Wahab, M. Biswas, M. S. Hossain and K. Andersson, Recommendation System for E-commerce using Alternating Least Squares (ALS) on Apache Spark, Intelligent Computing and Optimization, Proceedings of the 3rd International Conference on Intelligent Computing and Optimization 2020, 2021.