**ĐẠI HỌC HUẾ**

**TRƯỜNG ĐẠI HỌC KINH TẾ HUẾ**

**KHOA HỆ THỐNG THÔNG TIN KINH TẾ**

-----🙞🙜🕮🙞🙜-----

****

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG HỆ TƯ VẤN SỬ DỤNG PHƯƠNG PHÁP LỌC CỘNG TÁC**

**Người thực hiện**: Lê Nguyễn Hoài Huy

*Niên khóa 2016 - 2020*

**ĐẠI HỌC HUẾ**

**TRƯỜNG ĐẠI HỌC KINH TẾ HUẾ**

**KHOA HỆ THỐNG THÔNG TIN KINH TẾ**

-----🙞🙜🕮🙞🙜-----

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG HỆ TƯ VẤN SỬ DỤNG PHƯƠNG PHÁP LỌC CỘNG TÁC**

**Sinh viên thực hiện: Giảng viên hướng dẫn:**

**Họ và tên: Lê Nguyễn Hoài Huy T.S. Nguyễn Đình Hoa Cương**

**Lớp: K50A – Tin học kinh tế**

**MSV: 16K4081028**

**Niên khóa: 2016-2020**

*Huế, tháng 12 năm 2019*

# LỜI CẢM ƠN

Được sự phân công của khoa Hệ Thống Thông Tin Kinh Tế, trường Đại Học Kinh Tế Huế, nay em đã hoàn thành khóa luận tốt nghiệp cuối khóa.

Để hoàn thành nhiệm vụ được giao, ngoài sự nỗ lực học hỏi của bản thân còn có sự hướng dẫn tận tình của thầy cô, đặc biệt hơn là sự động viên tinh thần lớn từ gia đình.

Em xin chân thành cảm ơn TS. Nguyễn Đình Hoa Cương, người đã hướng dẫn cho em trong suốt thời gian thực tập. Thầy đã chỉ ra những vấn đề mà em gặp phải lúc làm khóa luận, định hướng đi cho em để em hoàn thành tốt nhiệm vụ.

Xin cảm ơn tất cả các bạn bè, thầy cô, các anh chị trong công ty SoftWorld và gia đình đã giúp đỡ, dìu dắt em trong suốt thời gian qua. Tất cả mọi người đều đã rất nhiệt tình giúp đỡ. Bên cạnh đó, động lực vô cùng to lớn để em cố gắng hoàn thành tốt khóa luận này đó là gia đình.

Để hoàn thành khóa luận này, em đã nhận được sự hướng dẫn, giúp đỡ và động viên tận tình từ nhiều phía. Tất cả những điều này đã trở thành một động lực rất lớn giúp em có thể hoàn thành tốt mọi công việc được giao.

Một lần nữa em xin chân thành cảm ơn!

*Huế, tháng 12 năm 2019*

*Sinh viên ký tên*

*Lê Nguyễn Hoài Huy*

# MỤC LỤC

[**LỜI CẢM ƠN i**](#_Toc29412703)

[**MỤC LỤC ii**](#_Toc29412704)

[**DANH MỤC HÌNH VẼ iv**](#_Toc29412705)

[**DANH MỤC BẢNG BIỂU v**](#_Toc29412706)

[**DANH MỤC THUẬT NGỮ vi**](#_Toc29412707)

[**MỞ ĐẦU 1**](#_Toc29412708)

[1. Lý do chọn đề tài 1](#_Toc29412709)

[2. Mục tiêu của đề tài 2](#_Toc29412710)

[3. Đối tượng và phạm vi nghiên cứu 2](#_Toc29412711)

[4. Phương pháp nghiên cứu 2](#_Toc29412712)

[5. Cấu trúc khóa luận 2](#_Toc29412713)

[**NỘI DUNG 4**](#_Toc29412714)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 4**](#_Toc29412715)

[1.1. Giới thiệu về hệ tư vấn 4](#_Toc29412716)

[1.2. Cơ sở toán học 5](#_Toc29412717)

[1.2.1. Ma trận tiện ích 5](#_Toc29412718)

[1.2.2. Phương pháp chuẩn hóa giá trị trong ma trận tiện ích 8](#_Toc29412719)

[1.2.3. Vectơ và khoảng cách giữa 2 vectơ 8](#_Toc29412720)

[1.3. Các phương pháp tư vấn cơ bản 10](#_Toc29412721)

[1.3.1. Lọc cộng tác 10](#_Toc29412722)

[1.3.2. Tư vấn dựa vào nội dung 13](#_Toc29412723)

[1.3.3. Phương pháp hỗn hợp (lai ghép) 15](#_Toc29412724)

[1.4. Vấn đề “khởi động lạnh” (Cold-start) 16](#_Toc29412725)

[1.5. Phát biểu bài toán nghiên cứu 18](#_Toc29412726)

[1.6. Các công trình liên quan 18](#_Toc29412727)

[**CHƯƠNG 2: XÂY DỰNG HỆ TƯ VẤN THEO PHƯƠNG PHÁP LỌC CỘNG TÁC 34**](#_Toc29412728)

[2.1. Dữ liệu và tiền xử lý dữ liệu 34](#_Toc29412729)

[2.1.1. Chuẩn bị dữ liệu 34](#_Toc29412730)

[2.1.2.Tiền xử lý dữ liệu 35](#_Toc29412731)

[2.2. Thuật toán lọc cộng tác dựa trên người dùng 36](#_Toc29412732)

[2.3. Phương pháp đánh giá 39](#_Toc29412733)

[2.3.1. Phép đo Precision – Recall 39](#_Toc29412734)

[2.3.2. Phép đo F1 41](#_Toc29412735)

[2.3.3. Phép đo MSE 42](#_Toc29412736)

[2.3.4. Phép đo RMSE 42](#_Toc29412737)

[**CHƯƠNG 3: THÍ NGHIỆM VÀ THẢO LUẬN 43**](#_Toc29412738)

[3.1. Thiết lập thí nghiệm 43](#_Toc29412739)

[3.2. Kết quả và thảo luận 44](#_Toc29412740)

[3.2.1. Bảng phân tích, thống kê kết quả thí nghiệm 44](#_Toc29412741)

[3.2.2. Biểu đồ 44](#_Toc29412742)

[3.2.3. Thảo luận 45](#_Toc29412743)

[**KẾT LUẬN 45**](#_Toc29412744)

[**DANH MỤC TÀI LIỆU THAM KHẢO 47**](#_Toc29412745)

# DANH MỤC HÌNH VẼ

[Hình 1.1: Hình ảnh đơn giản về cách hệ thống đề xuất 5](#_Toc29880652)

[Hình 1.2: Hình ảnh biểu diễn vectơ với hướng và độ lớn (Nguồn Internet) 9](file:///E:\documents\Documents\KLTN2019\16K408012855555.docx#_Toc29880653)

[Hình 1.3: UBCF (trái) và IBCF (phải) 10](#_Toc29880654)

[Hình 1.4: Hình ảnh đơn giản về hệ thống lọc nội dung 13](#_Toc29880655)

[Hình 1.5: Hình ảnh đơn giản sử dụng phương pháp tư vấn hỗn hợp 15](#_Toc29880656)

Hình 1.6: Hình ảnh đơn giản sử dụng phương pháp tư vấn hỗn hợp (vẽ lại)……… 17

[Hình 2.1 : Sơ đồ thuật toán UB-CF 36](file:///E:\documents\Documents\KLTN2019\16K408012855555.docx#_Toc29880657)

[Hình 3.1: Giao diện ban đầu của phần mềm 43](#_Toc29880658)

[Hình 3.2: Giao diện phần mềm dự đoán các bộ phim người dùng có thể thích 44](#_Toc29880659)

[Hình 3.3: Giao diện trang chủ phần mềm để đánh giá mô hình 45](#_Toc29880660)

[Hình 3.4: Giao diện phần mềm hiển thị danh sách các phim đã được đánh giá 46](#_Toc29880661)

[Hình 3.5: Biểu đồ đường của các tập dữ liệu trên phương pháp MSE 4](#_Toc29880661)7

[Hình 3.6: Biểu đồ đường của các tập dữ liệu trên phương pháp RMSE 4](#_Toc29880661)8

# DANH MỤC BẢNG BIỂU

[**Bảng 1: Ví dụ về UM với hệ thống Gợi ý bài hát** 6](#_Toc29967754)

[**Bảng 2: Bảng ma trận đánh giá** 20](#_Toc29967755)

[**Bảng 3: Bảng ví dụ đánh giá giữa các phương pháp** 43](#_Toc29967756)

[**Bảng 4: Kết quả chạy mô hình với bộ dữ liệu ml-100k** 49](#_Toc29967757)

# DANH MỤC THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| **Viết tắt** | **Tên tiếng Anh** | **Tên Tiếng Việt** |
| CBF | Content Based Filtering | Lọc nội dung |
| CF | Collaborative Filtering | Lọc cộng tác |
| fv | *feature* *vector* | Vectơ đặc trưng |
| IBCF | Item - Base Collaborative Filtering | Lọc cộng tác dựa trên vật phẩm |
| IP | *Item Profiles* | Hồ sơ vật phẩm |
| Items/item | *Items* | Vật phẩm/sản phẩm hoặc hàng hóa |
| KNN | K-nearest neighbor | K – hàng xóm lân cận |
| ML | *Machine Learning* | Học máy |
| MSE | *Mean squared error* | Sai số toàn phương trung bình |
| NBCF | Neighborhood - Based Collaborative Filtering | Lọc cộng tác làng giềng gần |
| Rating/ Preference | *Rating* | Đánh giá |
| RMSE | *Root mean square error* | Độ lệch trung bình bình phương |
| RP | *Rating Predirect* | Dự đoán đánh giá |
| RS | Recommendation systems | Hệ thống tư vấn |
| UBCF | User - Base Collaborative Filtering | Lọc cộng tác dựa trên người dùng |
| UM | *Utility Matrix* | Ma trận tiện ích |
| UP | *User Profiles* | Hồ sơ người dùng |
| Users/user | *Users* | Người dùng |

# MỞ ĐẦU

# 1. Lý do chọn đề tài

Hệ tư vấn (*Recommendation systems - RS*) [1] là một trong những lĩnh vực nghiên cứu rất quan trọng, được quan tâm từ khi bài báo đầu tiên được xuất bản [2] về phương pháp lọc cộng tác (Collaborative Filtering - CF) [3] vào giữa những năm 1990.

Ngày nay, khoa học kỹ thuật trên thế giới nói chung và ở Việt Nam nói riêng đang trên đà phát triển ngày càng mạnh mẽ và không ngừng nâng cao phát triển về mọi mặt. Đặc biệt là ngành công nghệ thông tin. Để có thể đáp ứng nhu cầu ngày càng cao của con người thì hàng loạt các công ty về máy tính được ra đời và các ứng dụng công nghệ thông tin đang được áp dụng ngày một nhiều hơn. Điều này cũng đem cho chúng ta đến một thời đại mà ở đó một lượng lớn thông tin được tạo ra hàng ngày. Hàng ngàn bản tin được viết và đưa lên *Internet*, hàng trăm bài hát được thu âm và nhiều đầu sách được số hóa mỗi ngày. Dẫn đến một vấn đề là làm thế nào để chúng ta có thể tìm được đúng những nội dung phù hợp với từng cá nhân.

Hầu hết mọi người có thể lên *Google* hoặc *Facebook* để tìm kiếm, tham khảo từ bạn bè. Việc đó có thể là cũng không hữu ích và tốn thời gian khi ta không biết cụm “từ khóa” ta cần tìm, hay là chúng ta có ít bạn bè có sự quan tâm tương tự. Không chỉ vậy việc đưa ra các khuyến nghị cho người dùng những sản phẩm, nội dung mà họ có thể quan tâm sẽ cải thiện hiệu quả kinh doanh của doanh nghiệp, cũng như sự trải nghiệm.

Trên cơ sở đó, việc áp dụng *RS* là đang rất cần được sự quan tâm, bởi việc ứng dụng của nó trong quá trình xử lý thông tin. Một vài ứng dụng nổi tiếng đã được xây dựng dựa trên các *RS* rất nổi tiếng như: Hệ tư vấn sách, CDs của *Amazon.com*, hệ tư vấn phim của *Netflix* [4]. Không chỉ có *Amazon*, *Netflix* mà còn có rất nhiều hệ thống khác trên thế giới đang sử dụng *RS*-*CF* để gợi ý cho khách hàng tới những sản phẩm có thể thích hợp với người dùng.

Có thể nói việc áp dụng *RS* vào các ngành thương mại điện tử ngày nay là điều rất cần thiết. Các hệ thống tư vấn giúp người dùng nhận các đề xuất được cá nhân hóa và giúp người dùng đưa ra được quyết định chính xác trong giao dịch trực tuyến của họ, từ đó tăng doanh số và nâng cao trải nghiệm trình duyệt của người dùng, giữ khách hàng ở lại hệ thống. Ngoài ra, các vấn đề về quá tải thông tin được giải quyết bởi công cụ tìm kiếm, nhưng chúng không cung cấp giải pháp cá nhân hóa dữ liệu. RS Rất nhiều công việc đã được thực hiện về chủ đề này, đây vẫn là một chủ đề rất được yêu thích trong các nhà khoa học dữ liệu.

# 2. Mục tiêu của đề tài

Trong khóa luận này, mục tiêu tổng quát của đề tài là nghiên cứu phương pháp xây dựng mô hình hệ tư vấn dựa trên phương pháp lọc cộng tác, từ đó tạo ra được một hệ thống tư vấn phù hợp nhất với nhu cầu của người dùng.

* Mục tiêu cụ thể:
  + Đánh giá thuật toán phương pháp lọc cộng tác;
  + Nghiên cứu các bộ cơ sơ dữ liệu trong tập dữ liệu Movilens;
  + Xem xét các vấn đề của hệ tư vấn sử dụng phương pháp lọc cộng tác đối với các phương pháp khác;
  + Tiến hành nghiên cứu các công cụ hỗ trợ xây dựng và triển khai website để tiến hành thử nghiệm thuật toán;
  + Nghiên cứu những hệ thống sử dụng hệ tư vấn để nắm được tính ứng dụng của hệ tư vấn hiện nay.
* Mục tiêu tổng quát: Để làm được điều đó, trước hết chúng ta cần xây dựng được một mô hình phục vụ cho việc dự đoán sở thích của người dùng và mối tương đồng giữa các người dùng với nhau, mô hình này xây dựng từ bộ cơ sở dữ liệu MovieLens. Tác giả tiến hành thực hiện đề tài: “Xây dựng hệ tư vấn phim movielens bằng phương pháp lọc cộng tác”.

# 3. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứ u: Phương pháp lọc cộng tác dựa trên người dùng dựa trên bộ cơ sở dữ liệu MovieLens

- Phạm vi nghiên cứu: Các công trình, bài báo và bài viết về lĩnh vực hệ tư vấn.

# 4. Phương pháp nghiên cứu

-Phương pháp thực nghiệm: Tiến hành thực nghiệm các kết quả tính toán với mẫu dữ liệu nhỏ để có thể nắm được được toán cũng như hiểu được bản chất của phương pháp lọc cộng tác.

-Phương pháp nghiên cứu tài liệu: Phương pháp này nghiên cứu các tài liệu liên quan về hệ tư vấn, đồng thời tìm kiếm các hệ thống sử dụng phương pháp lọc cộng tác hay các phương pháp khác trong hệ tư vấn.

# 5. Cấu trúc khóa luận

Ngoài phần mở đầu và kết luận, khóa luận gồm 3 chương:

Chương I: Cơ sở lý thuyết

Chương này sẽ trình bày khái quát về hệ tư vấn, phát biểu bài toán cũng như giới thiệu các cơ sở toán và các phương pháp thường được sử dụng trong hệ tư vấn.

Ngoài ra, ở chương này sẽ nêu ra những nghiên cứu các công trình, bài báo, luận văn liên quan đến hệ tư vấn.

Chương II: Xây dựng hệ tư vấn với phương pháp lọc cộng tác

Chương này sẽ trình bày quá trình xây dựng mô hình, giới thiệu các tệp dữ liệu được sử dụng cũng như các phương pháp dùng để đánh giá mô hình.

Chương III: Thí nghiệm và kết luận

Chương này sẽ xây dựng các mô hình và đánh giá kết quả thí nghiệm. Và đưa ra những kết luận dựa trên những số liệu có được.

# NỘI DUNG

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

# 1.1. Giới thiệu về hệ tư vấn

*RS* là một hệ thống gợi ý được tạo ra dựa trên cơ sở lý thuyết luật số lớn [5] và tâm lý học đám đông [6]. Bởi con người có xu hướng đưa ra các lựa chọn dựa trên ý kiến hay lời khuyên của những người xung quanh và khuynh hướng có những sở thích giống nhau. Nhưng trong kỷ nguyên thông tin, hàng triệu thông tin được đưa lên mạng mỗi ngày, điều này dẫn tới yêu cầu phải có các phương pháp tự động thu thập thông tin và đưa ra lời khuyên để hỗ trợ cho các phương pháp truyền thống trên.

Đầu tiên bằng việc thu thập các dữ liệu của người dùng, thông qua các hoạt động như mua hàng, số lượng truy cập vào các đường dẫn mà từ đó chúng ta có thể dự đoán được mối tương quan giữa các người dùng với nhau. Hay nói cách khác, bằng việc khám phá các mẫu trong bộ dữ liệu rồi từ đó đưa ra các dự đoán, gợi ý cho người dùng những sản phẩm, dịch vụ mà người dùng được cho là có thể sẽ thích, qua đó nâng cao được chất lượng dịch vụ và thu lại lợi nhuận.

Trên thế giới, việc sử dụng *RS* vào các công việc kinh doanh đang được sử dụng rất nhiều ví dụ như *Netflix*, *Google* *News*, *Amazon*. Là các ông lớn đã áp dụng kỹ thuật này vào hệ thống. Cụ thể hơn, theo thống kê hai phần ba (2/3) số phim trên hệ thống *Netflix* được xem, 38% lượng nhấp chuột trên hệ thống tin tức Google News và 35% sản phẩm bán được trên hệ thống thương mại điện tử *Amazon* đều được nhờ vào *RS*. Gần đây nhất, thuật toán khuyến nghị cũng được áp dụng vào cho các ứng dụng phát nhạc như *Spotify* và *Deezer* để giới thiệu các bản nhạc bạn có thể thích thông qua lịch sử nghe nhạc của người dùng, do đó có thể nói *RS* là một hệ thống được áp dụng rất nhiều trong thương mại điện tử và công nghệ thông tin hiện nay.



*Hình 1.1: Hình ảnh đơn giản về cách hệ thống đề xuất*

([Nguồn http://triconinfotech](file:///E:\KLTN\Nguồn%20http:\triconinfotech) )

*RS* là một nhu cầu hết sức phổ biến và cần thiết cho hầu hết các hệ thống kinh doanh trực tuyến hiện nay. Đây là một hệ thống phức tạp và đòi hỏi các kỹ thuật xử lý thông minh dựa trên trí tuệ nhân tạo và các phương pháp học máy. Nhằm đặt vấn đề cho bài toán hệ tư vấn tự động cũng như đưa ra một số khái niệm, kỹ thuật cụ thể. Trong bài nghiên cứu này, sẽ đưa ra các mô tả về các hệ tư vấn tự động, các loại hệ tư vấn, cũng như khảo sát một số kỹ thuật học máy có thể áp dụng hiệu quả cho các hệ tư vấn tự động, đồng thời thảo luận về việc mở rộng hệ thống cỡ lớn trên nền tảng dữ liệu lớn (Big Data) [7].

# 1.2. Cơ sở toán học

Dưới đây là những khái niệm toán học được áp dụng để giải quyết bài toán dự đoán các đánh giá của người dùng với các vật phẩm chưa được đánh giá trước đó. Bài toán này sẽ được phát biểu rõ ràng hơn ở mục 1.6.

Qua đó chúng ta cần nằm những vấn đề lý thuyết cơ sở như:

* Ma trận tiện ích;
* Phương pháp chuẩn hóa ma trận trong ma trận tiện ích; và
* Vectơ và khoảng các giữa 2 vectơ.

Nắm được các vấn đề trên, chúng ta có thể hiểu được quy trình giải thuật *UBCF* hoạt động và việc tính độ tương tự giữa các người dùng trên cơ sở nào.

# 1.2.1. Ma trận tiện ích

Ma trận tiện ích (*Utility Matrix - UM*) gồm có mỗi cột đại diện cho một người dùng (gọi là - *users*), mỗi hàng đại diện cho một sản phẩm cụ thể (gọi là - *items*) và số ở giao điểm của một hàng và một cột biểu thị giá trị xếp hạng/đánh giá của *users* (được xem là mức độ quan tâm của user đến item - ta gọi giá trị này là *ratings* và thường được biểu diễn giá trị từ 0-5). Việc không có giá trị được đánh giá tại giao điểm giữa các hàng và cột cho thấy rằng người dùng chưa đánh giá vật phẩm.

Mức độ quan tâm, nếu đã là được biết trước thì sẽ được gán cho một giá trị ứng với mỗi cặp *user-item*. Qua đó, một tập hợp tất cả *ratings*, bao gồm cả những giá trị chưa biết cần được dự đoán. Tạo nên một ma trận gọi là *UM*.

Để hiểu rõ hơn, chúng ta cụ thể như sau ở bảng dưới. Hệ thống “Gợi ý bài hát”, các bài hát được *user* đánh giá theo mức độ từ 0 đến 5 sao. Các dấu '?' nền màu xám ứng với việc dữ liệu chưa tồn tại trong cơ sở dữ liệu. *RS* cần phải tự điền các giá trị này.

**Bảng 1: Ví dụ về UM với hệ thống Gợi ý bài hát**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** | **F** |
| Mưa nửa đêm | 5 | 5 | 0 | 0 | 1 | ? |
| Cỏ úa | 5 | ? | ? | 0 | ? | ? |
| Vùng lá me bay | ? | 4 | 1 | ? | ? | 1 |
| Con cò bé bé | 1 | 1 | 4 | 4 | 4 | ? |
| Em yêu trường em | 1 | 0 | 5 | ? | ? | ? |

Có tất cả 6 users A, B, C, D, E, F và 5 bài hát. Các ô màu xanh thể hiện việc một *user* đã đánh giá một bài hát với *ratings* từ 0 (không thích) đến 5 (rất thích). Các ô có dấu ‘?’ màu xám tương ứng với các ô chưa có dữ liệu. Công việc của một *RS* là dự đoán giá trị tại các ô màu xám này, từ đó đưa ra gợi ý cho người dùng. *RS*, đôi khi cũng được coi là bài toán *Matrix Completion* (Hoàn thiện ma trận).

Trong ví dụ đơn giản này, dễ thấy có 2 thể loại nhạc khác nhau: 3 bài đầu là nhạc Bolero [8] và 2 bài sau là nhạc thiếu nhi [9]. Từ dữ liệu này, ta cũng có thể đoán được rằng A, B thích thể loại Bolero; C, D, E, F thích thể loại thiếu nhi. Từ đó, một hệ thống tốt nên gợi ý Cỏ úa cho B; Vùng lá me bay cho A; Em yêu trường em cho D, E, F. Giả sử chỉ có hai thể loại nhạc này, khi có một bài hát mới, ta chỉ cần phân lớp nó vào thể loại nào, từ đó đưa ra gợi ý với từng *user*.

Thông thường, sẽ có rất nhiều *users* và *items* trong hệ thống, và mỗi *user* thường chỉ *đánh giá* một số lượng rất nhỏ các *item*, thậm chí có những *user* không *đánh giá item* nào (với *users* này thì cách tốt nhất là gợi ý *items* phổ biến nhất). Vì vậy, lượng ô màu xám (ô có dấu ‘?’) của *UM* trong các bài toán đó thường là rất lớn, và lượng các ô đã được điền là một số rất nhỏ.

Có thể thấy rằng càng nhiều ô được điền thì độ chính xác của hệ thống sẽ càng được cải thiện. Do đó, các hệ thống luôn luôn hỏi *users* về sự quan tâm của họ tới *items*, và hệ thống luôn muốn *user* đánh giá càng nhiều *item* càng tốt. Việc đánh giá các *item* không những giúp *users* khác biết được chất lượng sản phẩm mà còn giúp hệ thống biết được sở thích của *user*, qua đó có chính sách quảng cáo, tư vấn hợp lý.

Không có *UM*, gần như không thể gợi ý được những *items* tới *users*, ngoài cách luôn luôn gợi ý các sản phẩm phổ biến nhất. Trong các *RS*, việc xây dựng *UM* là điều rất quan trọng. Tuy nhiên, việc xây dựng ma trận này thường có gặp nhiều khó khăn. Có hai hướng tiếp cận phổ biến để xác định giá trị *rating* cho mỗi cặp *user-item* trong *UM*:

- Nhờ *users* đánh giá *items*, *Amazon* luôn nhắc nhở *users* bằng cách gửi *email* nhắc nhở nhiều lần và rất nhiều hệ thống khác cũng làm việc tương tự. Tuy nhiên, cách tiếp cận này có một vài hạn chế, do thường thì người dùng ít khi đánh giá sản phẩm. Và nếu có, đó có thể là những đánh giá thiên lệch bởi những người không sẵn sàng đánh giá.

- Hướng tiếp cận thứ hai là dựa trên hành vi của *users*. Rõ ràng, nếu một người dùng mua một sản phẩm trên *Amazon*, xem một đoạn phim trên *Youtube* (có thể là nhiều lần) hay đọc một bài báo thì có thể khẳng định rằng *user* này thích *item* đó. *Facebook* cũng dựa trên việc bạn thích những nội dung nào để hiển thị các bản tin có thông tin, nội dung liên quan phù hợp với bạn. Bạn càng đam mê *Facebook*, thì tập đoàn này càng được hưởng lợi do đó nó luôn mang tới bạn những thông tin mà khả năng cao là bạn muốn đọc. Nhưng với cách này, ta chỉ xây dựng được một ma trận với các thành phần là 1 và 0, với 1 thể hiện người dùng thích sản phẩm, 0 thể hiện chưa có thông tin. Trong trường hợp này, 0 không có nghĩa là thấp hơn 1, nó chỉ có nghĩa là người dùng chưa cung cấp thông tin. Chúng ta cũng có thể xây dựng ma trận với các giá trị cao hơn 1 thông qua thời gian hoặc số lượt mà người dùng xem một sản phẩm nào đó. Đôi khi, nút không thích cũng mang lại những lợi ích nhất định cho hệ thống, lúc này có thể gán giá trị tương ứng bằng -1 chẳng hạn.

# 1.2.2. Phương pháp chuẩn hóa giá trị trong ma trận tiện ích

Có rất nhiều cách đề chuẩn hóa ma trận, trong phạm vi bài nghiên cứu tôi sẽ tập trung vào vấn đề về quá trình chuẩn hóa giá trị ma trận nhằm phục vụ cho *RS-CF.* Do các hệ thống đề xuất, có các vấn đề như đã nêu và một trong số đó vấn đề là thiếu dữ liệu. Nên việc chuẩn hóa ma trận giúp ta dễ dàng hơn trong việc tính toán, cũng như đưa ra các đề xuất một cách phù hợp mà không làm ảnh hưởng đến kết quả đề xuất của hệ thống.

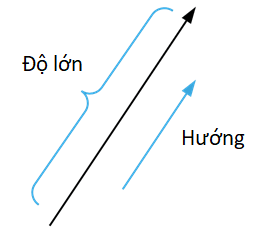
Ngoài ra ,một trong những vấn đề phổ biến trong mô hình đề xuất là *users* khác nhau thì sẽ áp dụng các tiêu chí khác nhau trong việc xác định xếp hạng đánh giá của *items*. Do đó, *user* có cùng sở thích vẫn có thể có những xếp hạng khác nhau cho cùng một *item*. Ví dụ: Một số *user* có khuynh hướng là dễ tính, cho nên xếp hạng các *item* của những *user* này có xu hướng cao hơn so với những *use*r khác. Mặc dù, có những *user* là có độ tương tự với nhau về item đó.

Khi tiến hành tính toán độ tương quan giữa các cặp *item-user*, ma trận ban đầu khi ta xây dựng các *fv* luôn sẽ có các giá trị khuyết. Vấn đề đặt ra sẽ là, chúng ta sẽ xử lý thế nào để điền các giá trị còn khuyết đó mà không ảnh hướng đến bài toán.

Nếu chúng ta, thay dấu ‘0’ vào ma trận, điều này sẽ ảnh hưởng đến quá trình tính độ tương quan do ‘0’ ở đây tượng trưng cho độ quan tâm đến sản phẩm đạt thấp nhất. Ngoài ra, chúng ta cũng không thể điền các số khác thay thế trong thang điểm của ma trận, phương pháp tối ưu nhất cho việc này sẽ được trình bày kĩ hơn ở phần sau.

# 1.2.3. Vectơ và khoảng cách giữa 2 vectơ

Vectơ [10] là một đại lượng có cả độ lớn và hướng. Về mặt hình học, chúng ta có thể hình dung một vectơ là một đoạn đường có hướng, có chiều dài là độ lớn của vectơ và với một mũi tên chỉ hướng. Hướng của vectơ là từ đuôi đến đầu của nó.

Hai vectơ giống nhau nếu chúng có cùng độ lớn và hướng. Điều này có nghĩa là nếu chúng ta lấy một vectơ và dịch nó sang một vị trí mới (không xoay nó), thì vectơ chúng ta thu được ở cuối quá trình này là cùng một vectơ chúng ta đã có lúc đầu.

Hình 1.2: Hình ảnh biểu diễn vectơ với hướng và độ lớn ([Nguồn Internet](https://linuxhint.com/wp-content/uploads/2019/06/1-34.png))

Có nhiều cách tính khoảng cách giữa 2 vectơ (hay tính khoảng cách giữa 2 điểm), một trong những cách phổ biến nhất là tính khoảng cách theo Euclidean [11]

(1)

Một cách tiếp cận khác của tính khoảng cách là tính khoảng cách theo Manhattan [12]. Việc tính khoảng cách, là một công đoạn để tính độ tương tự, khoảng cách giữa hai vật thể càng cao, chúng càng cách xa nhau, hay nói cách khác khi 2 vectơ càng gần nhau thì chúng càng có độ tương tự với nhau cao. Thông thường các số liệu tương tự trả về giá trị từ 0 đến 1, trong đó 0 biểu thị không có sự tương đồng (hoàn toàn không giống nhau) và 1 biểu thị tổng độ tương tự (chúng hoàn toàn giống nhau).

Việc ứng dụng tính khoảng cách 2 vectơ vào việc xác định độ tương tự giữa 2 *users* cũng làđược áp dụng vào trong bài toán hệ tư vấn. Để xác định mức độ quan tâm của *users*, ta sẽ dựa trên hành vi của *user* lên *item*. Để tính toán Sim giữa hai *user* , cách thường làm là xây dựng *feature* *vector (fv)* (vectơ đặc trưng đơn giản là một vectơ có một chuỗi các số, nó giống như một ma trận nhưng chỉ có một hàng nhưng nhiều cột. Ví dụ: [1,5,3]) cho mỗi *user* rồi áp dụng một hàm có khả năng đo độ tương quan giữa hai vectơ đó. Chú ý rằng việc xây dựng *fv* này được xây dựng trực tiếp dựa trên *UM*. Với mỗi *user*, thông tin duy nhất chúng ta biết là các đánh giá mà user đó đã thực hiện, tức là cột tương ứng với *user* đó trong *UM.*

# 1.3. Các phương pháp tư vấn cơ bản

Để xây dựng một *RS*, hiện nay có các phương pháp phổ biến được sử dụng gồm:

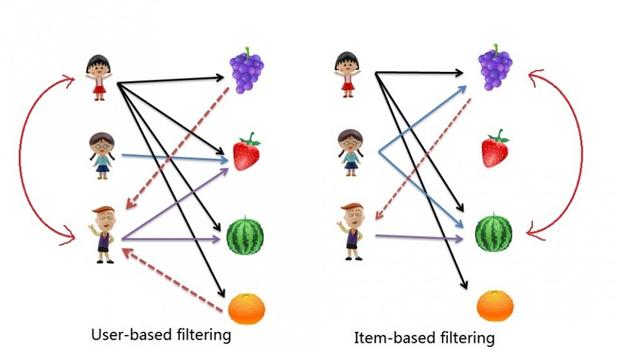
-Lọc trên nội dung (*Content Based Filtering - CBF*) [13]: người dùng được gợi ý những sản phẩm tương tự như các sản phẩm từng được họ đánh giá cao. Mà tiêu chí để đưa ra gợi ý những sản phẩm tương tự dựa trên các mục của thông tin sản phẩm như tên, vị trí, mô tả,v.v.

-Lọc cộng tác (*Collaborative Filtering - CF*) [3]: hệ thống đưa ra các gợi ý trên sự tương quan (*similarity*) giữa các người dùng và/hoặc vật phẩm/hàng hóa sau đó dự đoán những sự quan tâm. Thông qua đó hệ thống sẽ đưa ra những khuyến nghị phù hợp cho người dùng những vật phẩm họ có thể quan tâm; và

-Lai ghép (*Hybrid*): Kết hợp các phương pháp ở trên lại với nhau.

# 1.3.1. Lọc cộng tác

Với mục tiêu chính của khóa luận này sẽ đi sâu vào xây dựng một hệ thống gợi ý, sử dụng phương pháp lọc cộng tác. Khi chỉ nói *CF*, chúng ta sẽ ngầm hiểu rằng phương pháp được sử dụng là *Neighborhood - Based Collaborative Filtering (NBCF)*  [14]. Trong phương pháp lọc cộng tác có nhiều phương pháp khác nhau, trong số đó 2 phương pháp nổi bật là lọc cộng tác dựa trên người dùng (*UBCF*) và lọc cộng tác dựa trên vật phẩm (*IBCF*).



*Hình 1.3: UBCF (trái) và IBCF (phải)*

*(*[*Nguồn Me*](https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwi2yNehybvlAhW9xosBHUlWCZUQjRx6BAgBEAQ&url=https%3A%2F%2Fviblo.asia%2Fp%2Flam-the-nao-de-xay-dung-mot-recommender-system-rs-phan-2-4dbZNxom5YM&psig=AOvVaw3sNe4xm86h9XrRDZRaxuFe&ust=1572235894831799)*dium.com)*

Tất cả các phương pháp *CF*, đều là tính toán độ tương đồng giữa các cặp *users* hoặc *items* được *rating* bởi họ. Có nhiều cách để tính toán độ tương tự và hầu hết các hệ thống *CF* cho phép người dùng sử dụng các cách khác nhau, mà tại đó người dùng có thể xác định cách nào phù hợp nhất với dữ liệu của mình.

Thuật toán *UBCF* là một thuật toán đề xuất, được sử dụng rộng rãi trong nhiều hệ thống đề xuất thương mại. Thuật toán *UBCF*, được dựa trên ý tưởng như sau: Mọi người có sự tương đồng giữa sở thích và các mối quan tâm, sở thích của họ thường là ổn định. Từ đó chúng ta có thể đưa ra các dự đoán về sự lựa chọn của họ dựa trên sở thích trước đây.

Do các giả định trên, *CF* dựa trên việc so sánh hành vi của một người dùng với hành vi của các người dùng khác (hay còn được gọi là người hàng xóm gần nhất- *KNN*), để tìm người giống anh ta. Từ đó, tìm theo các sở thích hoặc mối quan tâm của người hàng xóm (người có độ tương quan, sở thích với ta nhất) mà hệ thống có thể dự đoán rồi đưa ra các đề xuất.

Bước đầu tiên của thuật toán *UBCF* là lấy hồ sơ lịch sử người dùng và biểu diễn dưới dạng một ma trận tiện ích. Bước thứ hai là tính toán sự giống nhau hay còn được xem là bước xác định *độ tương tự* giữa các người dùng với nhau và cuối cùng là tìm nhóm người có độ tượng tự cao nhất của một *user* lên một *item*, để tiến hành dự đoán *rating* còn thiếu trong ma trận.

Trong đó, có rất nhiều phương pháp để đo lường *độ tương tự*: Hệ số tương quan *Pearson* là một trong những cách được sử dụng rộng rãi nhất và được dùng làm chuẩn cho *CF*; ngoài ra người ta cũng thường sử dụng phương pháp đo *Cosine* và rất nhiều phương pháp khác. Trong bài nghiên cứu này, sẽ sử dụng *Cosine* để tính toán độ tương tự.

Để xây dựng một *RS* với phương pháp *CF*, có nhiều cách tiếp cận khác nhau:

- Lọc cộng tác dựa trên bộ nhớ (*memory-based*) cũng được biết đến như là thuật toán lọc cộng tác dựa trên người dùng (*user-based*) (*UBCF*); và

- Lọc cộng tác dựa trên mô hình (*model-based*) là một cách tiếp cận khác được biết đến như là một phương pháp *Kernel-Mapping Recommender.*

Trong bài nghiên cứu này, chúng ta sẽ chỉ tập trung đi vào *UBCF*. Ý tưởng chính của phương pháp này chính là sự tương quan giữa các người có tính cách giống nhau, thì sẽ có sở thích giống nhau.

Ngoài ra, Khi xây dựng RS với CF phải gặp những vấn đề sau: Khởi động lạnh (*cold start*), Độ thưa dữ liệu (*sparsity*) và khả năng mở rộng (*scalability*).

-Khởi động lạnh: Với một người dùng mới hoặc một vật phẩm mới, sẽ là không đủ thông tin để đưa ra đề xuất hay khuyến nghị một cách chính xác được.

-Khả năng mở rộng: Trong nhiều môi trường mà các hệ thống này đưa ra khuyến nghị, có hàng triệu người dùng và sản phẩm. Do đó, một lượng lớn sức mạnh tính toán thường là cần thiết để tính toán các khuyến nghị.

-Độ thưa dữ liệu: Số lượng mặt hàng được bán trên các trang web thương mại điện tử rất là nhiều. Những người dùng tích cực nhất, cũng sẽ chỉ có thể đánh giá một tập hợp con nhỏ của cơ sở dữ liệu tổng thể. Vì vậy, ngay cả những mặt hàng phổ biến nhất cũng có rất ít đánh giá từ người dùng.

Quy trình trên, là cách để xây dựng *RS* theo *UBCF*. Việc áp dụng phương pháp *IBCF* cũng tương tự như vậy. Chỉ khác là, thay vì chúng ta tập trung tìm sự tương quan giữa các người dùng với nhau. Hệ thống sẽ đi tìm sự tương quan giữa các item với nhau, cách làm này được phát minh bởi *Amazon* năm 1998 [15]. Với phương pháp này, các vấn đề của *UBCF* được khắc phục. Do việc thay vì xác định sự tương quan giữa *user-user*, hệ thống lại xác định sự tương quan giữa các *item-item* nên giá trị *rating* trung bình của một *item* thường không thay đổi nhanh chóng. Điều này dẫn đến sự ổn định của mô hình, do mô hình không phải thường xuyên cập nhập lại. Ví dụ: Khi người dùng mua sản phẩm và đánh giá một sản phẩm, các mặt hàng tương tự của mặt hàng đó được chọn từ hệ thống hiện có và đề xuất với người dùng.

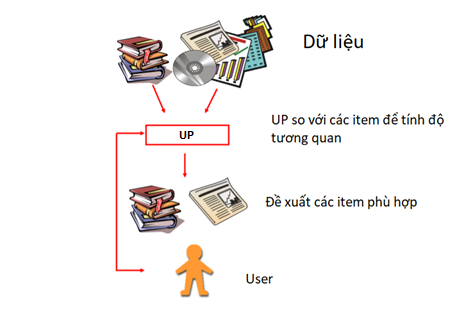
*IBCF* giúp cải thiện đáng kể khả năng mở rộng của *RS* dùng phục vụ cho hàng triệu khách hàng và hàng triệu mặt hàng. Trước khi *IBCF* ra mắt, đã có nhiều hệ thống *UBCF* được phát triển trong giới hàn lâm và ngành công nghiệp, những kỹ thuật này có một số vấn đề về độ phức tạp tính toán cao và khả năng mở rộng bị hạn chế. Nên từ khi thuật toán *IBCF* đã được xuất bản trong *IEEE* *Internet Computing* vào năm 2003 [16], nó đã được áp dụng rộng rãi trên tất cả các gã khổng lồ Website bao gồm *YouTube*, *Netflix* và nhiều website khác.

# 1.3.2. Tư vấn dựa vào nội dung

Một cách tiếp cận phổ biến khác khi thiết kệ *RS* là *CBF*, phương pháp này dựa trên việc thành lập hồ sơ về các item (*Item Profiles - IP*) và việc thành lập hồ sơ về *user* (*User Profile - UP*). Những đề xuất dựa trên nội dung coi đề xuất là một vấn đề về phân loại cụ thể của *user* và học phân loại cho việc thích và không thích trên từng sản phẩm. Hay có thể hiểu đặc điểm của phương pháp *CBF* là việc xây dựng một mô hình cho mỗi *user* không phụ thuộc vào các *users* khác, mà sẽ phụ thuộc hoàn toàn *IP*.

Để đơn giản hơn, chúng ta có thể hiểu CF là phương pháp đề xuất dựa trên việc tìm kiếm mối liên hệ của *users.* Thì UBCF, là một ý tưởng với cách tiếp cận là *users* đang có vẻ quan tâm đến *item* nào, bằng cách xem bộ dữ liệu của *user* đã xem hay tìm kiếm những *item* này thì rất có khả năng cao họ cũng sẽ thích những *item* có nội dung tương tự. Đặc biệt lưu ý, độ tương tự giữa các *item* ở đây được đánh giá dựa trên khía cạnh nội dung của các *item* mà không phải là độ tương tự của các item thông qua các *user* như ở *CF.*

Để hiểu rõ hơn chúng ta có thể xem hình vẽ, được dựa trên những nội dung ta vừa được nêu ở trên. Trong đó từ dữ liệu các vật phẩm ta xây dựng nên một bộ dữ liệu người dùng, qua đó ta có thể đưa ra những khuyến nghị phù hợp với từng người dùng, điều này được dựa trên định nghĩa về CBF của [1]



Hình 1.4: Hình ảnh đơn giản về hệ thống lọc nội dung

Ngoài ra, hệ thống này sử dụng các từ khóa mà những từ khóa này được dùng để mô tả các *items* và *UP* đã được xây dựng để chỉ ra *item* mà *users* thích. Nghĩa là, các thuật toán cố gắng đưa ra đề xuất *items* mà có độ tương tự với *user* đã từng thích *item* nàođó trong quá khứ hoặc đang được tìm kiếm trong hiện tại. Cách tiếp cận này có nguồn gốc từ việc thu hồi thông tin và nghiên cứu lọc thông tin.

Để tạo một UP, hệ thống chủ yếu tập trung vào hai loại thông tin:

-Một mô hình đánh giá của người dùng; và

-Lịch sử của các tương tác người dùng với hệ thống khi đưa ra các khuyến nghị.

Một vấn đề quan trọng với *CBF* là liệu hệ thống có thể tìm hiểu tùy chọn của người dùng từ các hành động của người dùng liên quan đến một nguồn nội dung và sử dụng chúng trên các loại nội dung khác.

Khi hệ thống bị giới hạn trong việc đề xuất nội dung trên cùng loại, giá trị từ RS sẽ bị thấp đi đáng kể so với khi các loại nội dung khác từ các dịch vụ khác có thể được đề xuất. Ví dụ: đề xuất các bài báo dựa trên trình duyệt tin tức là hữu ích, nhưng sẽ hữu ích hơn nhiều khi âm nhạc, video, sản phẩm, thảo luận, vv. Từ các dịch vụ khác nhau có thể được đề xuất dựa trên trình duyệt tin tức. Để khắc phục điều này, hầu hết các hệ thống đề xuất dựa trên nội dung hiện nay đều sử dụng một số dạng hệ thống kết hợp.

Ngoài điểm yếu ở trên, hệ thống còn có hai nhược điểm cơ bản:

- Khi xây dựng mô hình cho một user, các hệ thống *CBF* không tận dụng được thông tin từ các users khác. Những thông tin này thường rất hữu ích vì hành vi mua hàng của *users* thường được nhóm lại với nhau, nên nếu biết hành vi mua hàng của *users* trong nhóm thì sẽ biết đoán được hành vi của *users* còn lại trong nhóm.

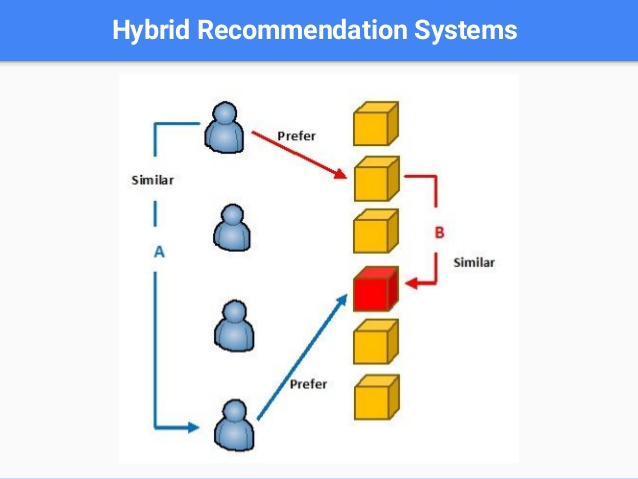
- Không phải lúc nào chúng ta cũng có *IP*, việc yêu cầu *users* gắn thẻ còn khó khăn hơn việc yêu cầu *users* đánh giá *items* đã mua.

Song song với các nhược điểm được kể trên của phương pháp CBF, các điểm mạnh của hệ thống như: Không phụ thuộc vào *user*; tính minh bạch; và dễ dàng với các item chưa được đánh giá bởi bất kỳ user nào.

Một phương pháp được sử dụng rộng rãi sử dụng trong CBG là “Tần suất thuật ngữ và tần suất tài liệu nghịch đảo” (Term frequency and inverse document frequency TF-IDF) (còn được gọi là biểu diễn không gian vectơ) [17]. Hệ thống tạo ra một hồ sơ dựa trên nội dung của người dùng dựa trên trọng số của *fv*. Các trọng số biểu thị tầm quan trọng của từng đặc trưng đối với *user*. Ngoài ra, các cách tiếp cận đơn giản như sử dụng phương pháp giá trị trung bình của vectơ *item* được *rating* trong khi các phương pháp khác sử dụng các kỹ thuật *ML* như: Phân loại Bayes, phân tích cụm (cluster analysis), cây quyết định (decision trees) và mạng nơ ron nhân tạo (artificial neural networks) để ước tính xác suất *user* sẽ thích vật phẩm đó.

# 1.3.3. Phương pháp hỗn hợp (lai ghép)

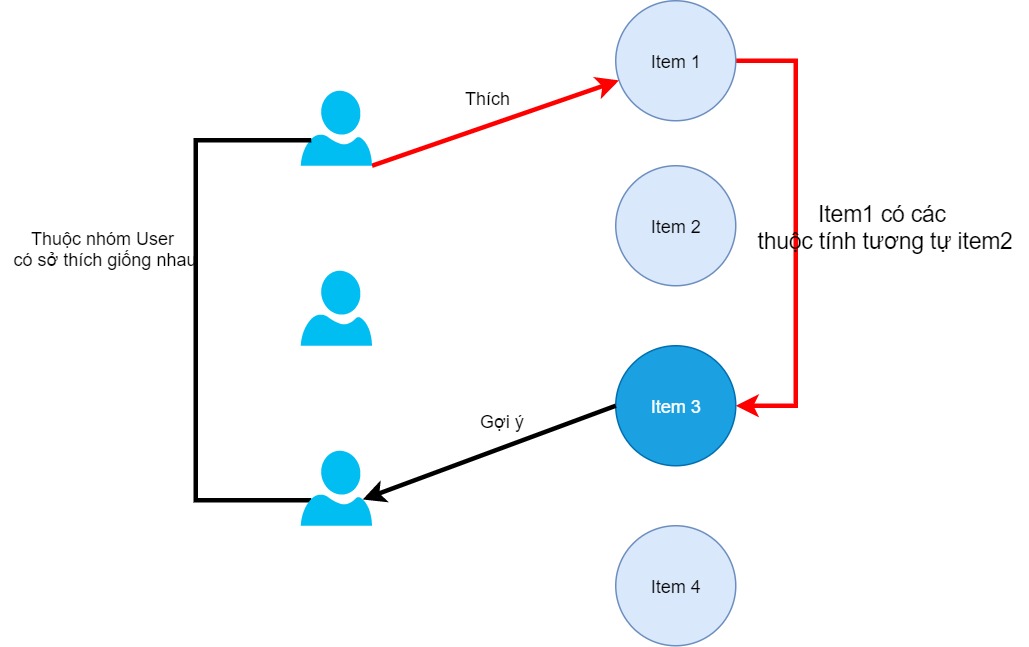
Hầu hết các *RS* hiện sử dụng phương pháp hỗn hợp (hybrid), là kỹ thuật kết hợp 2 phương pháp *CBF*, *CF* và một số cách tiếp cận khác. Rất rõ ràng, ở phía trên chúng ta đã thấy được các điểm yếu cũng như điểm mạnh của CBF và CF. Do đó, không có lý do nào để không kết hợp 2 phương pháp khác nhau nhưng cùng loại lại với nhau.



Hình 1.5: Hình ảnh đơn giản sử dụng phương pháp tư vấn hỗn hợp

*(Nguồn* [*https://www.slideshare.net/*](https://www.slideshare.net/)*)*

Để dễ hình dung dưới đây là một hình ảnh minh họa được vẽ lại



Hình 1.6: Hình ảnh đơn giản sử dụng phương pháp tư vấn hỗn hợp (vẽ lại)

Phương pháp lai có thể được thực hiện theo nhiều cách:

-Bằng cách đưa ra các dự đoán dựa trên nội dung và dự đoán theo lọc cộng tác riêng biệt, rồi sau đó kết hợp chúng (và ngược lại);

-Hoặc bằng cách thống nhất các phương pháp tiếp cận thành một mô hình (xem [18] để đánh giá đầy đủ các *RS*).

Ngoài ra, một số nghiên cứu so sánh thực nghiệm hiệu suất của hybrid với các phương pháp hợp tác và lọc dựa trên nội dung thuần túy, cũng đã được chứng minh rằng các phương pháp lai có thể cung cấp các khuyến nghị chính xác hơn so với các phương pháp thuần túy. Các phương pháp này cũng có thể được sử dụng để khắc phục một số vấn đề phổ biến trong các hệ thống đề xuất.

Netflix là một ví dụ điển hình về việc sử dụng các hệ thống đề xuất lai. Trang web đưa ra khuyến nghị bằng cách so sánh thói quen xem và tìm kiếm của những người dùng tương tự (nghĩa là lọc cộng tác) và cũng bằng cách cung cấp các phim có chung đặc điểm với các phim mà người dùng đánh giá cao (lọc dựa trên nội dung).  
Một vài kỹ thuật mà phương pháp hỗn hợp này sử dụng, bao gồm:

* Có trọng số: Kết hợp số điểm của các thành phần đề xuất khác nhau bằng số.
* Chuyển đổi: Lựa chọn giữa các thành phần khuyến nghị và áp dụng một trong những thành phần được chọn.
* Hỗn hợp: Các khuyến nghị từ các khuyến nghị khác nhau được trình bày cùng nhau để đưa ra khuyến nghị.
* Kết hợp các đặc trưng (Feature Combination**)**: Các đặc trưng xuất phát từ các nguồn kiến ​​thức khác nhau được kết hợp với nhau và được cung cấp cho một thuật toán đề xuất duy nhất.
* Tính năng mở rộng (Feature Augmentation): Tính toán một tính năng hoặc tập hợp các tính năng, sau đó là một phần của đầu vào cho kỹ thuật tiếp theo.
* Cascade: Người giới thiệu được ưu tiên nghiêm ngặt, với những người ưu tiên thấp hơn phá vỡ mối quan hệ trong việc chấm điểm của những người cao hơn.
* Cấp độ meta: Một kỹ thuật đề xuất được áp dụng và tạo ra một số loại mô hình, sau đó là đầu vào được sử dụng bởi kỹ thuật tiếp theo.

# 1.4. Vấn đề “khởi động lạnh” (Cold-start)

Một trong những vấn đề lớn nhất mà các *RS* gặp phải chỉnh là thiếu dữ liệu. Chúng ta gọi thuật ngữ này là “cold-start”, giống như khi khởi động ô tô, động cơ ô tô ban đầu là bị lạnh, thì sẽ có vấn đề với việc khởi động, nhưng một khi nó đạt đến nhiệt độ hoạt động tối ưu, nó sẽ chạy trơn tru. Vấn đề trên có thể hiểu đơn giản hơn nghĩa là, khi bắt đầu khởi dựng hệ thống thì các kết quả chúng ta đề xuất sẽ chưa tối ưu và đưa ra kết quả tốt nhất.

Khởi động lạnh là một vấn đề tiềm ẩn trong các hệ thống thông tin dựa trên máy tính liên quan đến một mức độ mô hình hóa dữ liệu tự động. Cụ thể, nó liên quan đến vấn đề hệ thống không thể rút ra bất kỳ suy luận nào cho người dùng hoặc các mục mà nó chưa thu thập đủ thông tin.

Trong RS, có ba loại khởi động lạnh, khác nhau:

* Khởi động lạnh cộng đồng (*community cold-start*): Vấn đề cộng đồng mới, đề cập đến sự khởi động ban đầu của hệ thống khi hầu như không có thông tin nào mà hệ thống có thể dựa vào. Đây là trường hợp của cả 2 trường hợp dưới,vì tất cả các *item* và *user* đều mới. Do đó, một số kỹ thuật được phát triển để đối phó với hai trường hợp này không áp dụng được cho việc khởi động hệ thống;
* Khởi động lạnh sản phẩm(*item cold-start*): Khác với vấn đề trên thì khởi động lạnh sản phẩm là vấn đề khi ta vừa thêm mới một *item* vào hệ thống. Về mặc lý thuyết, vấn đề này mang ảnh hưởng rất lớn đề *RS-CF* và đặc biệt là *IBCF* . Khi mà việc đề xuất được tập trung vào tính độ tương quan giữa các *item* dẫn đến các item vừa được thêm vào bị hệ thống bỏ qua, và chỉ đề xuất những *item* phổ biến. Dần dần, khiến các *item* phổ biến ngày một phổ biến, còn các *item* mới khó lòng có thể được đề xuất trên hệ thống*.* Ngược lại với *RS-CBF* lại ít bị ảnh hưởng hơn, do CBF dựa vào IP để đưa ra các đề xuất phù hợp;và
* Khởi động lạnh người dùng (*user cold-start*): Trường hợp người dùng mới truy cập hay đăng ký vào hệ thống và chỉ vừa sử dụng hệ thống trong một khoảng thời gian nhất định. Khi đó *RS* phải đưa ra đề xuất mà không cần dựa vào các tương tác trong quá khứ của người dùng, vì chưa có gì xảy ra nên hệ thống không có đủ dữ liệu để đưa ra các đề xuất phù hợp được.

Ở trên là các vấn đề ta gặp phải khi xây dựng một *RS,* và nhiều chiến lược để giảm thiểu vấn đề khởi động lạnh đã được phát triển. Cách tiếp cận chính là dựa vào các khuyến nghị hỗn hợp được đề cập ở phần trên, từ đó giảm thiểu những nhược điểm của một loại hoặc mô hình bằng cách kết hợp nó với một loại khác. Do các chiến lược nhằm khắc phục các vấn đề được sử dụng bằng cách kết hợp các phương pháp lại. Nên nằm ngoài phạm vi của đề tài, do đó đây là một trong những lỗ hổng mà đề tài chưa thể giải quyết, và cần được nghiên cứu và phát triển thêm.

# 1.5. Phát biểu bài toán nghiên cứu

Gọi: - *U={u1,u2,…,um}* là tập hợp tất các người dùng (*user*),

* *I={i1,i2,…,in}* là tập hợp các sản phẩm (*item*),
* S*={s1,s2,…,sk}* là tập hợp các thang đo.

Các *user* d tiến hành đánh giá từng *item* theo thang đo . Tập hợp dữ liệu đánh giá này đánh giá được gọi là ma trận đánh giá (rating matrix). Ma trận được mô tả như sau:

**Bảng 2: Bảng ma trận đánh giá**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | u0 | u1 | ..….. | u m |
| i0 | 5 | 5 | ..….. | ? |
| i1 | 4 | ? | ..….. | ? |
| ….. | ? | 4 | ..….. | 1 |
| ….. | 2 | 2 | ..….. | 4 |
| in | 2 | 0 | ..….. | 5 |

Trong đó ma trận ; .

Bài toán hệ tư vấn đặt ra là dự đoán đánh giá của *user* *uj* đối với *item* ik (*rkj)* mà *user* đó chưa từng đánh giá trước đây.

# 1.6. Các công trình liên quan

Theo Adomavicius và Tuzhilin trong [18], đề xuất một giải pháp dự đoán sở thích của người dùng dựa trên kỹ thuật phân rã ma trận (Matrix Factorization – MF) có tích hợp yếu tố thời gian trong hệ thống gợi ý (Recommender Systems – RS). Do sở thích của người dùng có thể thay đổi theo thời gian, để kết quả gợi ý có độ chính xác cao hơn chúng tôi đề xuất tích hợp phương pháp dự báo san bằng hàm mũ (Exponential Smoothing - ES) vào mô hình Tensor Factorization với mục tiêu khai thác và tận dụng được các thông tin về thời gian cũng như trình tự (sequence) mà người dùng đã đưa ra phản hồi. Thực nghiệm ban đầu trên các tập dữ liệu chuẩn trong lĩnh vực gợi ý và đánh giá bằng độ đo RMSE (Root Mean Squared Error) đã cho thấy hướng tiếp cận này cho kết quả rất khả quan.

Theo Michael D. Ekstrand, John T. Riedl và Joseph A. Konstan [19], hệ thống giới thiệu là một phần quan trọng của thông tin và hệ sinh thái trong thương mại điện tử. Chúng đại diện cho một phương pháp mạnh mẽ để cho phép người dùng để lọc một lượng lớn thông tin với nhiều loại sản phẩm. Gần hai thập kỷ vừa qua, nghiên cứu về lọc cộng tác đã dẫn đến một loạt các thuật toán và các công cụ được dùng để đánh giá hiệu suất của chúng. Nghiên cứu trong lĩnh vực này đang ngày càng phát triển và trở nên thông minh hơn về cách thức hay các kỹ thuật công nghệ được nhúng vào các tên miền xác định. Tính độc đáo được thể hiện bởi các thuật toán đề xuất khác nhau cho thấy rằng hệ tư vấn, khuyến nghị không thể được giải quyết bằng chỉ một thuật toán. Với những nhu cầu mang tính đặc biệt, thì nhu cầu thông tin và các tên miền đại diện duy nhất cho các vấn đề của người giới thiệu, thiết kế và đánh giá của người tư vấn cần được thực hiện dựa trên các tác vụ của người dùng để được hỗ trợ. Để việc triển khai trở nên hiệu quả phải được bắt đầu bằng việc phân tích rất cẩn thận của các người dùng tiềm năng và mong muốn của họ. Dựa trên phân tích này, các nhà thiết kế hệ thống đã có một loạt các tùy chọn cho sự lựa chọn thuật toán và cho nó nhúng vào các hệ thống để đánh giá, dự đoán dựa trên những trải nghiệm của người dùng xung quanh. Bài viết, đưa ra một loạt các lựa chọn có sẵn và ý nghĩa của chúng, nhằm giới thiệu cho cả các học viên và các nhà nghiên cứu về tầm quan trọng và cách tiếp cận về vấn đề khuyến nghị cơ bản và nghiên cứu giải pháp tốt nhất hiện nay nhằm giải quyết một số vấn đề.

Giải quyết vấn với các người dùng mới trong khuyến nghị hệ thống - “ Addressing Cold-Start Problem in Recommendation Systems” – Xuan Nhat Lam, Thuc Vu, Truong Duc Le và Anh Duc Vuong [20], hệ thống tư vấn đưa ra các gợi ý tự động mà người dùng quan tâm là trở nên ngày càng phổ biến trong các lĩnh vực, mà ở đó tính cá nhân hóa là được đánh giá cao. Kỹ thuật được cho là phổ biến trong hệ thống tư vấn là kỹ thuật lọc cộng tác (Collaborative Filtering); lọc nội dung (Content-based Filtering) và kỹ thuật lai. Trong bài báo này, Sẽ nghiên cứu về cách tiếp cận kỹ thuật lai, việc sử dụng lọc cộng tác và đồng thời lọc nội dung để giải quyết vấn đề, đó là đưa ra các khuyến nghị cho những người dùng mới chưa có cơ sở để hệ thống đưa ra các khuyến nghị phù hợp với người dùng đó. Bởi vì, các nghiên cứu về giải pháp lọc cộng tác dựa trên sản phẩm, hay các giải pháp lọc cộng tác dựa trên người dùng là không thể giải quyết vấn đề này. Do đó, chúng tôi phát triển một mô hình áp dụng kỹ thuật lai dựa trên phân tích mô hình khía cạnh xác suất bằng cách sử dụng bộ lọc cộng tác thuần túy để kết hợp với thông tin của người dùng. Các thử nghiệm với dữ liệu Movielens cho thấy những cải tiến đáng kể và nhất quán của mô hình này trong việc khắc phục vấn đề đưa ra khuyến nghị với những người dùng mà hệ thống không có cơ sở để đưa ra khuyến nghị phù hợp với các phương pháp lọc cộng tác khác.

Theo Francesco Ricci, Lior Rokach và Bracha Shapira [21]: Hệ thống đề xuất (RS) là các công cụ và kỹ thuật phần mềm cung cấp các đề xuất cho các mục được cho rằng sẽ được sử dụng cho người dùng. Trong chương giới thiệu này, chúng tôi thảo luận ngắn gọn về các ý tưởng và khái niệm RS cơ bản. Mục tiêu chính của chúng tôi là phân định một cách mạch lạc và có cấu trúc, các chương có trong cuốn cẩm nang này và giúp người đọc điều hướng nội dung cực kỳ phong phú và chi tiết mà cuốn cẩm nang cung cấp.

Khảo sát kỹ thuật lọc cộng tác - A Survey of Collaborative Filtering Techniques của Xiaoyuan Su and Taghi M. Khoshgoftaar [22] là một trong những cách tiếp cận thành công nhất nhằm xây dựng các hệ thống tư vấn, CF sử dụng các tùy chọn đã biết của một nhóm người dùng để đưa ra đề xuất hoặc dự đoán về các tùy chọn chưa biết cho người dùng khác. Trong bài báo này, đầu tiên chúng tôi giới thiệu các nhiệm vụ và các thách thức chính của chúng, chẳng hạn như data sparsity, scalability, synonymy, gray sheep, shilling attacks, privacy protection, v.v. và các giải pháp khả thi của chúng. Sau đó, chúng tôi trình bày ba loại kỹ thuật CF chính: thuật toán CF dựa trên bộ nhớ (memory-based), dựa trên mô hình (*model-based*) và kỹ thuật lai (kết hợp CF với các kỹ thuật đề xuất khác), với các ví dụ về thuật toán đại diện của từng loại và phân tích hiệu suất dự đoán và khả năng của chúng để giải quyết những thách thức. Từ các kỹ thuật cơ bản đến hiện đại, chúng tôi cố gắng trình bày một cuộc khảo sát toàn diện về các kỹ thuật CF, có thể được dùng làm trong quá trình nghiên cứu và thực hành trong lĩnh vực này.

Theo Prem Melville and Vikas Sindhwani [23], mục tiêu của hệ tư vấn là đưa ra các đề xuất có ý nghĩa cho việc thu thập dữ liệu của người dùng đối với các sản phẩm hoặc mặt hàng có thể khiến họ quan tâm đến chúng. Gợi ý về sách trên Amazon, hoặc phim trên Netflix là những ví dụ cụ thể về sự hoạt động mạnh mẽ của hệ thống đề xuất. Thiết kế của các hệ tư vấn (recommendation engines) phụ thuộc vào miền và các đặc điểm cụ thể của dữ liệu có sẵn. Ví dụ: người xem phim trên Netflix thường cung cấp xếp hạng theo thang điểm từ 1 (không thích) đến 5 (thích). Một nguồn dữ liệu như vậy ghi lại chất lượng tương tác giữa người dùng và vật phẩm. Ngoài ra, hệ thống có thể có quyền truy cập vào các thuộc tính hồ sơ cụ thể của người dùng và mục cụ thể, chẳng hạn như nhân khẩu học và mô tả sản phẩm, tương ứng. Các hệ thống đề xuất khác nhau về cách họ phân tích các nguồn dữ liệu này để phát triển các khái niệm về mối quan hệ giữa người dùng và các mục, có thể được sử dụng để xác định các cặp được kết hợp tốt.

“A survey of active learning in collaborative filtering recommender systems” - Elahi, Mehdi; Ricci, Francesco; Rubens, Neil [24]. trong các hệ thống đề xuất lọc cộng tác, các tùy chọn của người dùng được biểu thị dưới dạng xếp hạng cho các mục và mỗi xếp hạng được thêm vào sẽ mở rộng kiến ​​thức của hệ thống và ảnh hưởng đến độ chính xác của hệ thống. Nói chung, càng nhiều xếp hạng được gợi ra từ người dùng, các khuyến nghị càng chính xác. Tuy nhiên, các lợi ích của từng xếp hạng có thể khác nhau đáng kể, tức là, các xếp hạng khác nhau có thể mang lại một lượng và loại thông tin khác nhau về thị hiếu của người dùng. Do đó, các kỹ thuật cụ thể, được định nghĩa như là chiến lược học tập tích cực “active learning strategies” có thể được sử dụng để chọn lọc các mục được trình bày cho người dùng để đánh giá. Trên thực tế, một chiến lược học tập tích cực xác định và áp dụng các tiêu chí để có được dữ liệu phản ánh tốt hơn các tùy chọn của người dùng và cho phép tạo các đề xuất tốt hơn.

Cho đến nay, một loạt các chiến lược học tập tích cực đã được đề xuất trong tài liệu. Trong bài viết này, chúng tôi khảo sát các chiến lược gần đây bằng cách nhóm chúng theo hai khía cạnh khác nhau: cá nhân hóa, tức là, liệu các mục được chọn của hệ thống có khác nhau đối với người dùng khác nhau hay không, và, ví dụ, liệu học tập tích cực có được hướng dẫn bởi một tiêu chí không (heuristic) hoặc theo nhiều tiêu chí. Ngoài ra, chúng tôi trình bày tổng quan toàn diện về các phương pháp đánh giá và số liệu đã được cộng đồng nghiên cứu sử dụng để kiểm tra các chiến lược học tập tích cực để lọc cộng tác. Cuối cùng, chúng tôi so sánh các chiến lược được khảo sát và cung cấp các hướng dẫn cho việc sử dụng chúng trong các hệ thống đề xuất.

Theo Recommendation System Based on Collaborative Filtering - Zheng Wen [25], hệ tư vấn là một loại kỹ thuật chuyên lọc thông mà tại đó nó trình bày các mục thông tin (như phim ảnh, âm nhạc, trang web, tin tức) có khả năng được người dùng quan tâm. Nó đóng một tầm quan trọng lớn trong sự thành công của ngành thương mại điện tử và công nghệ thông tin ngày nay, và dần đang được phổ biến trong các ứng dụng khác nhau (ví dụ: dự án Netflix, tin tức Google, Amazon). Theo trực giác, một hệ tư vấn xây dựng hồ sơ người dùng dựa trên quá khứ của anh ấy / cô ấy và so sánh nó với một số đặc điểm tham chiếu rồi tìm cách dự đoán “xếp hạng- rating” mà người dùng chưa đánh giá. Trong hầu hết các trường hợp, hệ tư vấn tương ứng với một vấn đề khai thác dữ liệu quy mô lớn. Dựa trên sự lựa chọn các đặc điểm tham chiếu, một hệ tư vấn có thể dựa trên phương pháp tiếp cận dựa trên nội dung hoặc phương pháp lọc cộng tác (CF) hoặc cả hai.

Theo Latent Semantic, Models for Collaborative Filtering THOMAS HOFMANN Brown University [26], lọc cộng tác nhằm mục đích tìm hiểu các mô hình dự đoán về sở thích, quan tâm hoặc hành vi của người dùng từ dữ liệu cộng đồng, nghĩa là một cơ sở dữ liệu về các tài liệu tham khảo người dùng có sẵn. Trong bài viết này, chúng tôi mô tả một loại thuật toán dựa trên mô hình mới được thiết kế cho nhiệm vụ này. Các thuật toán này dựa trên một kỹ thuật mô hình thống kê giới thiệu các biến lớp ẩn trong một thiết lập mô hình hỗn hợp để khám phá cộng đồng người dùng và hồ sơ quan tâm nguyên mẫu. Chúng tôi điều tra một số biến thể để đối phó với các biến trả lời rời rạc và liên tục cũng như với các hàm mục tiêu khác nhau. Ưu điểm chính của kỹ thuật này so với các phương pháp dựa trên bộ nhớ tiêu chuẩn là độ chính xác cao hơn, dự đoán thời gian không đổi và biểu diễn mô hình rõ ràng và nhỏ gọn. Cái sau cũng có thể được sử dụng để khai thác cho cộng đồng người dùng. Đánh giá thử nghiệm cho thấy những cải thiện đáng kể về sự phù hợp so với các phương pháp hiện có và kết quả được công bố có thể thu được.

Theo Yehuda Koren, Robert Bell and Chris Volinsky, Matrix factorization techniques for recommender system [27]*,* người tiêu dùng hiện nay đang tràn ngập các lựa chọn. Các nhà bán lẻ điện tử và các nhà cung cấp nội dung cung cấp nhiều sự lựa chọn cho các sản phẩm, với những cơ hội chưa từng có để đáp ứng nhiều nhu cầu và thị hiếu đặc biệt. Đề xuất người tiêu dùng với các sản phẩm phù hợp nhất là chìa khóa để nâng cao sự hài lòng và sự tin dùng của người dùng. Do đó, nhiều nhà bán lẻ đã quan tâm đến các hệ tư vấn, phân tích các mô hình quan tâm của người dùng đối với các sản phẩm để cung cấp các đề xuất được cá nhân hóa phù hợp với sở thích của người dùng. Vì các đề xuất được cá nhân hóa tốt có thể thêm một khía cạnh khác cho trải nghiệm người dùng, các nhà lãnh đạo thương mại điện tử như Amazon.com và Netflix đã biến các hệ thống đề xuất trở thành một phần nổi bật trang web của họ. Nói rộng hơn, các hệ thống đề xuất dựa trên một trong hai chiến lược. Phương pháp lọc nội dung tạo một hồ sơ cho từng người dùng hoặc sản phẩm để mô tả nó. Ví dụ: hồ sơ phim có thể bao gồm các thuộc tính liên quan đến thể loại của nó, các diễn viên tham gia, mức độ phổ biến phòng vé của nó, v.v. Hồ sơ người dùng có thể bao gồm thông tin nhân khẩu học hoặc câu trả lời được cung cấp trên một bảng câu hỏi phù hợp. Các cấu hình cho phép các chương trình liên kết người dùng với các sản phẩm phù hợp. Tất nhiên, các chiến lược dựa trên nội dung yêu cầu thu thập thông tin bên ngoài có thể không có sẵn hoặc dễ thu thập. Các hệ tư vấn dựa trên các loại dữ liệu đầu vào khác nhau, thường được đặt trong một ma trận với một kích thước đại diện cho người dùng và kích thước khác đại diện cho các mục quan tâm. Dữ liệu thuận tiện nhất là phản hồi rõ ràng chất lượng cao, bao gồm đầu vào rõ ràng của người dùng liên quan đến sự quan tâm của họ đối với sản phẩm. Ví dụ: Netflix thu thập xếp hạng sao cho phim và người dùng TiVi cho biết tùy chọn của họ đối với các chương trình truyền hình bằng cách nhấn nút thích hoặc không thích. Chúng tôi đề cập đến sự phản hồi rõ ràng của người dùng như xếp hạng. Thông thường, phản hồi rõ ràng bao gồm một ma trận thiếu dữ liệu, vì bất kỳ người dùng nào cũng có thể chỉ đánh giá một tỷ lệ nhỏ các mặt hàng có thể.

Theo Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl trong Item-Based Collaborative Filtering Recommendation Algorithms [28], các hệ tư vấn áp dụng các kỹ thuật khai phá dữ liệu cho vấn đề đưa ra các đề xuất các thông tin, sản phẩm hoặc dịch vụ phù hợp với từng cá nhân trong quá trình tương tác trực tiếp. Các hệ thống này, đặc biệt là các hệ thống lọc cộng tác láng giềng gần nhất -k, đang đạt được sử dụng rộng rãi trên website. Sự tăng trưởng vượt bậc về lượng thông tin có sẵn và số lượng khách truy cập vào các trang web trong những năm gần đây đặt ra một số thách thức chính cho các hệ tư vấn. Đó là: tạo ra các đề xuất chất lượng cao, thực hiện nhiều đề xuất mỗi giây cho hàng triệu người dùng và vật phẩm đồng thời đạt được độ bao phủ cao khi đối mặt với sự thiếu dữ liệu. Trong các hệ thống lọc cộng tác truyền thống, lượng công việc tăng theo số lượng người tham gia trong hệ thống. Các công nghệ hệ tư vấn mới là cần thiết để có thể nhanh chóng tạo ra các khuyến nghị chất lượng cao và để có giải quyết ngay cả đối với các vấn đề quy mô rất lớn. Để giải quyết những vấn đề này, chúng tôi đã tìm hiểu các kỹ thuật lọc cộng tác dựa trên vật phẩm. Các kỹ thuật dựa trên vật phẩm trước tiên phân tích ma trận mục người dùng để xác định mối quan hệ giữa các mục khác nhau và sau đó sử dụng các mối quan hệ này để gián tiếp tính toán các đề xuất cho người dùng.

Trong bài báo này, chúng tôi phân tích các thuật toán tạo đề xuất dựa trên vật phẩm khác nhau. Chúng tôi xem xét các kỹ thuật khác nhau để tính toán sự tương đồng của vật phẩm (ví dụ: tương quan vật phẩm với sự tương đồng cosin giữa các vectơ vật phẩm) và các kỹ thuật khác nhau để có được các đề xuất từ ​​chúng (ví dụ: tổng trọng số so với mô hình hồi quy). Cuối cùng, chúng tôi thực nghiệm đánh giá kết quả của chúng tôi và so sánh chúng với phương pháp tiếp cận cơ bản gần nhất của k. Các thử nghiệm của chúng tôi cho thấy các thuật toán dựa trên vật phẩm cung cấp có hiệu suất tốt hơn đáng kể so với các thuật toán dựa trên người dùng, đồng thời cung cấp chất lượng tốt hơn các thuật toán dựa trên người dùng tốt nhất hiện có.

Theo Léon Bottou, Stochastic Gradient Descent Tricks, Microsoft Research, Redmond [29], ủng hộ phương pháp lan truyền ngược ngẫu để huấn luyện mạng lưới thần kinh (neural networks). Trên thực tế đây là một ví dụ của kỹ thuật tổng quát hơn gọi là stochastic gradient descent (SGD). Tài liệu này cung cấp dữ liệu nền, giải thích tại sao SGD là một thuật toán tốt khi sử dụng để huấn luyện một tập dữ liệu lớn và cung cấp các khuyến nghị có ích.

Luận án Matrix Factorization Methods for Recommender Systems [30], luận án này là một nghiên cứu tổng quan về các phương pháp nhân tố ma trận được sử dụng trong các hệ thống đề xuất. Nó nghiên cứu và phân tích các mô hình hiện có, cụ thể là các mô hình xác suất được sử dụng cùng với các phương pháp nhân tố ma trận, cho các hệ tư vấn từ góc độ học máy. Luận án còn thực hiện hai phương pháp khác nhau được đề xuất trong tài liệu khoa học và tiến hành các thí nghiệm về độ chính xác dự đoán của các mô hình trên Yahoo.

Theo Ron zacharski, A programmer’s Guide to Data Mining, The Ancient Art of the Numerati [31], nghiên cứu này áp dụng một sự đổi mới trong việc phát triển mua sắm trực tuyến, sử dụng hệ tư vấn. Hệ tư vấn áp dụng kỹ thuật tìm kiếm kiến thức được gọi là “Lọc cộng tác dựa trên vật phẩm”. Được hoạt động bằng cách xây dựng thông tin về các mặt hàng được khách hàng ưa thích. Lọc cộng tác sắp xếp dữ liệu dựa trên sự tương đồng hoặc đặc điểm nhất định, để hệ thống có thể cung cấp thông tin dựa trên các mẫu từ một nhóm dữ liệu gần giống nhau. Với hệ tư vấn, khách hàng có thể hưởng lợi từ các mặt hàng được hệ thống tự động tạo ra các đề xuất mà họ có thể quan tâm. Với hy vọng rằng nó có thể cải thiện sự thuận tiện để mua sắm và giảm thời gian khách hàng cần để tìm kiếm các mặt hàng. Do đó, nó có thể tăng khả năng cạnh tranh của các cửa hàng trực tuyến mà có sử dụng hệ tư vấn so với các cửa hàng trực tuyến không áp dụng kỹ thuật này.

Theo Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl [32], các hệ thống đề xuất đã được đánh giá theo nhiều cách, thường là không thể so sánh được. Trong bài viết này, chúng tôi xem xét các quyết định chính trong việc đánh giá các hệ thống đề xuất lọc cộng tác: các tác vụ người dùng được đánh giá, các loại phân tích và bộ dữ liệu được sử dụng, cách đo lường chất lượng dự đoán, đánh giá các thuộc tính dự đoán khác với chất lượng và đánh giá dựa trên người dùng của toàn bộ hệ thống. Ngoài việc xem xét các chiến lược đánh giá được sử dụng bởi các nhà nghiên cứu trước đây, chúng tôi trình bày kết quả thực nghiệm từ phân tích các số liệu chính xác khác nhau trên một miền nội dung trong đó tất cả các số liệu được kiểm tra đã thu gọn thành ba lớp tương đương. Các số liệu trong mỗi lớp tương đương có mối tương quan mạnh mẽ, trong khi các số liệu từ các lớp tương đương khác nhau không tương quan.

Theo N. Nikzad–Khasmakhi, M.A. Balafar và M. Reza Feizi–Derakhshi [33], sự phát triển nhanh chóng gần đây của nội dung Internet đã dẫn đến việc xây dựng các hệ tư vấn hướng dẫn người dùng theo nhu cầu của họ thông qua quy trình truy xuất thông tin. Một hệ tư vấn chuyên nghiệp là một lĩnh vực mới phát triển đang cố gắng phát hiện, đưa ra những thông tin phù hợp nhất trong một số chủ đề cụ thể. Các đề xuất này được đưa ra dựa trên các thông tin được trích xuất từ các hoạt động của người dân và nội dung của các tài liệu liên quan đến họ. Hơn nữa, một hệ tư vấn chuyên nghiệp lấy một chủ đề hoặc truy vấn người dùng rồi sau đó cung cấp một danh sách những người được sắp xếp theo mức độ chuyên môn liên quan của họ với chủ đề hoặc truy vấn nhất định. Các hệ thống này có thể được mô hình hóa bằng các phương pháp truy xuất thông tin, cùng với các công cụ tìm kiếm hoặc kết hợp các hệ thống xử lý ngôn ngữ tự nhiên. Nghiên cứu sau đây cung cấp một cái nhìn tổng quan quan trọng về các hệ tư vấn hiện có và các ưu điểm và nhược điểm của chúng, xem xét các kỹ thuật khác nhau được sử dụng bởi chúng.

Trong nghiên cứu “Time and ontology for resource recommendation system” [34], ngày nay các hệ thống gắn thẻ xã hội có sự tăng trưởng đáng kể. Các hệ thống này giúp người dùng tìm thấy các tài nguyên yêu thích của họ trong số lượng lớn thông tin. Kể từ khi các thẻ là các ý nghĩa của người dùng thể hiện sở thích và thời điểm gắn thẻ cho biết sở thích hiện tại của người dùng, do đó, sự kết hợp ảnh hưởng ngữ nghĩa của thẻ và thời gian của thông tin gắn thẻ có thể ảnh hưởng đến độ chính xác của các khuyến nghị. Trong bài viết này, một hệ thống đề xuất mục được đề xuất rằng bằng cách sử dụng thông tin quan trọng có sẵn trong các hệ thống gắn thẻ, ví dụ: thời gian và sử dụng các bản thể học có sẵn, độ chính xác của các kết quả được hệ tư vấn là đã được cải thiện. Việc đánh giá hệ thống đề xuất được thực hiện trên bộ dữ liệu của Movielens.org. Kết quả so sánh với các phương pháp khác đã chứng minh chất lượng cải thiện của hệ thống được đề xuất.

Theo Morteza Zihayat, Anteneh Ayanso, Xing Zhao, Heidar Davoudi, Aijun An [35], Các nền tảng tin tức thể hiện cả những thách thức cũng như cơ hội để tăng cường chức năng của các hệ tư vấn trong môi trường dữ liệu lớn ngày nay. Việc sử dụng các mô hình lập trình và lưu trữ dữ liệu lớn có thể cải thiện các hệ thống đề xuất tin tức thông qua xử lý và phân tích hiệu quả dữ liệu nhấp chuột và hiểu rõ hơn về lợi ích của người dùng. Hầu hết các cách tiếp cận hiện tại đối với đề xuất tin tức đều coi các nhấp chuột của người dùng là phản hồi ngầm để hiểu hành vi của người dùng. Tuy nhiên, các nhấp chuột có thể không phải là một chỉ số hiệu quả về lợi ích người dùng thực sự. Chúng tôi giải quyết vấn đề này bằng cách phát triển một hệ tư vấn tin tức mới dựa trên mô hình tiện ích tin tức. Với mô hình tiện ích mới, chúng tôi đề xuất một khung khuyến nghị tin tức hai giai đoạn. Khung trước tiên tạo quy tắc đề xuất cấp bài viết dựa trên mô hình tiện ích, sau đó tích hợp khái niệm mô hình chủ đề tiện ích và xác suất và tạo quy tắc đề xuất cấp chủ đề. Chúng tôi lập luận rằng hệ tư vấn tin tức dựa trên tiện ích được đề xuất cũng giải quyết vấn đề bắt đầu lạnh tin tức là một trong những trở ngại thách thức nhất đối với các cơ quan báo chí. Chúng tôi đánh giá khuôn khổ trên một bộ dữ liệu thực lớn (hai tỷ bản ghi) thu được từ một tờ báo lớn (tức là Quả cầu và Thư) ở Canada và cho thấy rằng nó vượt trội so với các phương pháp hiện có.

Trong nghiên cứu Hybrid Filtrations Recommendation System based on Privacy Preserving in Edge Computing [36], thật khó khăn khi thiết kế một hệ tư vấn an toàn trên Internet, điều này có thể giúp người dùng chọn các sản phẩm yêu thích của họ càng ít bị rò rỉ quyền riêng tư càng tốt. Trong bài báo này, chúng tôi trình bày một hệ tư vấn lọc hỗn hợp dựa trên việc bảo vệ quyền riêng tư trong điện toán cạnh (HFRS-PP), có thể ngăn chặn thông tin bảo mật của người dùng bị rò rỉ thông qua các ưu điểm của điện toán cạnh trong quá trình tính toán và đảm bảo tính thực -thời gian, độ chính xác và ổn định của kết quả truy vấn. Đặc biệt, chúng tôi đề xuất một thuật toán khuyến nghị để bảo vệ quyền riêng tư và thu được kết quả mong muốn cho người dùng cuối thông qua các phép lọc lai. Thuật toán lý thuyết bộ lọc thô được đưa ra để phân biệt các đánh giá hợp lệ với các đánh giá spam cho lần lọc tiếp theo.

Dựa trên nghiên cứu “ Development of Mobile System for Medical Recommendations” [37], bài viết này xem xét vấn đề hiện tại của việc điều tra ứng dụng di động và phát triển hệ thống di động cho các khuyến nghị y tế. Bài viết phát triển một phương pháp để lựa chọn các yếu tố cấu thành để xây dựng một mẫu văn bản. Một thuật toán cho các tìm kiếm trùng lặp mờ trong các văn bản ngôn ngữ tự nhiên đã được xây dựng. Bài viết này trình bày phương pháp hình thành một mẫu mà không có trong cơ sở dữ liệu mẫu và phương pháp phân bổ một dấu hiệu thực dụng từ một văn bản ngôn ngữ tự nhiên bán cấu trúc được phát triển. Cuối cùng, các tác giả trình bày mô hình và phát triển hệ thống di động cho khuyến nghị y tế.

Theo Shuo Yanga, Mohammed Korayemb, Khalifeh AlJadda, Trey Grainger và Sriraam Natarajana [38], Các hệ tư vấn thường liên quan đến việc khai thác mối quan hệ giữa các đặc trưng và nội dung đã biết mô tả các mục (lọc dựa trên nội dung) hoặc sự chồng chéo của những người dùng tương tự đã tương tác hoặc đánh giá mục tiêu (lọc cộng tác). Để kết hợp hai phương pháp lọc này, các hệ tư vấn lai dựa trên mô hình hiện tại thường yêu cầu kỹ thuật tính năng mở rộng để xây dựng hồ sơ người dùng. Học tập quan hệ thống kê (SRL) cung cấp một cách đơn giản để kết hợp hai cách tiếp cận thông qua khả năng đại diện trực tiếp cho các phụ thuộc xác suất giữa các thuộc tính của các đối tượng liên quan. Tuy nhiên, do quy mô lớn của dữ liệu được sử dụng trong các hệ tư vấn trong thế giới thực, có rất ít nghiên cứu về việc áp dụng các mô hình SRL cho các hệ tư vấn lai và về cơ bản không có nghiên cứu nào được áp dụng cho các hệ thống quy mô dữ liệu lớn thực sự. Trong bài báo này, chúng tôi đã đề xuất một cách để điều chỉnh các phương pháp tiên tiến trong các phương pháp tiếp cận SRL để xây dựng một hệ thống đề xuất công việc lai thực sự. Hơn nữa, để đáp ứng yêu cầu chung trong các hệ tư vấn, cách tiếp cận của chúng tôi cũng có thể cho phép điều chỉnh sự đánh đổi giữa độ chính xác và thu hồi của hệ thống một cách nguyên tắc. Kết quả thử nghiệm của chúng tôi chứng minh tính hiệu quả của phương pháp đề xuất cũng như hiệu suất được cải thiện về độ chính xác khuyến nghị.

Theo Masateru Tsunoda, Takeshi Kakimoto, Naoki Ohsugi, Akito Monden và Kenichi Matsumoto là tác giả của “Javawock: A Java Class Recommender System Based on Collaborative Filtering” [39], nhiều nền tảng phát triển phần mềm cung cấp một số lượng lớn các thư viện để giúp dễ dàng hơn trong việc xây dựng phần mềm đạt chất lượng cao. Mặt khác, ngày càng khó khăn hơn cho các nhà phát triển trong việc tìm kiếm các thư viện hữu ích trong từng bối cảnh phát triển vì số lượng thư viện được cung cấp trở nên quá lớn ngày nay. Bài viết này đề xuất một hệ thống đề xuất cung cấp các thành phần Java hữu ích (các tệp lớp thư viện) cho nhà phát triển dựa trên bộ lọc cộng tác (CF). Khi một nhà phát triển cung cấp một chương trình Java chưa hoàn thành cho hệ thống, nó sẽ điều tra các tệp lớp thư viện Java được sử dụng trong chương trình đã cho và tìm thấy các chương trình Java tương tự như chương trình đã cho từ kho lưu trữ chương trình. Sau đó, hệ tư vấn các tệp lớp thư viện Java của nhà phát triển đã được sử dụng trong các chương trình tương tự nhưng không được sử dụng trong chương trình của nhà phát triển. Một đánh giá thử nghiệm cho thấy độ chính xác khuyến nghị của hệ thống được đề xuất cao hơn nhiều so với phương pháp truyền thống (không phải CF) trong cả bốn tiêu chí đánh giá (recall, precision, F1 value, and half-life utility).

Theo “Construction of intelligent traffic information recommendation system based on long short-term memory” [40], dịch vụ thông tin giao thông có thể được cải thiện bằng việc cải thiện việc sử dụng đường phố trong các vấn đề về giảm tắc nghẽn giao thông và tai nạn. Trong bài báo này, chúng tôi thiết kế hệ tư vấn thông tin giao thông thông minh dựa trên học sâu – Deep learning. Hệ tư vấn trước tiên xử lý dữ liệu lưu lượng truy cập thông qua công nghệ Internet of Things (IoT), sau đó nó sử dụng mạng học sâu để dự đoán các tham số lưu lượng. Thời lượng tắc nghẽn giao thông và xu hướng tiến hóa khuếch tán không gian được dự đoán tương ứng dựa trên bộ nhớ ngắn hạn (LSTM), đây là một mạng lưới thần kinh sinh ra theo thời gian một đặc tính điển hình của việc học sâu. Theo hiểu biết của chúng tôi, là lần đầu tiên xây dựng hệ tư vấn thông tin giao thông thông minh để cải thiện tính thực tiễn của vấn đề giao thông. Kết quả thử nghiệm cho thấy hệ tư vấn được đề xuất có thể mở rộng khoảng thời gian dự đoán tắc nghẽn giao thông và cải thiện hơn nữa độ tin cậy đồng thời dự đoán từ đó đưa ra cơ sở để hỗ trợ cho người quản lý giao thông và khách du lịch.

Theo Yang Li, Suhang Wang, Quan Pan, Haiyun Peng, Tao Yang và Erik Cambria [41], kịch bản thương mại điện tử đang phát triển rất nhanh, mang đến những thách thức mới cho bộ lọc cộng tác truyền thống. Bởi vì, lượng người dùng và lượng hàng hóa khổng lồ đòi hỏi hệ thống lưu trữ một lượng lớn thông tin và hệ thống đề xuất phải hiệu quả. Do đó, băm để lọc cộng tác đã thu hút sự chú ý ngày càng tăng vì mã nhị phân có thể làm giảm đáng kể yêu cầu lưu trữ và làm cho tính toán tương tự vẫn hiệu quả. Trong bài báo này, chúng tôi nghiên cứu vấn đề mới về các mã băm cộng tác sâu trên xếp hạng vật phẩm của người dùng. Chúng tôi đề xuất một khung học sâu mới cho nó, thông qua các neural networks để học tốt hơn cả việc thể hiện người dùng và vật phẩm. Làm cho các mã này gần với mã nhị phân giúp việc giảm lượng tử hóa được tối thiểu hóa. Ngoài ra, chúng tôi mở rộng khung đề xuất cho các trường hợp ngoài mẫu, tức là, các giao dịch với người dùng mới, mặt hàng mới và xếp hạng mới. Các thí nghiệm mở rộng trên các bộ dữ liệu trong thế giới thực chứng minh tính hiệu quả của khung đề xuất.

Trong bài nghiên cứu “An effective distributed predictive model with Matrix factorization and random forest for Big Data recommendation systems” [42], các hệ tư vấn đã được triển khai rộng rãi để giải quyết thách thức về thông tin áp đảo. Chúng được sử dụng để cho phép người dùng tìm thông tin thú vị từ một khối lượng lớn dữ liệu. Tuy nhiên, trong kỷ nguyên của Dữ liệu lớn, khi dữ liệu trở nên lớn hơn và phức tạp hơn, thuật toán đề xuất chạy trong môi trường truyền thống không thể nhanh và hiệu quả. Nó đòi hỏi tính toán cao để thực hiện nhiệm vụ đào tạo, điều này có thể hạn chế khả năng ứng dụng của nó trong các ứng dụng dữ liệu lớn trong thế giới thực.

Trong bài báo này, chúng tôi đề xuất một giải pháp đề xuất phân tán mới cho dữ liệu lớn. Nó được thiết kế dựa trên Apache Spark để xử lý dữ liệu quy mô lớn, cải thiện chất lượng dự đoán và giải quyết vấn đề thưa thớt dữ liệu. Đặc biệt, nhờ vào một quá trình học tập mới lạ, mô hình có thể tăng tốc đáng kể việc đào tạo phân tán, cũng như cải thiện hiệu suất trong bối cảnh của dữ liệu lớn. Kết quả thử nghiệm trên ba bộ dữ liệu trong thế giới thực chứng minh rằng đề xuất của chúng tôi vượt trội hơn các phương pháp đề xuất hiện có về Lỗi trung bình tuyệt đối (MAE), Lỗi bình phương trung bình (RMSE) và thời gian tính toán.

Theo Liang Ying và LiuBoqi [43], để đáp ứng nhu cầu học tập cá nhân, nền tảng giảng dạy hiện tại sử dụng đến hệ tư vấn, bao gồm khuyến nghị về tài nguyên học tập, khuyến nghị về nhiệm vụ và bài tập, v.v. Các vấn đề như mốc thời gian lỗi thời hay thiếu dữ liệu vẫn tồn tại trong nền tảng khuyến nghị nền tảng giảng dạy. Lấy hệ tư vấn nhiệm vụ của các khóa học máy tính làm ví dụ, kiến thức máy tính cập nhật nhanh chóng. Các kiến thức cũ sẽ hết hạn trong hai đến ba năm. Một trường hợp khác là, khi trang web được đưa vào sử dụng, vì lưu lượng truy cập trang web khá thấp, mốc thời gian hoàn thành nhiệm vụ của các sinh viên là ít. Cả hai trường hợp này có thể gây ra thiếu dữ liệu của hệ thống nhiệm vụ trên nền tảng giảng dạy online. Để giải quyết vấn đề này, bài viết này cố gắng áp dụng học chuyển vào hệ tư vấn giảng dạy online và xác minh tính khả thi của nó thông qua thử nghiệm.

Theo như nghiên cứu “Attention and saliency on the internet: Evidence from an online recommendation system” [44], sử dụng dữ liệu sản phẩm có tần số cao từ một cửa hàng bán lẻ trực tuyến, chúng tôi kiểm tra xem các lựa chọn của người tiêu dùng trên internet có phù hợp với các mô hình hay không. Chúng tôi kiểm tra xem người tiêu dùng có nhiều khả năng mua các sản phẩm nhận khi chúng được khuyến nghị hay không. Chúng tôi thấy mức tăng mạnh 6% và mạnh mẽ trong tổng doanh số của các sản phẩm hiện có sau khi chúng được đề xuất bởi bởi một sản phẩm khác. Chúng tôi cũng xác định rằng tác động lan tỏa đối với các sản phẩm ở xa hơn trong mạng khuyến nghị là rất nhỏ, cho thấy rằng việc quay lại tìm kiếm thông qua mạng khuyến nghị sẽ giảm nhanh chóng. Sử dụng một mô hình cấu trúc, chúng tôi thấy rằng khuyến nghị ảnh hưởng lớn đến sự xem xét của người tiêu dùng, những ảnh hưởng nhỏ hơn đến sự lựa chọn tiếp theo của họ. Các đề nghị trái ngược với mong muốn của khách hàng gây hại cho các sản phẩm đang được bán chạy nhất nhưng các hệ tư vấn cũng làm giảm bớt việc tìm kiếm.

Dựa trên đề tài “Nghiên cứu lớp bài toán luật kết hợp trong lĩnh vực khai phá dữ liệu” [45], Báo cáo giới thiệu về quá trình tiền xử lý để làm sạch dữ liệu trước khi tiến hành khai phá, sau đó sẽ đi vào tiến hành bốn giải thuật, tiếp đến là giới thiệu một số ứng dụng mà khai phá dữ liệu có thể áp dụng trong thực tế, sau đó giới thiệu về giải thuật Apriori và cải tiến của nó.

Theo một bài nghiên cứu về “Hệ tư vấn lọc cộng tác theo người dùng dựa trên độ đo hàm ý thống kê” [46], từ khi ra đời đến nay, hệ tư vấn lọc cộng tác nói chung, hệ tư vấn lọc cộng tác dựa trên người dùng nói riêng đã có một sự phát triển vượt bậc về mặt ứng dụng kỹ thuật, công nghệ cũng như ứng dụng vào thực tế cuộc sống. Đặc biệt, hệ tư vấn được các nhà quản lý sử dụng làm công cụ hỗ trợ hữu hiệu trong nhiều lĩnh vực kinh doanh như Amazon, Netflix và Pandora. Tuy nhiên, các thế hệ hiện tại của hệ tư vấn vẫn chưa đáp ứng đầy đủ các yêu cầu của người sử dụng. Trong bài viết này chúng tôi đề xuất một tiếp cận mới cho hệ tư vấn lọc cộng tác dựa trên người dùng. Hệ tư vấn lọc cộng tác theo người dùng được xây dựng dựa trên độ đo hàm ý thống kê. Trong hệ tư vấn này, chúng tôi xây dựng một độ đo tương đồng dựa trên độ đo chỉ số hàm ý thống kê gọi là độ đo tương đồng hàm ý thống kê để xác định sự tương đồng giữa hai người dùng trong hệ thống. Thông qua thực nghiệm trên hai tập dữ liệu Movielens và MSWeb cho thấy rằng độ đo tương đồng mà chúng tôi đề xuất cho kết quả khá tốt trên hệ tư vấn lọc cộng tác dựa trên người dùng so với các độ đo tương đồng truyền thống như Pearson correlation, Cosine similarity và Jaccard.

Theo một chuyên đề “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” [47], việc tìm kiếm thông tin khoa học để thực hiện các công việc liên quan đến nghiên cứu là nhu cầu thường xuyên, không thể thiếu đối với những người làm nghiên cứu khoa học, đặc biệt là các nghiên cứu viên. Các nghiên cứu viên trẻ thì thiếu kinh nghiệm tìm kiếm và xác định các thông tin hữu ích liên quan. Trong khi, các nghiên cứu viên có kinh nghiệm thì phải đương đầu với quá tải thông tin. Để giúp họ dễ dàng hơn trong việc tiếp cận các thông tin học thuật hữu ích liên quan, hệ khuyến nghị trong lĩnh vực học thuật là giải pháp đang được quan tâm nghiên cứu trong những năm gần đây. Các bài toán khuyến nghị thông tin học thuật phổ biến như: khuyến nghị bài báo, cộng tác, gửi bài. cũng như các cách tiếp cận truyền thống cho hệ khuyến nghị là lọc dựa trên thông tin lý lịch (Demographic Filtering), CB, CF, lai (Hybrid) phải đương đầu với một số khó khăn, thách thức như: dữ liệu lớn, chưa có dữ liệu chuẩn (benchmark) cho đánh giá thực nghiệm, độ chính xác chưa cao, vấn đề khởi động lạnh (cold-start), chưa có phương pháp phù hợp để đánh giá chất lượng khuyến nghị. Xu hướng tiếp cận để phát triển các phương pháp mới cho hệ khuyến nghị đó là: phân tích mạng xã hội, khai thác thông tin ngữ cảnh và các phương pháp lai. Trên thực tế, sở thích và quyết định của con người thường chịu ảnh hưởng bởi những người có quan hệ. Các nghiên cứu viên thường cần lời khuyên từ bạn bè, đồng nghiệp, thầy cô để đưa ra những quyết định quan trọng liên quan đến các công việc nghiên cứu. Do đó, luận án chọn tiếp cận phân tích mạng xã hội (có xem xét yếu tố thời gian) kết hợp một số thông tin khác, nhằm giải quyết những hạn chế của một số phương pháp phổ biến, ứng dụng khuyến nghị thông tin học thuật.

Theo Nguyễn Song Hà [46], hệ tư vấn (recommender system) đã trở thành một trong những lĩnh vực nghiên cứu quan trọng kể từ khi bài báo đầu tiên về CF xuất hiện vào giữa những năm 1990. Hiện nay, sự quan tâm đối với hệ tư vấn đang rất cao vì sự cần thiết của những ứng dụng có thể giúp người dùng xử lý với tình trạng quá tải thông tin & đưa ra những nội dung hoặc lời khuyên phù hợp cho từng cá nhân. Một vài ứng dụng nổi tiếng như: hệ tư vấn sách, CDs của Amazon.com, hệ tư vấn phim của MovieLens… Nhưng so với sách, phim. thì số lượng website bùng nổ mỗi ngày còn lớn hơn rất nhiều. Khóa luận đề xuất phương pháp xây dựng một hệ thống tư vấn website dựa trên việc khai phá query logs của máy tìm kiếm. Các website được tư vấn là kết quả có được dựa trên phân tích những lựa chọn của hàng nghìn người dùng trước đó. Thực nghiệm ban đầu của hệ thống cho kết quả khá tốt.

Đề tài “Thuật toán phân lớp văn bản Web và thực nghiệm trong máy tìm kiếm VietSeek” [48], Do kích thước khổng lồ của dữ liệu Web, việc xây dựng cũng như tích hợp các yếu tố khai phá dữ liệu Web vào công cụ tìm kiếm trên mạng Internet đang thu hút được sự quan tâm rất lớn của rất nhiều nhà nghiên cứu. Khóa luận đề cập tới vấn đề cải tiến chất lượng và tốc độ của máy tìm kiếm bằng việc nghiên cứu bài toán phân lớp trong máy tìm kiếm. Nội dung chính của khóa luận trình bày cấu trúc cũng như mô hình hoạt động của một chức năng đánh chỉ mục trong máy tìm kiếm VietSeek, các kỹ thuật cơ bản và các thuật toán thông dụng liên quan đến quá trình khai phá dữ liệu Web trong máy tìm kiếm, mà cụ thể là bài toán phân lớp trang văn bản Web. Đặc biệt khóa luận tập trung vào giải pháp phân lớp theo phương pháp Bayes thứ nhất. Đi kèm với giải pháp phân lớp Bayes là các đề xuất nhằm giải quyết vấn đề cho các lớp. Khóa luận đã tích hợp thành công các đề xuất này vào máy tìm kiếm VietSeek và thu được kết quả rất khả quan.

Theo Sachin Walunj và Kishor Sadafale trong “An Online Recommendation System for E-commerce Based on Apache Mahout Framework” [51]. Việc chọn một nền tảng là bước quan trọng trong việc phát triển RS cho mục đích nghiên cứu hoặc thương mại. Việc trên có thể được thực hiện theo nhiều cách khác nhau, mà ở đó nền tảng được phát triển từ đầu, một hệ tư vấn có thể được thực hiện bằng cách ký kết (Cá nhân hóa OracleAS), sử dụng thư viện mã được điều chỉnh hoặc một nền tảng có thể được chọn và cập nhập cho phù hợp (LensKit, MymediaLite , Mahout Apache, v.v.). Trong một số trường hợp, sự kết hợp của các phương pháp này là được sử dụng. Đối với các dự án thương mại điện tử, và đặc biệt trong các trang web thương mại điện tử, các thức lý tưởng nhất là tìm một nền tảng mã nguồn mở có nhiều người sử dụng và đóng góp từ đó đáp ứng tất cả hoặc hầu hết các yêu cầu phát triển cơ bản.

Với giải pháp lý tưởng này, một số tùy chỉnh nhỏ cho một hệ thống đã có sẵn và có thể được tiếp cận một cách dễ dàng từ đó đáp ứng các yêu cầu phát triển cụ thể. Nhiều thư viện đã được xây dựng nhằm hỗ trợ sự phát triển của các hệ tư vấn, nhưng chỉ gần đây mới có quy mô lớn hơn, các nền tảng mã nguồn mở trở nên có sẵn. Trong bối cảnh của các nền tảng như vậy, các công cụ đánh giá rất quan trọng nhằm để hỗ trợ, thử nghiệm các kỹ thuật và phương pháp tiếp cận mới được phát triển trên nền tảng. Apache Mahout như một nền tảng cho phép nghiên cứu và đã phải đối mặt với cả hai vấn đề này trong việc sử dụng nó như một phần công việc trong các đề xuất lọc cộng tác.

# CHƯƠNG 2: XÂY DỰNG HỆ TƯ VẤN THEO PHƯƠNG PHÁP LỌC CỘNG TÁC

# 2.1. Dữ liệu và tiền xử lý dữ liệu

# 2.1.1. Chuẩn bị dữ liệu

[Bộ cơ sở dữ liệu MovieLens 100k](https://grouplens.org/datasets/movielens/100k/) được công bố năm 1998 bởi dự án nghiên cứu  [GroupLens](https://grouplens.org/) tại University of Minnesota. Bộ cơ sở dữ liệu này bao gồm 100,000 (100k) ratings (1-5 và giá trị tăng thấp nhất là 1) từ 943 users cho 1682 bộ phim.

Cấu trúc của bộ dữ liệu này sau khi tải về gồm như sau:

* u.data: Chứa toàn bộ các *ratings* của 943 *users* cho 1682 movies. Mỗi user *đánh giá* ít nhất 20 movies. Thông tin về thời gian đánh giá cũng được cho nhưng chúng ta không sử dụng trong bài nghiên cứu này.
* u1.base, u1.test,...,u5.base, u5.test: là các bộ kiểm tra rời rạc, được chia theo tỷ lệ 80/20 trên u.data vào huấn luyện (trainning) và kiểm thử (test).
* ua.base, ua.test, ub.base, ub.test: là hai cách chia toàn bộ dữ liệu ra thành hai tập con; một cho training; và một cho test. Với chính xác 10 *ratings* trên một *user* trong file test.
* Tất cả các file u1,...u5 và ua, ub đều được tạo ra bởi mku.sh.
* mku.sh : Một tập lệnh shell để tạo tất cả các tập dữ liệu u từ u.data.
* u.user: Chứa thông tin về *users*, bao gồm: id, tuổi, giới tính, nghề nghiệp, zipcode (vùng miền), vì những thông tin này cũng có thể ảnh hưởng tới sở thích của các *users*. Tuy nhiên, chúng ta sẽ không sử dụng các thông tin này, trừ thông tin *id* được dùng để xác định các user khác nhau.
* u.genre: Chứa tên của 19 thể loại phim, các thể loại bao gồm: unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War và Western.
* u.item: thông tin về mỗi bộ phim, trong file: Mỗi dòng, thể hiện *id* của phim, tên phim, ngày phát hành, link trên imdb, và các số nhị phân 0, 1 ở phía cuối để chỉ ra bộ phim thuộc các thể loại nào trong 19 thể loại đã cho trong u.genre. Một bộ phim có thể thuộc nhiều thể loại khác nhau và thông tin về thể loại này sẽ được dùng để xây dựng item profiles.

# 2.1.2.Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là một bước rất quan trọng trong việc giải quyết các vấn đề trong *ML.* Hầu hết các dữ liệu được sử dụng trong *ML,* thường cần được xử lý, làm sạch và biến đổi trước khi đưa vào mô hình.

Có rất nhiều kỹ thuật xử lý dữ liệu, ví dụ như: xử lý dữ liệu bị khuyết (missing data), mã hóa các biến nhóm (encoding categorical variables), chuẩn hóa dữ liệu (standardizing data), co giãn dữ liệu (scaling data).

Nhưng trong khóa luận này, sẽ tập trung vào một phương pháp đơn giản tính trung bình để chuẩn hóa ma trận. Phương pháp này giúp việc điền các giá trị còn thiếu trong ma trận khởi tạo ban đầu *UM*, nhưng không ảnh hưởng đến kết quả của mô hình.

Như đã được đề cập ở phần trên, các ma trận được thiết lập trong *RS – CF* ban đầu thường rất thiếu dữ liệu. Và vấn đề đặt ra, khi lập ma trận ta sẽ điền giá trị nào vào đó. Cách tốt nhất đó là:

* Tính trung bình cộng các giá trị *rating* của từng *user*.
* Đối với từng user, thay giá trị ‘?’ bằng giá trị trung bình tương ứng.
* Duyệt trên từng item của user và tiến hành lấy giá trị hiện tại trừ đi cho giá trị trung bình tương ứng.
* Lặp lại trên toàn bộ cơ sở dữ liệu, ta thu được một ma trận dữ liệu đã được chuẩn hóa.

Công thức tổng quát như sau:

(2)

Trong đó là giá trị các phần tử của ma trận

# 2.2. Thuật toán lọc cộng tác dựa trên người dùng

Hình 2.1 : Sơ đồ thuật toán UB-CF

Hình 2.1: Sơ đồ thuật toán UB-CF

Thuật toán lọc cộng tác dựa trên người dùng ( *User-Based Collaborative Filtering* Algorithm *-* *UBCF* Algorithm) là một trong những phương pháp truyền thống dựa trên việc xác định độ tương đồng giữa các *users* với nhau hay còn được xem là xác định sự tương quan (*similarity*) giữa hai *user*. Theo đó, ta sẽ có thể đưa ra các gợi ý đến *users* dựa vào *items* thực tế mà *user* có sở thích tương tự.

Các bước thực hiện:

Đầu vào: Đầu vào của dữ liệu, ở đây có thể là các dữ liệu được đọc ở các tệp hoặc trên cơ sở dữ liệu bất kỳ. Dữ liệu được dùng ở đây là bộ cơ sở dữ liệu movielens.

Đầu ra: Trả ra danh sách các item cao nhất được dự đoán cho người dùng.

Bước 1: Là giai đoạn đầu của quá trình tiền xử lý dữ liệu, để tiến hành lập ma trận dựa trên các dữ liệu từ bộ cơ sở dữ liệu. Chúng ta, sẽ tiến hành khởi tạo một ma trận tiện ích với các phần tử trong ma trận là các đánh giá bởi *user* (cột) lên item (hàng).

Bước 2: Còn gọi là bước chuẩn hóa ma trận, bước này cũng nằm trong quá trình tiền xử lý dữ liệu. Ở bước 1, khi khởi tạo ma trận sẽ tồn tại một vấn đề với các giá trị chưa được *user* đánh giá. Nếu chúng ta gán ‘0’ hoặc ‘2.5’ hay một giá trị nào đó trong thang 5 điểm vào ma trận sẽ làm ảnh hưởng đến kết quả bài toán. Bởi vì, mỗi *user* sẽ là khác nhau trong xu thế đưa ra đánh giá cho *items.* Như vậy, nếu ta thay giá trị trung bình của mỗi *user* sẽ mang tính đồng nhất cho từng bài toán.

Do vậy nên phương pháp chung để chuẩn hóa ma trận là:

* Tính giá trị trung bình cho từng user

(2)

Trong đó là giá trị các phần tử của ma trận

* Chuẩn hóa ma trận

(3)

Tạo thành một ma trận mới

Bước 3: Lập ma trận tính độ tương tự, đây là bước quan trọng nhất trong thuật toán này. Có rất nhiều hàm tính độ tương tự, trong đó hàm *Cosine* được sử dụng rất phổ biến trong việc tính độ tương tự, giống nhau giữa 2 *user*.

(4)

Trong đó là vectơ tương ứng với users i đã được chuẩn hóa tại ma trận .

Bước 4: Dự đoán các giá trị còn thiếu, việc xác định mức độ quan tâm của một *user* lên một item dựa trên phương pháp KNN, các giá trị còn thiếu được xác định dựa trên thông tin về số *k-neighbor users*. Chúng ta cũng chỉ quan tâm đến các users đã đánh giá, tức là các giá trị khác ‘0’ trong ma trận .

Công thức phổ biến chung được dùng để dự đoán là:

(5)

Trong đó:

*N (u, i)* là tập hợp k *users* có *similarity* cao nhất của u mà đã đánh giá item i.

là giá trị *rating* được dự đoáncủa *user u* lên *item i*.

là giá trị *rating* đã được đánh giá của *user uj* lên *item i*trong ma trận

là giá trị độ tương tự của giữa user u với users đã đánh giá item i

Để thực hiện việc dự đoán , thuật toán *UBCF* được thực hiện như sau:

- Tạo một danh sách với các phần tử chứa các vị trí *users* đã đánh giá lên *item i* trong ma trận, theo một thứ tự.

* + Lọc ra các *user* đã đánh giá *item i* trong ma trận .
  + Tiến hành sắp xếp theo thứ tự *users* đã lọc có độ tương tự cao nhất với *user* qua ma trận *.*
* Tạo ra một dãy số chứa n phần tử, với các phần tử là chỉ mục các cột của ma trận :
* Tiến hành dự đoán với chỉ số K: Thông qua , ta duyệt các ma trận .

Để hiểu rõ hơn ta xem đoạn code dưới đây

 - Cuối cùng, sau khi tiến hành dự đoán các giá trị chưa được đánh giá trong ma trận. Ta thu được một ma trận mới ma trận này là một ma trận đầy đủ các giá trị. So sánh ma trận và tiến hành đưa các giá trị vừa được dự đoán từ ma trận F sang R, rồi cộng với số trung bình mà tính được ở bước 2.

- Sau khi thu được ma trận mới, ta tiến hành đưa ra các đề xuất với những item được dự đoán có độ đánh giá cao nhất của user mà ta xác định.

# 2.3. Phương pháp đánh giá

# 2.3.1. Phép đo Precision – Recall

Precision-Recall lại là một thước đo hữu ích của sự thành công của dự đoán khi các lớp rất mất cân bằng. Trong truy xuất thông tin, độ chính xác là thước đo mức độ phù hợp của kết quả, trong khi thu hồi là thước đo xem có bao nhiêu kết quả thực sự có liên quan được trả về.

Đường cong precision-recall cho thấy sự đánh đổi giữa độ chính xác và thu hồi cho các ngưỡng khác nhau. Vùng cao dưới đường cong biểu thị cả mức thu hồi cao và độ chính xác cao, trong đó độ chính xác cao liên quan đến tỷ lệ dương tính giả thấp và thu hồi cao liên quan đến tỷ lệ âm tính giả thấp. Điểm cao cho cả hai cho thấy rằng bộ phân loại đang trả về kết quả chính xác (độ chính xác cao), cũng như trả về phần lớn tất cả các kết quả tích cực (thu hồi cao).

Một hệ thống có độ thu hồi cao nhưng độ chính xác thấp trả về nhiều kết quả, nhưng hầu hết các nhãn dự đoán của nó không chính xác khi so sánh với nhãn đào tạo. Một hệ thống có độ chính xác cao nhưng độ thu hồi thấp thì ngược lại, trả về rất ít kết quả, nhưng hầu hết các nhãn dự đoán của nó là chính xác khi so sánh với nhãn đào tạo. Một hệ thống lý tưởng với độ chính xác cao và thu hồi cao sẽ trả về nhiều kết quả, với tất cả các kết quả được dán nhãn chính xác.

*Độ chính xác (P)* được định nghĩa là số lượng dương thực sự () so với số lượng dương thực sự cộng với số lượng dương tính giả ()

(6)

*Recall ()* được định nghĩa là số lượng dương thực sự () so với số lượng dương thực sự cộng với số lượng âm tính giả ().

(7)

Các đại lượng này cũng liên quan đến điểm (), được định nghĩa là giá trị trung bình hài hòa của độ chính xác và thu hồi.

(8)

Lưu ý rằng độ chính xác có thể không giảm khi thu hồi. Độ chính xác () cho thấy việc hạ thấp ngưỡng của bộ phân loại có thể làm tăng mẫu số, bằng cách tăng số lượng kết quả trả về. Nếu ngưỡng trước đó được đặt quá cao, tất cả các kết quả mới có thể là dương thực sự, điều này sẽ làm tăng độ chính xác. Nếu ngưỡng trước đó là về đúng hoặc quá thấp, việc hạ thấp hơn nữa ngưỡng sẽ đưa ra kết quả dương tính giả, giảm độ chính xác.

# 2.3.2. Phép đo F1

Trong thống kê phân tích phân loại nhị phân, các  điểm (còn *F-score* hoặc *F-measure*) là thước đo chính xác của thử nghiệm.

Công thức viết như sau:

(9)

-score có giá trị nằm trong nửa khoảng (0,1]. F1 càng cao, bộ phân lớp càng tốt. Khi cả recall và precision đều bằng 1 (tốt nhất có thể), =1. Khi cả recall và precision đều thấp, ví dụ bằng 0.1, = 0.1. Dưới đây là một vài ví dụ về

**Bảng 3: Bảng ví dụ đánh giá giữa các phương pháp**

| **Precision** | **recall** | **F1** |
| --- | --- | --- |
| 1 | 1 | 1 |
| 0.1 | 0.1 | 0.1 |
| 0.5 | 0.5 | 0.5 |
| 1 | 0.1 | 0.182 |
| 0.3 | 0.8 | 0.36 |

*(Nguồn Internet)*

Như vậy, một bộ phân lớp với precision = recall = 0.5 tốt hơn một bộ phân lớp khác với precision = 0.3, recall = 0.8 theo cách đo này.

chính là một trường hợp đặc biệt của Fβ khi β=1. Khi β>1, recall được coi trọng hơn precision, khi β<1, precision được coi trọng hơn. Hai đại lượng β thường được sử dụng là β=2 và β=0.5.

# 2.3.3. Phép đo MSE

Phép đo MSE [50] – phép đo sai số toàn phương trung bình (Mean squared error) là một phép ước lượng trung bình của bình phương các sai số. Tức là sự khác biệt, giữa các ước lượng so với những gì được quan sát. MSE là một hàm rủi ro, tương ứng với các giá trị dự đoán, kỳ vọng của sự mất mát sai số bình phương hoặc mất mát bậc hai.

MSE là thước đo chất lượng của một công cụ ước tính, nó luôn không âm và các giá trị gần bằng 0 thì tốt hơn. Định nghĩa của MSE khác nhau tùy theo việc người ta mô tả một người dự đoán hay người ước tính.

Công thức của MSE được phát biểu như sau:

*(10)*

Trong đó, nếu là một vectơ của n trị dự báo và là vectơ các giá trị quan sát được trong thực tế.

# 2.3.4. Phép đo RMSE

Phép đo RMSE ( *Root mean square error )*  độ lệch trung bình bình phương là một phương pháp thường được sử dụng trong những khác biệt giữa các giá trị mẫu. Được tiến hành dự đoán bởi một mô hình hay một ước lượng và các giá trị quan sát được.

RMSE luôn không âm và giá trị 0 (gần như không bao giờ đạt được trong thực tế) sẽ cho thấy sự phù hợp hoàn hảo với dữ liệu. Nói chung, RMSE thấp tốt hơn RMSE cao hơn. Tuy nhiên, so sánh giữa các loại dữ liệu khác nhau sẽ không hợp lệ vì số đo phụ thuộc vào quy mô của các số được sử dụng.

RMSE là căn bậc hai của MSE. Ảnh hưởng của từng lỗi trên RMSE tỷ lệ thuận với kích thước của lỗi bình phương. Ta có thể tính RMSE như sau:

*(11)*

Phương pháp RMSE và MSE sẽ là 2 phép đo được sử dụng trong khóa luận này. Bởi như chúng ta đã biết các phép đo precision – recall và F1 được tính toán trên ma trận nhị phân. Cho nên để thuận tiện, khóa luận sẽ sử dụng phép đo RMSE và MSE để ước lượng mô hình.

# CHƯƠNG 3: THÍ NGHIỆM VÀ THẢO LUẬN

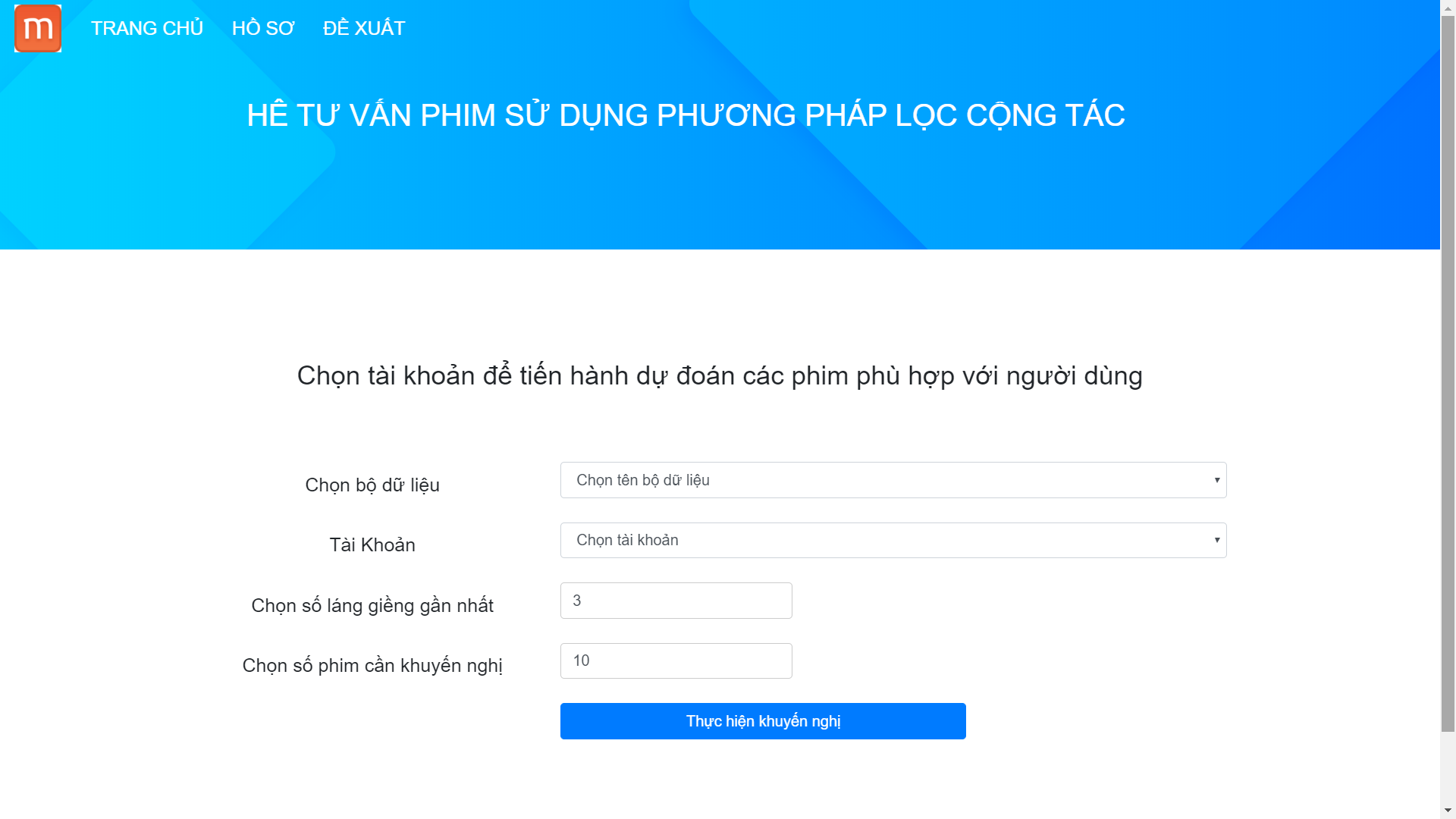
# 3.1. Thiết lập thí nghiệm

* Thí nghiệm được thực hiện trên bộ cơ sở dữ liệu movielens 100k, gồm các bộ dữ liệu ua, ub, u1, u2, u3, u4 và u5. Thí nghiệm thực hiện trên các bộ dữ liệu này để đảm bảo tính ngẫu nhiên và khách quan của thuật toán trên bộ dữ liệu movielens 100k.
* Quá trình thiết lập phần mềm được sử dụng trên nền tảng Java với môi trường xây dựng trên nền tảng Spring Tool suite 4
* Quá trình thiết lập phần mềm trên nền tảng Java SE Development Kit 13, được thiết lập trên nền tảng Win 10 Pro Education
* Giao diện phần mềm**:**

Giao diện ban đầu của phần mềm khi khởi động, chọn “Click để truy cập vào hệ thống”



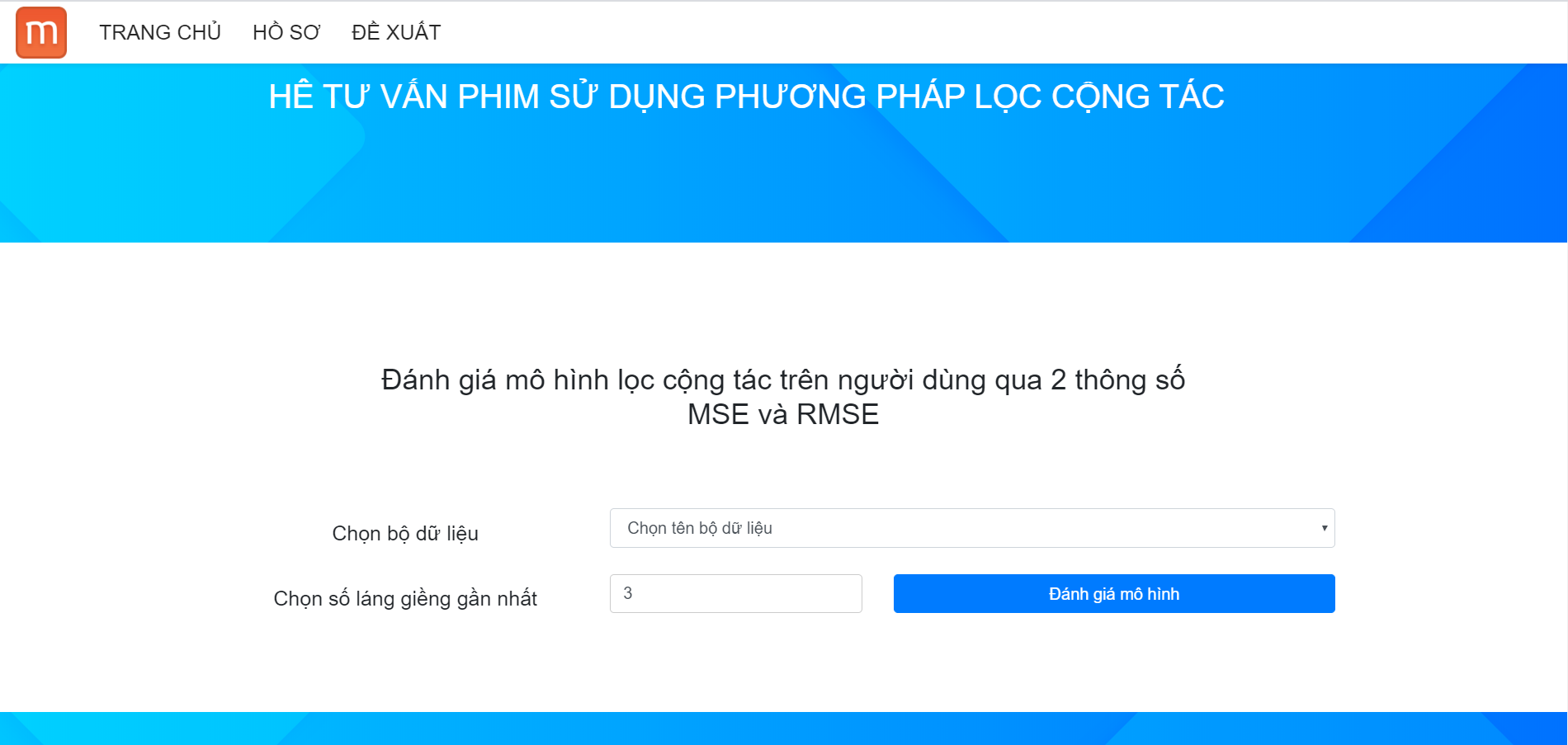
Hình 3.1: Giao diện ban đầu của phần mềm



Hình 3.2: Giao diện phần mềm dự đoán các bộ phim người dùng có thể thích

Ở giao diện này, có 4 phần sau:

* Tiến hành chọn bộ dữ liệu muốn thử nghiệm.
* Chọn tài khoản muốn hệ thống dự đoán các bộ phim có thể thích.
* Chọn số làng giếng gần nhất, với số láng giềng gần nhất càng cao thì kết quả càng chính xác.
* Chọn số phim muốn dự đoán cho tài khoản được chọn.

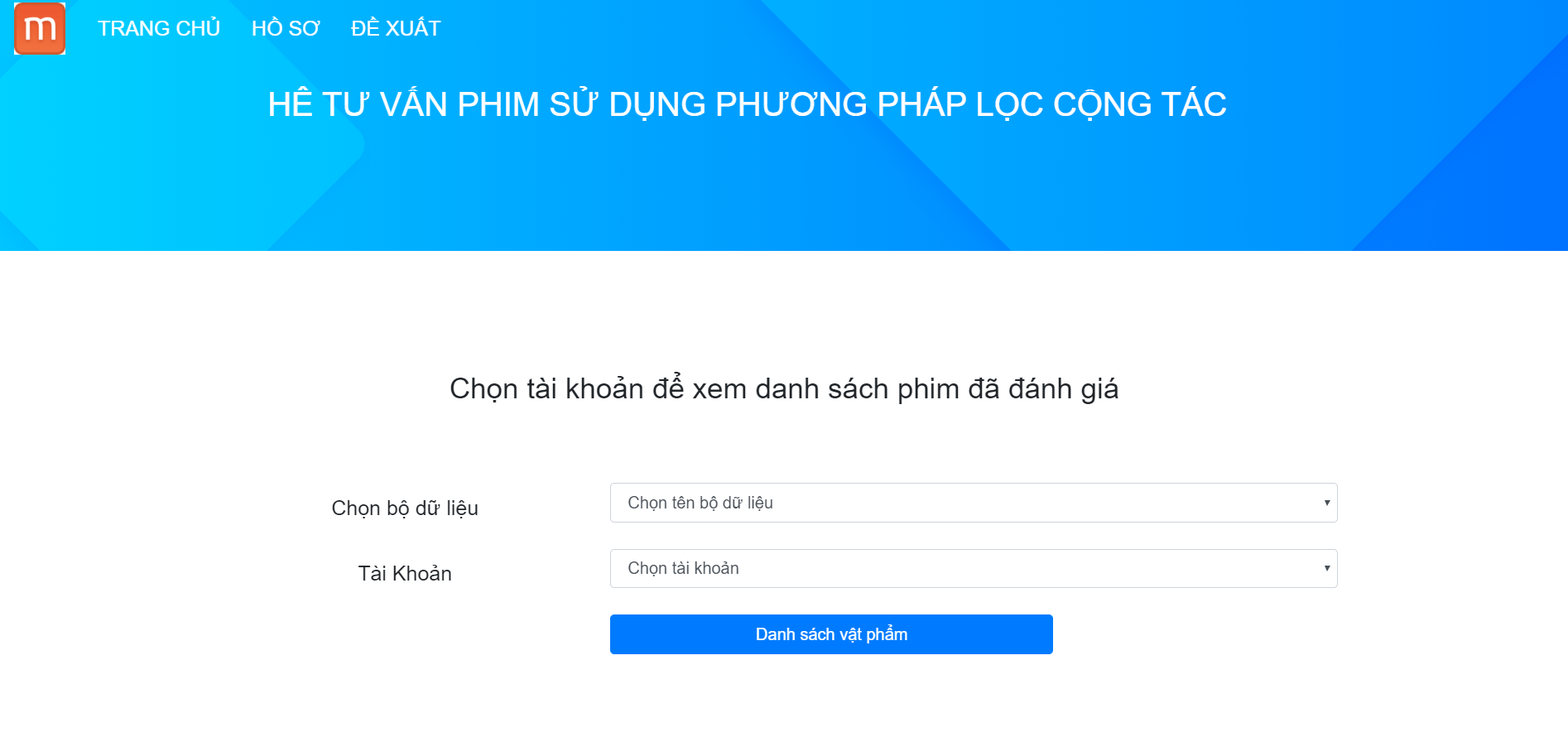


Hình 3.3: Giao diện trang chủ phần mềm để đánh giá mô hình

Ở giao diện này, có 2 phần sau:

* Tiến hành chọn bộ dữ liệu muốn thử nghiệm.
* Chọn số làng giếng gần nhất, với số láng giềng gần nhất càng cao thì kết quả càng chính xác.

Nhấn nút “Đánh giá mô hình” để tiến hành đánh giá mô hình, kết quả sẽ được hiển thị ở phần dưới.



Hình 3.4: Giao diện phần mềm hiển thị danh sách các phim đã được đánh giá

Ở giao diện này, có 2 phần sau:

* Tiến hành chọn bộ dữ liệu muốn thử nghiệm.
* Chọn tài khoản muốn hệ thống đưa ra các bộ phim người dùng đã đánh giá trong bộ cơ sở dữ liệu đã chọn.

Nhấn nút “Danh sách vật phẩm” để tiến hành, việc này giúp ta dễ dàng xem được người dùng đã đánh giá các vật phẩm nào.

# 3.2. Kết quả và thảo luận

# 3.2.1. Bảng phân tích, thống kê kết quả thí nghiệm

**Bảng 4: Kết quả chạy mô hình với bộ dữ liệu ml-100k**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UBCF | Movielens 100K | | | | | | |
| u1.test | u2.test | u3.test | u4.test | u5.test | ua.test | ub.test |
| MSE | 2.11 | 2.10 | 2.12 | 2.07 | 2.098 | 2.14 | 2.11 |
| RMSE | 1.45 | 1.45 | 1.46 | 1.44 | 1.448 | 1.46 | 1.45 |

Nhận xét: Dựa vào kết quả độ sai lệch trong bảng 4, kết quả sai lệch giữa các bộ dữ liệu không quá lớn . Trong đó bộ dữ liệu ua có kết quả sai lệch lớn nhất từ dự đoán so với thực tế, ngoài ra bộ dữ liệu u4 có kết quả nhỏ nhất.

Từ kết quả của bảng 4, ta có biểu đồ của các mô hình trên các tập dữ liệu sau.

# 3.2.2. Biểu đồ

Hình 3.5: Biểu đồ đường của các tập dữ liệu với phương pháp MSE

Hình 3.6: Biểu đồ đường của các tập dữ liệu trên phương pháp RMSE

Theo quan sát, hai biểu đồ ở trên đều thể hiện những ý nghĩa khá tương đồng nhau. Cụ thể là với bộ cơ sở dữ liệu u4 đạt giá trí thấp nhấp và giá trị ua đạt giá trị cao nhất, điều này mang ý nghĩa gì sẽ được trình bày cụ thể hơn ở phần thảo luận phía dưới.

# 3.2.3. Thảo luận

Với mô hình lọc cộng tác dựa trên người dùng – *UBCF* thì ta thấy đường biểu diễn từng hình 3.5 và 3.6 của các tập dữ liệu u1.test; u2.test; u3.test; u4.test; u5.test; ua.test; và ub.test.

Theo quan sát, kết quả MSE và RMSE cho ta biết sai số dự đoán của thuật toán trên các bộ dữ liệu. Kết quả được so sánh trên biểu đồ cho thấy thuật toán được xây dựng không có sai lệch lớn (biên độ lệch không quá 0,1).

Điều này có thể giải thích do sự khác nhau cơ bản giữa các bộ dữ liệu được dùng ở trên, được phân chia từ bộ dữ liệu mẫu 100k một cách ngẫu nhiên. Do các dữ liệu các bộ dữ liệu u1.test; u2.test; u3.test; u4.test; u5.test; ua.test; và ub.test được phân chia một cách ngẫu nhiên từ bộ dữ liệu mẫu u.data nên ta có thể xác định thuật toán đã xây dựng đảm được bảo độ chính xác.

Nhưng nhìn chung, kết quả dự đoán vẫn chưa đạt được độ chính xác tuyệt đối mà vẫn còn có sự sai lệch. Sự sai lệch này chứng tỏ quá trình xây dựng thuật toán dưới dạng mã nguồn code theo sơ đồ ở *Hình 2.1* vẫn còn lỗi, khiến cho kết quả bài toán bị sai lệch đi. Và đây cũng là vấn đề, mà khóa luận này cần nghiên cứu và cải thiện thêm, qua đó đưa ra kết quả chính xác hơn cho bài toán.

# KẾT LUẬN

Với sự phát triển bùng nổ của công nghệ thông tin hiện tai, các ứng dụng về *RS* ngày càng được phát triển và đánh giá cao. Nắm bắt được xu thế của thời đại, tôi đã bắt đầu nghiên cứu đề tài “Xây dựng hệ tư vấn sử dụng phương pháp lọc cộng tác” và đã được một số kết quả như sau:

- Nắm được lý thuyết cơ bản về xây dựng hệ tư vấn.

- Xây dựng được thuật toán hệ tư vấn theo phương pháp lọc cộng tác dựa trên người dùng.

Ngoài những kết quả đạt được thì bên cạnh đó còn một số thiếu sót:

- Phương pháp xây dựng mô hình *UBCF* để đưa ra các đề xuất chỉ dừng ở mức nghiên cứu chưa có tính sáng tạo để nâng cao hiệu quả của bài toán hệ tư vấn.

- Chưa xây dựng một hệ thống hoàn chỉnh áp dụng phương pháp lọc cộng tác đề xuất phim mà chỉ dừng ở ngang mức kiểm nghiệm độ chính xác của thuật toán.

- Vấn đề về khởi động lạnh chưa được giải quyết.

Nhìn chung, việc nghiên cứu lĩnh vực này hoàn toàn khả thi và thực tiễn, phù hợp với nhu cầu hiện nay. Do đó trong thời gian tới, tôi sẽ tiếp tục nghiên cứu thêm các phương pháp khác của hệ tư vấn để cải thiện kết quả độ chính xác và có thể xác định được chính xác hơn, cũng như đưa ra các khuyến nghị phù hợp hơn với từng người dùng trong hệ thống.

# DANH MỤC TÀI LIỆU THAM KHẢO

1. (2019). Recommender system. Wikipedia, <https://en.wikipedia.org/w/index.php?title=Recommender\_system&oldid=927234992>, accessed: 07/12/2019.

2. Goldberg D., David Nichols, Brian M. Oki và cộng sự. (1992). Using collaborative filtering to weave an information Tapestry. Association for Computing Machinery.

3. (2019). Collaborative filtering. Wikipedia, <https://en.wikipedia.org/w/index.php?title=Collaborative\_filtering&oldid=923933501>, accessed: 06/12/2019.

4. Gomez-Uribe C.A. và Hunt N. (2015). The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans Manag Inf Syst*, **6**(**4**), 1–19.

5. (2019). Luật số lớn. Wikipedia tiếng Việt, <https://vi.wikipedia.org/w/index.php?title=Lu%E1%BA%ADt\_s%E1%BB%91\_l%E1%BB%9Bn&oldid=50200374>, accessed: 22/11/2019.

6. (2019). Tâm lý học đám đông. Wikipedia tiếng Việt, <https://vi.wikipedia.org/w/index.php?title=T%C3%A2m\_l%C3%BD\_h%E1%BB%8Dc\_%C4%91%C3%A1m\_%C4%91%C3%B4ng&oldid=50237198>, accessed: 22/11/2019.

7. (2019). Big data. Wikipedia, <https://en.wikipedia.org/w/index.php?title=Big\_data&oldid=923021290>, accessed: 27/10/2019.

8. (2019). Bolero Việt Nam. Wikipedia tiếng Việt, <https://vi.wikipedia.org/w/index.php?title=Bolero\_Vi%E1%BB%87t\_Nam&oldid=56303669>, accessed: 05/12/2019.

9. (2019). Nhạc thiếu nhi. Wikipedia tiếng Việt, <https://vi.wikipedia.org/w/index.php?title=Nh%E1%BA%A1c\_thi%E1%BA%BFu\_nhi&oldid=55175651>, accessed: 05/12/2019.

10. (2019). Vector. Wikipedia, <https://en.wikipedia.org/w/index.php?title=Vector&oldid=927424924>, accessed: 07/12/2019.

11. (2019). Euclidean distance. Wikipedia, <https://en.wikipedia.org/w/index.php?title=Euclidean\_distance&oldid=914121508>, accessed: 07/12/2019.

12. Manhattan distance - Wiktionary. <https://en.wiktionary.org/wiki/Manhattan\_distance>, accessed: 07/12/2019.

13. Vu T. (2017). Bài 23: Content-based Recommendation Systems. Tiep Vu’s blog, <https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/>, accessed: 04/10/2019.

14. Vu T. (2017). Bài 24: Neighborhood-Based Collaborative Filtering. Tiep Vu’s blog, <https://machinelearningcoban.com/2017/05/24/collaborativefiltering/>, accessed: 27/09/2019.

15. Gregory D. Linden;, Jennifer A. Jacobi, và Eric A. Benson Collaborative recommendations using item-to-item similarity mappings. <https://patents.google.com/patent/US6266649B1/en>.

16. Linden G., Smith B., và York J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput*, **7**(**1**), 76–80.

17. Wang D., Liang Y., Xu D. và cộng sự. (2018). A content-based recommender system for computer science publications. *Knowl-Based Syst*, **157**, 1–9.

18. Adomavicius G. và Tuzhilin A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng*, **17**(**6**), 734–749.

19. Ekstrand M.D. (2011). Collaborative Filtering Recommender Systems. *Found Trends® Human–Computer Interact*, **4**(**2**), 81–173.

20. Lam X.N., Vu T., Le T.D. và cộng sự. (2008). Addressing cold-start problem in recommendation systems. *Proceedings of the 2nd international conference on Ubiquitous information management and communication - ICUIMC ’08*, Suwon, Korea, ACM Press, 208.

21. Francesco Ricci L.R. and B.S. và Bracha Shapira *Introduction to Recommender Systems Handbook*, .

22. Su X. và Khoshgoftaar T.M. (2009). A Survey of Collaborative Filtering Techniques. *Adv Artif Intell*, **2009**, 1–19.

23. Buhmann M.D., Melville P., Sindhwani V. và cộng sự. (2011). Recommender Systems. *Encyclopedia of Machine Learning*. Springer US, Boston, MA, 829–838.

24. Elahi M., Ricci F., và Rubens N. (2016). A survey of active learning in collaborative filtering recommender systems. *Comput Sci Rev*, **20**, 29–50.

25. Wen - Recommendation System Based on Collaborative Filte.pdf. <http://www.zheng-wen.com/WenRecommendation.pdf>, accessed: 12/10/2019.

26. Hofmann T. Latent Semantic Models for Collaborative filtering. *ACM Trans Inf Syst*, 89–115.

27. Koren Y., Bell R., và Volinsky C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, **42**(**8**), 30–37.

28. Sarwar B., Karypis G., Konstan J. và cộng sự. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the tenth international conference on World Wide Web - WWW ’01*, Hong Kong, Hong Kong, ACM Press, 285–295.

29. Bottou L. (2012). Stochastic Gradient Descent Tricks. *Neural Networks: Tricks of the Trade*. Springer Berlin Heidelberg, Berlin, Heidelberg, 421–436.

30. Dr. Jerry Eriksson và Prof. Dr. Patrik Eklund Matrix Factorization Methods for Recommender Systems Shameem Ahamed Puthiya Parambath. <http://www.diva-portal.org/smash/get/diva2:633561/FULLTEXT01.pdf>.

31. Badriyah T. và Prasetyaningrum I. (2015). BUILDING A RECOMMENDATION SYSTEM FOR ONLINE SHOPPING BASED ON ITEM-BASED COLLABORATIVE FILTERING. 10.

32. Herlocker J.L., Konstan J.A., Terveen L.G. và cộng sự. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans Inf Syst*, **22**(**1**), 5–53.

33. Nikzad–Khasmakhi N., Balafar M.A., và Reza Feizi–Derakhshi M. (2019). The state-of-the-art in expert recommendation systems. *Eng Appl Artif Intell*, **82**, 126–147.

34. Milovančević N.S. và Gračanac A. (2019). Time and ontology for resource recommendation system. *Phys Stat Mech Its Appl*, **525**, 752–760.

35. Zihayat M., Ayanso A., Zhao X. và cộng sự. (2019). A utility-based news recommendation system. *Decis Support Syst*, **117**, 14–27.

36. Ni L., Lin H., Zhang M. và cộng sự. (2018). Hybrid Filtrations Recommendation System based on Privacy Preserving in Edge Computing. *Procedia Comput Sci*, **129**, 407–409.

37. Shakhovska N., Fedushko S., ml M.G. và cộng sự. (2019). Development of Mobile System for Medical Recommendations. *Procedia Comput Sci*, **155**, 43–50.

38. Yang S., Korayem M., AlJadda K. và cộng sự. (2017). Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. *Knowl-Based Syst*, **136**, 37–45.

39. (PDF) Javawock: A Java Class Recommender System Based on Collaborative Filtering. ResearchGate, <https://www.researchgate.net/publication/221390198\_Javawock\_A\_Java\_Class\_Recommender\_System\_Based\_on\_Collaborative\_Filtering>, accessed: 18/10/2019.

40. Kong F., Li J., và Lv Z. (2018). Construction of intelligent traffic information recommendation system based on long short-term memory. *J Comput Sci*, **26**, 78–86.

41. Yang Li, Suhang Wang, Quan Pan và cộng sự. Learning binary codes with neural collaborative filtering for efficient recommendation systems. <https://doi.org/10.1016/j.knosys.2019.02.012>.

42. Badr Ait Hammou, Ayoub Ait Lahcen, và Salma Mouline An effective distributed predictive model with Matrix factorization and random forest for Big Data recommendation systems. .

43. Liang Ying và LiuBoqin Application of Transfer Learning in Task Recommendation System. <sci-hub.tw/10.1016/j.proeng.2017.01.178>.

44. Helmers C., Krishnan P., và Patnam M. (2019). Attention and saliency on the internet: Evidence from an online recommendation system. *J Econ Behav Organ*, **161**, 216–242.

45. Phạm Nhật Trí, Nguyễn Bình Long, và Phạm Nguyễn Đức Dương Nghiên cứu lớp bài toán luật kết hợp trong lãnh vực khai phá dữ liệu. Nghiên cứu cơ sở lý thuyết họ giải thuật Apriori. .

46. Phan Quốc Nghĩa, Nguyễn Minh Kỳ, Đặng Hoài Phƣơng và cộng sự. HỆ TƯ VẤN LỌC CỘNG TÁC THEO NGƯỜI DÙNG DỰA TRÊN ĐỘ ĐO HÀM Ý THỐNG KÊ. .

47. HUỲNH NGỌC TÍN 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. .

48. K45\_Dang\_Thanh\_Hai\_Thesis.pdf. <http://uet.vnu.edu.vn/~thuyhq/Student\_Thesis/K45\_Dang\_Thanh\_Hai\_Thesis.pdf>, accessed: 12/10/2019.

49. Walunj S.G. và Sadafale K. (2013). An online recommendation system for e-commerce based on apache mahout framework. *Proceedings of the 2013 annual conference on Computers and people research - SIGMIS-CPR ’13*, Cincinnati, Ohio, USA, ACM Press, 153.

50. (2019). Mean squared error. Wikipedia, <https://en.wikipedia.org/w/index.php?title=Mean\_squared\_error&oldid=929853135>, accessed: 15/12/2019.

