HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

_____ o0o _____

BÀI TẬP LỚN NHẬP MÔN TRÍ TUỆ NHÂN TẠO

Tên đề tài: Sử dụng phương pháp lọc cộng tác cho bài toán gợi ý sản phẩm

LỚP : N07
Số thứ tự nhóm: 01

Nguyễn Khánh Linh	**MSSV: B21DCCN484**
Nguyễn Minh Giang	**MSSV: B21DCCN304**
Nguyễn Văn Hùng	**MSSV: B21DCCN417**
Trần Công Hiếu	**MSSV: B21DCCN369**
Phạm Quỳnh Chi	**MSSV: B21DCCN177**

Giảng viên hướng dẫn: Ths. Vũ Hoài Thư

HÀ NỘI, 04/2024

LỜI CẢM ƠN

Trước hết, chúng em xin được tỏ lòng biết ơn và gửi lời cám ơn chân thành đến Giảng viên Vũ Hoài Thư, người đã dành thời gian và công sức để hướng dẫn và giảng dạy cho chúng em trong suốt quá trình học tập môn Nhập môn trí tuệ nhân tạo. Sự tận tình và sự hỗ trợ của cô đã giúp chúng em hiểu sâu hơn về môn học và truyền đạt cho chúng em những kiến thức bổ ích trong suốt học kỳ vừa qua.

Chúng em cũng muốn cảm ơn cô vì đã tạo cơ hội cho nhóm 1 chúng em để thực hiện bài tập lớn này. Qua việc thực hành và áp dụng những kiến thức được học vào thực tế, chúng em đã có cơ hội trải nghiệm và học hỏi nhiều điều mới mẻ và bổ ích.

Tuy nhiên, sau tất cả những nỗ lực và cố gắng, chúng em nhận thấy rằng bài báo cáo của nhóm chúng em vẫn còn nhiều thiếu sót và hạn chế. Với vốn kiến thức và kinh nghiệm ít ỏi, chúng em hiểu rằng còn phải nỗ lực và rèn luyện thêm nhiều để có thể hoàn thiện hơn trong tương lai.

Vì vậy, chúng em xin kính mong sự thông cảm và góp ý từ phía cô để chúng em có thể rút kinh nghiệm và cải thiện bản thân mình mỗi ngày. Những đóng góp và ý kiến phản hồi của cô sẽ là động lực quan trọng giúp chúng em tiến xa hơn trên con đường học tập và nghiên cứu.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Tiếng Anh	Tiếng Việt	Giải thích
User	Người dùng, người sử dụng	Chỉ những người dùng hệ thống để tìm kiếm lựa chọn sản phẩm
Item	Sản phẩm, mục	Chỉ những sản phẩm trên hệ thống như: sản phẩm, phim, ảnh, bản nhạc, trang web, đoạn văn bản,...
Rating	Đánh giá	Chỉ mức độ thích của một người dùng với sản phẩm.
CF	Lọc cộng tác	CF có tên là Neighborhood-based Collaborative Filtering (NBCF). Khi chỉ nói CF, chúng ta sẽ ngầm hiểu rằng phương pháp được sử dụng là Neighborhood-based.
Adjusted Cosine Similarity (ACS)	Độ tương tự co-sine điều chỉnh	Phương pháp tính toán độ tương đồng giữa người dùng và vật phẩm dựa trên sự khác biệt về góc vectơ giữa các đánh giá.
Root Mean Square Estimation (RMSE)	Lỗi trung bình bình phương	Phương pháp ước tính sai số trung bình giữa giá trị dự đoán và giá trị thực tế.

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Sự phát triển mạnh mẽ của thương mại điện tử (E-Commerce) là một trong những yếu tố chính mang lại nhiều lợi ích to lớn cho nền kinh tế toàn cầu. Thương mại điện tử đã tạo ra nhiều hình thức kinh doanh mới, trong đó có mua bán trực tuyến. Phương thức mới này cho phép người tiêu dùng tiếp cận sản phẩm dễ dàng và nhanh chóng hơn nhiều so với các hình thức mua bán truyền thống.

Hiện nay, hệ thống bán hàng trực tuyến tạo nhiều điều kiện thuận lợi cho phép người mua tiếp cận nhiều sản phẩm cùng một lúc. Tuy nhiên, các website thương mại luôn mong muốn tăng lượng khách hàng. Nếu muốn thu hút nhiều khách hàng hơn, người bán cần đa dạng hóa chủng loại sản phẩm để đáp ứng nhu cầu mua sắm của nhiều loại khách hàng khác nhau. Vì vậy, số lượng sản phẩm, chủng loại sản phẩm hiển thị trên một website ngày càng tăng, hạn chế khả năng giao tiếp và lựa chọn sản phẩm của khách hàng. Để tìm được sản phẩm mình muốn, người mua phải tìm kiếm qua nhiều liên kết và lọc qua rất nhiều thông tin. Họ khó có thể chọn ra cho mình một sản phẩm ưng ý nhất.

Trong kinh doanh, việc đưa ra lời khuyên, sự trợ giúp để khách hàng có thể tìm và mua được sản phẩm phù hợp là rất quan trọng. . Trong các phương thức bán hàng truyền thống, lời khuyên kiểu này của nhân viên bán hàng mang lại rất nhiều lợi nhuận cho doanh nghiệp. Vì vậy, để mua sắm trực tuyến thực sự phát triển, việc tìm thêm "người trợ giúp" bên cạnh những lợi thế vốn có là rất cần thiết. Hệ thống tư vấn được thiết kế và phát triển nhằm đáp ứng các yêu cầu trên. Một hệ thống tư vấn tốt là có thể đóng vai trò như một người hỗ trợ khách hàng bằng cách xác định mục tiêu và nhu cầu của khách hàng, hệ thống có thể đưa ra những khuyến nghị giúp khách hàng lựa chọn được sản phẩm mình thích. Hệ thống này giống như một người bán hàng có thể thu thập thông tin về sở thích của khách hàng và sau đó tìm kiếm trong kho vô tận của nó những sản phẩm phù hợp nhất với những sở thích đó. Cốt lõi của hệ thống tư vấn này là quá trình giúp khách hàng đưa ra quyết định mua sắm đúng đắn nhất. Qua đó hiệu suất của việc mua bán hàng trực tuyến được tăng cao một cách đáng kể.

Hệ thống gợi ý được phát triển bằng nhiều phương pháp khác nhau. Phương pháp lọc cộng tác (Collaborative filtering) là một trong những phương pháp được nghiên cứu chuyên sâu và áp dụng khá thành công trong nhiều hệ thống gợi ý. Về bản chất, lọc cộng tác là một hình thức tư vấn tự động bằng cách dựa trên sự tương tự giữa những người dùng hoặc giữa những sản phẩm trong hệ thống và đưa ra dự

đoán sự quan tâm của người dùng tới những sản phẩm, hoặc đưa ra gợi ý một sản phẩm mới cho người dùng. Kỹ thuật này được áp dụng thành công trong nhiều ứng dụng và website như website mua sắm trực tuyến Amazon (www.amazon.com)m, cổng video clip YouTube (www.youtube.com),...

Với những lý do trên, chúng em đã lựa chọn đề tài "Sử dụng phương pháp lọc cộng tác cho bài toán gợi ý sản phẩm" làm đề tài nghiên cứu cho bài tập lớn môn học Nhập môn trí tuệ nhân tạo.

1.2 Mục tiêu và phạm vi đề tài

1.2.1 Mục tiêu nghiên cứu

Mục tiêu của báo cáo này là so sánh các phương pháp Lọc cộng tác, chủ yếu là Lọc cộng tác dựa trên người dùng và Lọc cộng tác dựa trên item, trên các tập dữ liệu được cung cấp bởi cơ sở dữ liệu MovieLens. Điều này nhằm mục đích xem xét hiệu suất, đồng bằng và khác biệt của chúng. Báo cáo nhằm mục đích điều tra các vấn đề sau:

- Dựa trên độ rải rác của cơ sở dữ liệu, kích thước của dữ liệu huấn luyện và kiểm tra, trong những tình huống nào các phương pháp Lọc cộng tác khác nhau vượt trội hơn so với nhau?
- Các điểm đồng nhất và khác biệt chính giữa các thuật toán khác nhau là gì?

1.2.2 Phạm vi nghiên cứu

Phạm vi của nghiên cứu này bao gồm việc sử dụng hai tập dữ liệu để tạo ra các tập con dữ liệu để huấn luyện chương trình. Mục tiêu không phải là đạt hiệu suất cao, mà là so sánh các phương pháp khác nhau với nhau. Lọc cộng tác dựa trên người dùng được dự kiến sẽ vượt trội khi xử lý lượng dữ liệu lớn, trong khi lọc cộng tác dựa trên mục được dự kiến sẽ hoạt động tốt hơn trên các tập dữ liệu nhỏ hơn.

1.3 Định hướng giải pháp

1.3.1 Phương pháp nghiên cứu

Thu thập, tìm hiểu, phân tích các tài liệu có liên quan đến đề tài.

Phân tích, thiết kế, triển khai xây dựng hệ thống gợi ý sử dụng phương pháp lọc cộng tác.

Kiểm thử, đưa ra nhận xét và đánh giá kết quả.

1.3.2 Ý nghĩa

Nghiên cứu phương pháp lọc cộng tác. Chứng minh khả năng ứng dụng của lọc cộng tác cho bài toán xây dựng hệ thống gợi ý sản phẩm. Ứng dụng vào thương

mại điện tử, giúp tư vấn khách hàng lựa chọn sản phẩm nhanh và hiệu quả hơn.

1.4 Bố cục bài tập lớn

Phần còn lại của báo cáo bài tập lớn này được tổ chức như sau.

Chương 2 trình bày về Cơ sở lý thuyết

Chương này trình bày phương pháp lọc cộng tác và hai hướng tiếp cận chính của phương pháp lọc cộng tác là lọc cộng tác dựa trên người dùng và lọc cộng tác dựa trên sản phẩm.

Trong Chương 3, chúng em trình bày về Thực nghiệm và kết quả

Chương này chúng em sẽ thử nghiệm phương pháp lọc cộng tác, đưa ra kết quả và đánh giá.

Chương 4 là phần Kết luận

Nội dung chương này là tổng kết về báo cáo của nhóm, đưa ra hướng phát triển trong tương lai.

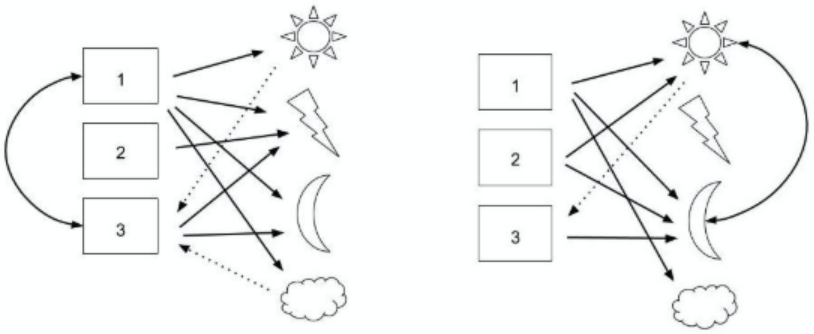
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu chung về phương pháp lọc cộng tác

Phương pháp lọc cộng tác hay hệ thống lọc cộng tác là phương pháp phân tích dữ liệu người dùng để tìm ra mối tương quan giữa các đối tượng người dùng. Lọc cộng tác hoạt động bằng cách xây dựng một cơ sở dữ liệu, lưu trữ dưới dạng ma trận người dùng (users) - sản phẩm (items) và mỗi dòng của nó là một véc tơ. Sau đó, phân tích dữ liệu, tính toán sự tương tự giữa các users với nhau để đưa ra gợi ý.

Có hai cách tiếp cận khác nhau trong lọc cộng tác, đó là lọc dựa trên item và lọc dựa trên người dùng. Lọc dựa trên item sử dụng sự tương đồng giữa các item để xác định liệu một người dùng có thích một item nào đó hay không, trong khi lọc dựa trên người dùng tìm kiếm những người dùng có các mẫu tiêu thụ tương tự như bạn và cung cấp cho bạn nội dung mà những người dùng tương tự này đã thấy thú vị. Cũng có các phương pháp kết hợp, mục tiêu của chúng là tận dụng những điểm mạnh của cả hai phương pháp này trong khi loại bỏ điểm yếu của mỗi phương pháp.

Hai phương pháp chính trong lọc cộng tác: dựa trên mô hình và dựa trên bộ nhớ. Bài báo cáo này sẽ thảo luận về lọc cộng tác dựa trên bộ nhớ, vì lọc dựa trên người dùng và lọc dựa trên item thuộc loại này. Hai phương pháp này chủ yếu khác nhau ở điều gì được tính đến khi tính toán các đề xuất. Lọc cộng tác dựa trên mục tìm kiếm mẫu tương đồng giữa các mục và đề xuất chúng cho người dùng dựa trên thông tin tính toán, trong khi lọc dựa trên người dùng tìm kiếm những người dùng tương tự và đề xuất cho họ dựa trên những gì người khác có các mẫu tiêu thụ tương tự đã đánh giá cao.



Hình 2.1: Hình ảnh mô tả cách tiếp cận khác nhau dựa trên người dùng và dựa trên item trong lọc cộng tác. Các mũi tên đứt quãng thể hiện các khuyến nghị dựa trên sở thích và điểm tương đồng của người dùng ở bên trái và dựa trên các mục tương tự ở bên phải.

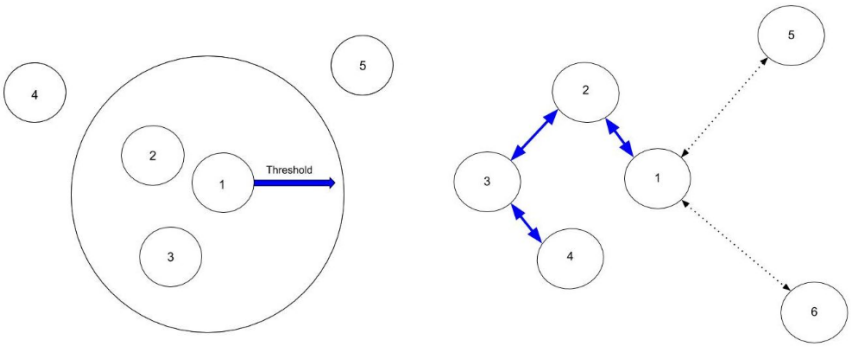
Lọc cộng tác là một trong những thuật toán được sử dụng rộng rãi nhất để đề xuất sản phẩm, và nó được coi là hiệu quả.

Các giải pháp kết hợp có thể hữu ích để đề xuất nội dung cho người dùng có gu thưởng thức độc đáo hoặc nhiều sở thích, vì sẽ khó hơn để tìm thấy một "người hàng xóm gần" hoặc ai đó có mẫu tiêu thụ tương tự với người dùng đó. Một giải pháp dựa trên item hoặc dựa trên người dùng thông thường có thể không đáp ứng được nhu cầu trong tình huống này.

2.2 User-user Collaborative Filtering

Báo cáo tập trung vào phương pháp "k-nearest neighbour" để đề xuất, trong đó xem xét các mẫu đánh giá của người dùng và tìm ra "k người hàng xóm gần nhất", tức là người dùng có các đánh giá tương tự với bạn. Sau đó, thuật toán tiếp tục đề xuất cho bạn dựa trên các đánh giá của những người hàng xóm này.

Trong một khu vực hàng xóm cố định, thuật toán tìm ra X người dùng giống nhất với bạn và sử dụng chúng làm cơ sở cho việc đề xuất. Trong một khu vực hàng xóm dựa trên ngưỡng, tất cả các người dùng nằm trong ngưỡng, tức là đủ giống nhau đủ, được sử dụng để cung cấp các đề xuất[8]. Báo cáo này sẽ sử dụng khu vực hàng xóm dựa trên ngưỡng vì nó hợp lý hơn khi sử dụng dữ liệu đủ giống nhau, và không đưa ra các đề xuất kém chất lượng cho một số người dùng chỉ vì người hàng xóm gần nhất xa xôi. Điều này sẽ dẫn đến một số người dùng nhận được các đề xuất tốt hơn so với những người khác (vì họ có nhiều người dùng giống nhau hơn để thuật toán làm việc), nhưng ít nhất nó sẽ không đưa ra các đề xuất kém chất lượng mà không được yêu cầu. Nó cũng sẽ không bỏ qua các người dùng giống nhau chỉ vì một số người dùng có thể giống nhau hơn, và hợp lý khi sử dụng tất cả dữ liệu tốt mà chúng ta có.



Hình 2.2: Hình ảnh bên trái mô tả vùng lân cận dựa trên ngưỡng. Người dùng 1 sẽ nhận được đề xuất từ người dùng 2 và 3, nhưng không phải từ 4 và 5 vì họ nằm ngoài phạm vi ngưỡng. Hình ảnh bên phải mô tả một vùng lân cận có kích thước cố định. Người dùng 1 sẽ nhận được đề xuất từ người dùng 2, 3 và 4, nhưng không phải từ 5 và 6 vì trong ví dụ này sử dụng ba người hàng xóm gần nhất để đưa ra khuyến nghị

2.2.1 Similarity functions - Hàm tương đồng

Công việc quan trọng nhất phải làm trước tiên trong User-user Collaborative Filtering là phải xác định được sự giống nhau (similarity) giữa hai users. Dữ liệu duy nhất chúng ta có là Utility matrix Y, vậy nên sự giống nhau này phải được xác định dựa trên các cột tương ứng với hai users trong ma trận này. Xét ví dụ trong Hình 2.3.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	5	5	2	0	1	?	?
i_1	3	?	?	0	?	?	?
i_2	?	4	1	?	?	1	2
i_3	2	2	3	4	4	?	4
i_4	2	0	4	?	?	?	5

Hình 2.3: Ví dụ về utility matrix dựa trên trọng số sao một user rate cho một item

Giả sử có các users từ $u0$ đến $u6$ và các items từ i_0 đến i_4 trong đó các số trong mỗi ô vuông thể hiện số sao mà mỗi user đã rated cho item với giá trị cao hơn thể hiện mức độ quan tâm cao hơn. Các dấu hỏi chấm là các giá trị mà hệ thống cần phải đi tìm. Đặt mức độ giống nhau của hai users u_i, u_j là $sim(u_i, u_j)$

Quan sát đầu tiên chúng ta có thể nhận thấy là các u_0, u_1 thích i_0, i_1, i_2 và không thích i_3, i_4 cho lắm. Điều ngược lại xảy ra ở các users còn lại. Vì vậy, một similiarity function tốt cần đảm bảo:

$$sim(u_0, u_1) > sim(u_0, u_i), \forall \boldsymbol{i} > 1.$$

Từ đó, để xác định mức độ quan tâm của u_0 lên i_2, chúng ta nên dựa trên hành vi của $u_1$lên sản phẩm này. Rất may rằng u1 đã thích i_2 nên hệ thống cần recommend i_2 cho u_0.

Câu hỏi đặt ra là: hàm số similarity nào là tốt? Để đo similarity giữa hai users, cách thường làm là xây dựng feature vector cho mỗi user rồi áp dụng một hàm có khả năng đo similarity giữa hai vectors đó. Với mỗi user, thông tin duy nhất chúng ta biết là các ratings mà user đó đã thực hiện, tức cột tương ứng với user đó trong Utility matrix. Tuy nhiên, khó khăn là các cột này thường có rất nhiều mising ratings vì mỗi user thường chỉ rated một số lượng rất nhỏ các items. Cách khắc phục là bằng cách nào đó, ta giúp hệ thống điền các giá trị này sao cho việc điền không làm ảnh hưởng nhiều tới sự giống nhau giữa hai vector. Việc điền này chỉ phục vụ cho việc tính similarity chứ không phải là suy luận ra giá trị cuối cùng.

Vậy mỗi dấu '?' nên được thay bởi giá trị nào để hạn chế việc sai lệch quá nhiều? Một lựa chọn có thể nghĩ tới là thay các dấu '?' bằng giá trị '0'. Điều này không thực sự tốt vì giá trị '0' tương ứng với mức độ quan tâm thấp nhất. Một giá trị an toàn hơn là 2.5 vì nó là trung bình cộng của 0, mức thấp nhất, và 5, mức cao nhất. Tuy nhiên, giá trị này có hạn chế đối với những users dễ tính hoặc khó tính. Với các users dễ tính, thích tương ứng với 5 sao, không thích có thể ít sao hơn 1 chút, 3 sao chẳng hạn. Việc chọn giá trị 2.5 sẽ khiến cho các items còn lại là quá negative đối với user đó. Điều ngược lại xảy ra với những user khó tính hơn khi chỉ cho 3 sao cho các items họ thích và ít sao hơn cho những items họ không thích.

Một giá trị khả dĩ hơn cho việc này là trung bình cộng của các ratings mà user tương ứng đã thực hiện. Việc này sẽ tránh được việc users quá khó tính hoặc dễ tính, tức lúc nào cũng có những items mà một user thích hơn so với những items khác.

Ta xét ví dụ trong hình 2.4: a) Utility Matrix ban đầu.

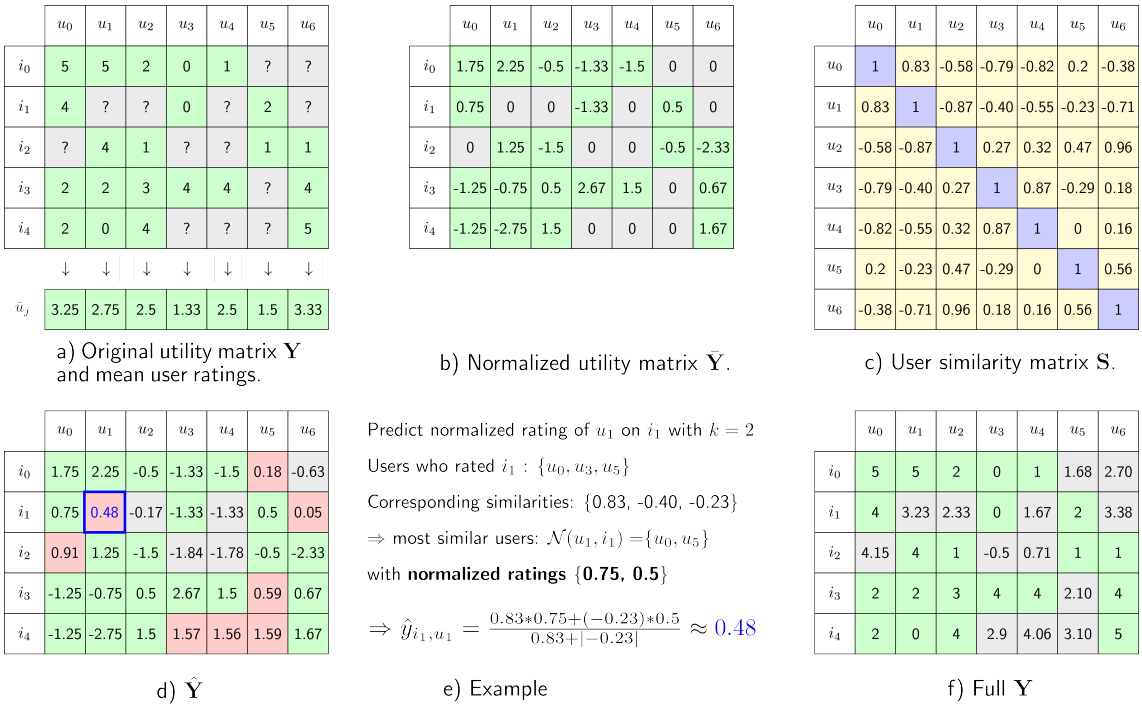
b) Utility Matrix đã được chuẩn hoá.

c) User similarity matrix.

d) Dự đoán các (normalized) ratings còn thiếu.

e) Ví dụ về cách dự đoán normalized rating của u_1 cho i_1.

f) Dự đoán các (denormalized) ratings còn thiếu.



Hình 2.4: Ví dụ mô tả User-user Collaborative Filtering

2.2.2 Chuẩn hoá dữ liệu

Hàng cuối cùng trong Hình 2.4a) là giá trị trung bình của ratings cho mỗi user. Giá trị cao tương ứng với các user dễ tính và ngược lại. Khi đó, nếu tiếp tục trừ từ mỗi rating đi giá trị này và thay các giá trị chưa biết bằng 0, ta sẽ được normalized utility matrix như trong Hình 2b).

- Việc trừ đi trung bình cộng của mỗi cột khiến trong trong mỗi cột có những giá trị dương và âm. Những giá trị dương tương ứng với việc user thích item, những giá trị âm tương ứng với việc user không thích item. Những giá trị bằng 0 tương ứng với việc chưa xác định được liệu user có thích item hay không.
- Về mặt kỹ thuật, số chiều của utility matrix là rất lớn với hàng triệu users và items, nếu lưu toàn bộ các giá trị này trong một ma trận thì khả năng cao là sẽ không đủ bộ nhớ. Quan sát thấy rằng vì số lượng ratings biết trước thường là một số rất nhỏ so với kích thước của utility matrix, sẽ tốt hơn nếu chúng ta lưu ma trận này dưới dạng sparse matrix, tức chỉ lưu các giá trị khác không và vị trí của giá trị khác không và vị trí của chúng. Vì vậy, tốt hơn hết, các dấu '?' nên được thay bằng giá trị '0', tức chưa xác định liệu user có thích item hay không. Việc này không những tối ưu bộ nhớ mà việc tính toán similarity matrix sau này cũng hiệu quả hơn.

Trong Collaborative Filtering, các đánh giá còn thiếu (missing rating) cũng được xác định dựa trên thông tin về $kneighborusers$. Tất nhiên, chúng ta chỉ quan tâm tới các người dùng đã đánh giá sản phẩm đang xét. Predicted rating thường được xác định là trung bình có trọng số của các ratings đã chuẩn hoá.

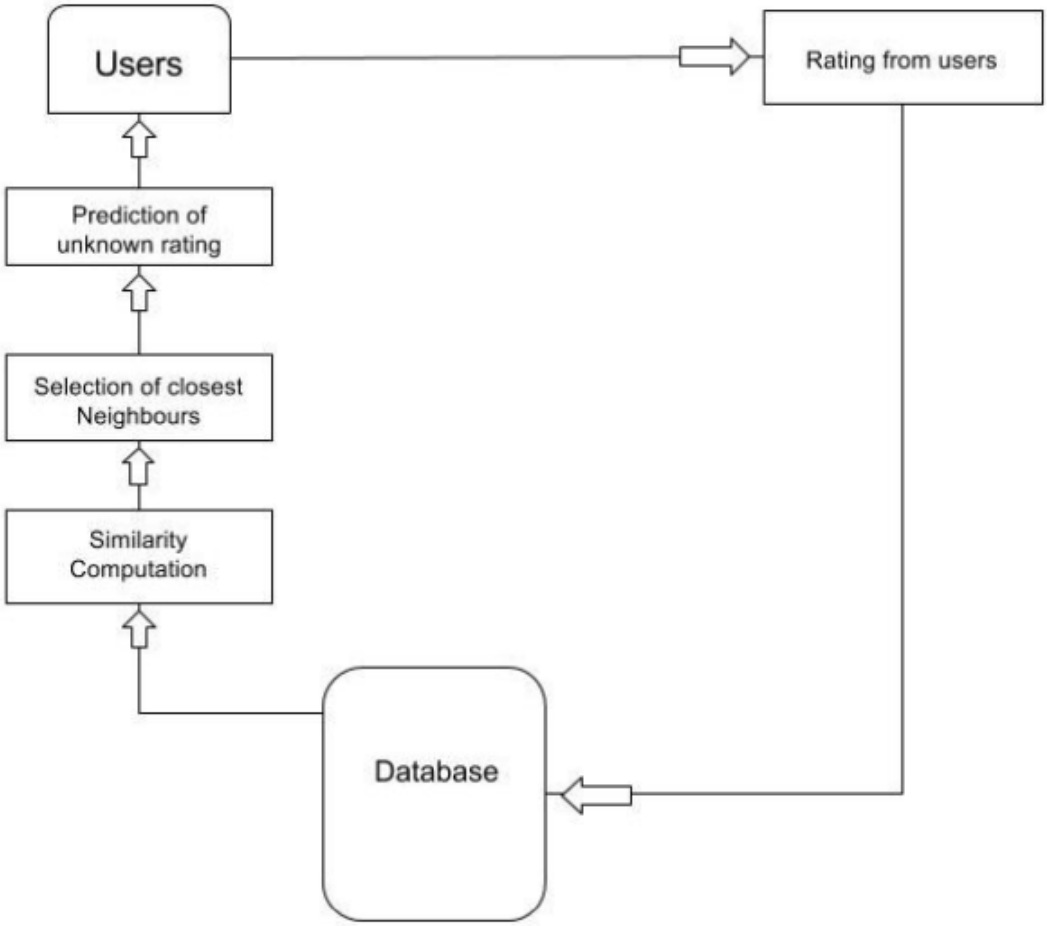
Công thức phổ biến được sử dụng để dự đoán rating của u cho i là:

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in \mathcal{N}(u,i)} \bar{y}_{i,u_j} \text{sim}(u,u_j)}{\sum_{u_j \in \mathcal{N}(u,i)} |\text{sim}(u,u_j)|}$$

Trong đó $\mathcal{N}(u,i)$ là tập hợp k *users* trong neighborhood (tức có similarity cao nhất) của u mà đã đánh giágiá i.

2.3 Item-item Collaborative Filtering

Lọc cộng tác dựa trên item được giới thiệu vào năm 1998 bởi Amazon. Không giống như lọc cộng tác dựa trên người dùng, lọc dựa trên mục xem xét sự tương đồng giữa các mục khác nhau, và thực hiện điều này bằng cách lưu ý xem bao nhiêu người dùng đã mua item X cũng đã mua item Y. Nếu tương quan đủ cao, có thể giả định rằng có một sự tương đồng giữa hai item, và chúng có thể được cho là giống nhau. Item Y sẽ từ đó được đề xuất cho người dùng đã mua item X và ngược lại.



Hình 2.5: Hình ảnh mô tả biểu đồ về cách xếp hạng của người dùng ảnh hưởng đến đề xuất của họ

Ý tưởng:

Nếu chúng ta tính toán similarity giữa các items rồi recommend những items gần giống với item yêu thích của một user - đây là cách tiếp cận thứ hai này hay còn được gọi là Item-item Collaborative Filtering. Hướng tiếp cận này được sử dụng nhiều trong thực tế hơn.

Quy trình dự đoán các đánh giá còn thiếu (missing rating) cũng tương tự như trong User-user CF (Hình 2.6):

a) Utility Matrix ban đầu.

b) Utility Matrix đã được chuẩn hoá.

c) User similarity matrix.

d) Dự đoán các (normalized) ratings còn thiếu.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6	
i_0	5	5	2	0	1	?	?	→ 2.6
i_1	4	?	?	0	?	2	?	→ 2
i_2	?	4	1	?	?	1	1	→ 1.75
i_3	2	2	3	4	4	?	4	→ 3.17
i_4	2	0	4	?	?	?	5	→ 2.75

a) Original utility matrix $\mathbf{Y}$ and mean item ratings.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	2.4	2.4	-.6	-2.6	-1.6	0	0
i_1	2	0	0	-2	0	0	0
i_2	0	2.25	-0.75	0	0	-0.75	-0.75
i_3	-1.17	-1.17	-0.17	0.83	0.83	0	0.83
i_4	-0.75	-2.75	1.25	0	0	0	2.25

b) Normalized utility matrix $\bar{\mathbf{Y}}$.

	i_0	i_1	i_2	i_3	i_4
i_0	1	0.77	0.49	-0.89	-0.52
i_1	0.77	1	0	-0.64	-0.14
i_2	0.49	0	1	-0.55	-0.88
i_3	-0.89	-0.64	-0.55	1	0.68
i_4	-0.52	-0.14	-0.88	0.68	1

c) Item similarity matrix $\mathbf{S}$.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	2.4	2.4	-.6	-2.6	-1.6	-0.29	-1.52
i_1	2	2.4	-0.6	-2	-1.25	0	-2.25
i_2	2.4	2.25	-0.75	-2.6	-1.20	-0.75	-0.75
i_3	-1.17	-1.17	-0.17	0.83	0.83	0.34	0.83
i_4	-0.75	-2.75	1.25	1.03	1.16	0.65	2.25

d) Normalized utility matrix $\bar{\mathbf{Y}}$.

Hình 2.6: Ví dụ mô tả Item-Item Collaborative Filtering

2.4 Các vấn đề trong hệ thống Collaborative Filtering

Các vấn đề thường gặp trong lọc cộng tác bao gồm các vấn đề sau đây:

The early rater problem xảy ra khi một người dùng mới được giới thiệu vào hệ thống và chưa có đủ lượng đánh giá cho một số lượng đáng kể các mục để dịch vụ có thể bắt đầu đề xuất các mục tương tự. Một giải pháp đơn giản cho vấn đề này là yêu cầu người dùng đánh giá nội dung mà họ có thể đã tiêu thụ trên một trang web hoặc nền tảng khác để cho thuật toán có đủ dữ liệu để bắt đầu đề xuất sản phẩm với khả năng thành công chấp nhận được. Tuy nhiên, nếu có một lượng lớn người dùng khác nhau nhưng lại có sở thích tương tự với người dùng hiện tại, nhưng lại đánh giá các mục khác nhau một cách rất khác biệt, thì việc xác định người hàng xóm gần nhất trở nên khó khăn hơn. Điều này làm cho quá trình tìm ra người hàng xóm gần nhất trở nên phức tạp và có thể dẫn đến kết quả không chính xác.

The sparsity problem xảy ra khi có quá ít thông tin để làm việc, điều này khiến cho hệ thống không thể cung cấp cho người dùng các ước lượng đáng tin cậy về những sản phẩm họ có khả năng ưa thích. Độ rải rác xảy ra khi dữ liệu phân tán quá rộng, và trong khi người dùng có thể đã đánh giá một lượng phim khá lớn, nhưng quá ít người đã đánh giá cùng một bộ phim để đưa ra dự

đoán chính xác. Điều này gây ra khó khăn trong việc tạo ra các đề xuất chính xác và đáng tin cậy cho người dùng.

Cold start problem xảy ra khi một hoặc một số người dùng hoặc sản phẩm được thêm vào hệ thống, và không có đủ dữ liệu được ghi lại để cung cấp các đề xuất tối ưu. Điều này có thể ảnh hưởng đến toàn bộ thuật toán đề xuất nếu dịch vụ mới được thành lập vì nó không thể cung cấp các đề xuất chấp nhận được cho bất kỳ người dùng nào cho đến thời điểm hiện tại. Đây là một vấn đề đã được nghiên cứu trong một thời gian dài và vẫn là một vấn đề trong nhiều hệ thống đề xuất ngày nay.

The gray sheep là một người dùng không có người hàng xóm gần nhất rõ ràng và có vẻ đánh giá nội dung theo một mẫu không thể dự đoán giống như bất kỳ người dùng nào khác. Một "gray sheep" là một vấn đề vì có thể khó khăn để ước lượng người dùng này có thể thích hoặc không thích những gì khi không có các mẫu tiêu thụ hoặc đánh giá tương tự. Một khoảng thời gian và số lượng đánh giá nhất định cần thiết để phần tử mới này được giới thiệu vào hệ thống một cách sao cho hoạt động tương tự như các phần tử khác trong đó. Đó chỉ là quá trình dẫn đến thời điểm khi phần tử đã được thiết lập chính xác trong môi trường.

The shilling attack xảy ra khi một cá nhân hoặc nhóm người tạo nhiều tài khoản để quảng cáo nội dung nhất định và làm mất đi sự quan tâm của người dùng đối với các nội dung khác, nhằm thúc đẩy sản phẩm của họ và làm ảnh hưởng xấu đến đối thủ của họ. Đây là một cố gắng can thiệp vào người dùng để mua, xem hoặc đăng ký theo dõi một loại nội dung cụ thể dựa trên một nhiệm vụ ẩn.

2.5 Giải quyết vấn đề người dùng mới và sản phẩm mới ở Collaborative Filtering

Để khắc phục vấn đề trên, chúng ta sẽ kết hợp cả lọc cộng tác (collaborative filtering) và lọc dựa trên một số thuộc tính của người dùng cung cấp để dự đoán sở thích của khách hàng mới.

Cụ thể, khi người dùng mới đăng ký tài khoản, hệ thống sẽ yêu cầu họ cung cấp một số thông tin cá nhân và sở thích thông qua một biểu mẫu (form). Các thông tin này có thể bao gồm:

- Tính cách: Ví dụ như hướng ngoại hay hướng nội,...
- Nghề nghiệp: Để xác định các sản phẩm phù hợp với công việc của họ.
- Thu nhập: Để gợi ý các sản phẩm phù hợp với khả năng tài chính.

- Sở thích: Ví dụ như thể loại phim yêu thích, loại sách yêu thích, hoặc các hoạt động giải trí yêu thích.

Lợi ích

- Cung cấp dữ liệu ban đầu: Những thông tin này giúp hệ thống có dữ liệu ban đầu để đưa ra các gợi ý ngay lập tức, mà không cần phải chờ đợi người dùng mới thực hiện nhiều hoạt động.
- Cá nhân hóa gợi ý: Dựa trên thông tin cá nhân, hệ thống có thể sử dụng Content-Based Filtering để đưa ra các gợi ý phù hợp với sở thích và đặc điểm của người dùng mới, từ đó sẽ giúp hệ thống tư vấn tốt hơn cho người dùng mới này.

Đối với sản phẩm mới nhập vào, chúng có thể được hiển thị đầu tiên trên trang web và có biểu tượng 'New' để nhận biết đây là sản phẩm mới của hệ thống.Điều này giúp thu hút sự chú ý của người dùng tới các sản phẩm mới. Ngoài ra, khi người dùng xem chi tiết một sản phẩm, trang web sẽ hiển thị các sản phẩm tương tự dựa trên một số thuộc tính như:

- Thể loại: Đối với phim hoặc sách.
- Thương hiệu: Đối với các sản phẩm tiêu dùng.
- Giá: Để đưa ra các gợi ý trong cùng tầm giá.

Lợi ích

- Tăng khả năng hiển thị sản phẩm: Giúp người dùng dễ dàng khám phá các sản phẩm kể cả sản phẩm mới.
- Tăng tính liên quan của gợi ý: Sử dụng các thuộc tính tương tự giúp hệ thống đưa ra các gợi ý có liên quan, ngay cả khi sản phẩm chưa có nhiều đánh giá.

2.6 Adjusted Cosine Similarity

Adjusted Cosine Similarity (ACS) là một phương pháp tính toán độ tương đồng giữa người dùng và vật phẩm dựa trên sự khác biệt về góc vectơ giữa các đánh giá. Nó được tính toán theo công thức sau:

$$AC(i,j) = \frac{\sum_{i \neq i_p}(r_{ui} - \overline{r}_u)(r_{uj} - \overline{r}_u)}{\sqrt{\sum_{i \neq i_p}(r_{ui} - \overline{r}_u)^2 \sum_{i \neq i_p}(r_{uy} - \overline{r}_u)^2}}$$

Trong đó:

* r_{ui} là đánh giá do người dùng u dành cho vật phẩm i

* r_{uj} là đánh giá do người dùng u dành cho vật phẩm j

* $\overline{r}_u$ là đánh giá trung bình của người dùng u

ACS đã điều chỉnh cũng xem xét rằng người dùng có thể đánh giá các mục khác nhau với các mức độ cao khác nhau. Một người dùng có thể có điểm trung bình là 3 và coi 4 là một đánh giá cao, trong khi người dùng khác có thể có điểm trung bình là 4 và coi 3 là một điểm số thực sự tồi tệ. Chúng ta thực hiện điều chỉnh cho điều này bằng cách trừ điểm số trung bình của mỗi người dùng từ các điểm số của người dùng đó.

ACS đã điều chỉnh hiệu quả hơn một chút so với tương quan Pearson khi xử lý các đề xuất dựa trên mục, và cũng hiệu quả như vậy đối với lọc cộng tác dựa trên người dùng.

2.7 Root Mean Square Estimation (RMSE)

Root Mean Square Estimation (RMSE) là một phương pháp ước tính sai số trung bình giữa giá trị dự đoán và giá trị thực tế. RMSE được sử dụng như một biện pháp bảo vệ để giảm thiểu tác động cá nhân từ lỗi của từng thiết bị đo lường riêng lẻ, do đó không có sai số ước tính chính nào ảnh hưởng quá nhiều đến kết quả. RMSE được tính toán theo công thức sau:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \overline{y})^2}$$

Trong đó:

* n là số lượng ước tính

* y_j là ước tính hiện tại

* $\overline{y}$ là giá trị trung bình của các ước tính

RMSE có một số ưu điểm so với các phương pháp ước tính sai số khác, chẳng hạn như Mean Absolute Error (MAE). RMSE ít bị ảnh hưởng bởi các giá trị ngoại lệ và có thể xử lý tốt hơn các trường hợp có nhiều giá trị sai số lớn.

2.7.1 Kết luận chương 2

Trong chương này, chúng em đã trình bày về hai phương pháp chính trong lọc cộng tác là User-user Collaborative Filtering và Item-item Collaborative Filtering. Trong chương tiếp theo, chúng em sẽ thử nghiệm và đưa ra kết quả đánh giá về phương pháp lọc cộng tác ở cả hai cách tiếp cận trên.

CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

3.1 Phương pháp

3.1.1 Software và Hardware

Chương trình được viết bằng ngôn ngữ lập trình Python, sử dụng các thư viện pandas, numpy và scikit-learn (sklearn). Chúng em chọn ngôn ngữ này vì các thử nghiệm của chúng em đòi hỏi nhiều tính toán ma trận, và Python sẽ là lựa chọn tốt nhất và nhanh nhất cho trường hợp này.

- `http://pandas.pydata.org/`
- `http://www.numpy.org/`
- `http://scikit-learn.org/stable/`

Các công cụ này được đề xuất mạnh mẽ cho học máy trong Python và cung cấp nhiều hàm thực tiễn giúp việc triển khai trở nên dễ dàng hơn.

Kết quả được tính toán trên laptop HP pavilion-14 với bộ xử lý Intel, Core i5, 11350U với dung lượng lên đến 2.4 gigahertz và bộ nhớ truy cập ngẫu nhiên DDR3 L 8 gigabyte.

3.1.2 Tập dữ liệu

Các bài kiểm tra đầu tiên được thực hiện trên cơ sở dữ liệu "ml-100k", chứa hơn một trăm nghìn đánh giá từ 671 người dùng khác nhau trên 9066 bộ phim khác nhau. Chúng em chọn bộ dữ liệu này vì cơ sở dữ liệu MovieLens đề xuất bộ dữ liệu này cho mục đích giáo dục và nghiên cứu. Nó cũng đủ lớn để cung cấp kết quả hợp lý mà không làm mất một khoảng thời gian không hợp lý cho mỗi lần thực thi.

Bản thử nghiệm thứ hai được thực hiện trên một cơ sở dữ liệu có kích thước tương đương, với một trăm nghìn đánh giá, với số lượng đánh giá bằng nhau và khoảng cùng số lượng người dùng, nhưng chỉ có một phần tư số lượng phim so với cơ sở dữ liệu khác. Điều này dẫn đến mỗi người đã xem các bộ phim tương tự với mọi người khác trong cơ sở dữ liệu, có nghĩa là có nhiều đánh giá và dữ liệu về mỗi bộ phim cụ thể. Điều này cũng có nghĩa là người dùng giống nhau hơn với nhau vì một số lượng người bằng nhau được phân bố trên một tập hợp nhỏ hơn các bộ phim. Do đó, kết quả được kỳ vọng sẽ tốt hơn.

Lý do sử dụng nhiều bộ dữ liệu là để xác minh rằng các lý thuyết là chính xác và một cơ sở dữ liệu với nhiều người tương tự nhau cung cấp nhiều dữ liệu hơn và sẽ mang lại kết quả tốt hơn. Cũng thú vị để xem xét xem lọc cộng tác dựa trên người dùng hay dựa trên mục bị ảnh hưởng nhiều hơn khi dữ liệu được nén lại.

Cơ sở dữ liệu được đề cập tới có thể được tìm thấy trên trang web của GroupLens (`https://grouplens.org/datasets/movielens/`) và tên của cơ sở dữ liệu này là ml-100k.zip.

3.1.3 Triển khai

Hai ma trận được tạo ra từ bộ dữ liệu, một cho huấn luyện và một cho kiểm tra. Mỗi hàng trong cả hai ma trận đại diện cho một người dùng duy nhất trong khi mỗi cột đại diện cho một mục duy nhất. Thuật toán sau đó tạo ra hai ma trận mới với các tương đồng ước tính của chúng được tính toán bằng công thức tương đồng cosine (dựa trên ma trận huấn luyện), một cho việc lọc cộng tác dựa trên người dùng và một cho việc lọc cộng tác dựa trên mục. Sau đó, thuật toán cố gắng dự đoán cách mà một số người dùng nhất định sẽ đánh giá các mục nhất định dựa trên hai ma trận đã tính toán trước đó, và sử dụng phương pháp ước tính bình phương gốc để tính toán mức độ chệch lệch của ước tính dựa trên ma trận kiểm tra. Chương trình sau đó được thực thi một trăm lần cho mỗi bài kiểm tra, ghi kết quả của nó cho cả lọc cộng tác dựa trên người dùng và lọc cộng tác dựa trên mục sau mỗi lần lặp. Kết quả sau đó được đưa vào một tập tin excel, nơi các biểu đồ được tạo dựa trên dữ liệu từ các biểu đồ.

Trong bài kiểm tra ban đầu, chúng em sử dụng 75% của cơ sở dữ liệu làm dữ liệu huấn luyện và 25% làm dữ liệu kiểm tra. Dữ liệu được ghép lại và được sử dụng để tạo biểu đồ trình bày kết quả.

Bài kiểm tra thứ hai được thực hiện một lần nữa trên cơ sở dữ liệu đầy đủ của một trăm nghìn đánh giá, nhưng lần này với 90% dữ liệu làm dữ liệu huấn luyện và 10% làm dữ liệu kiểm tra.

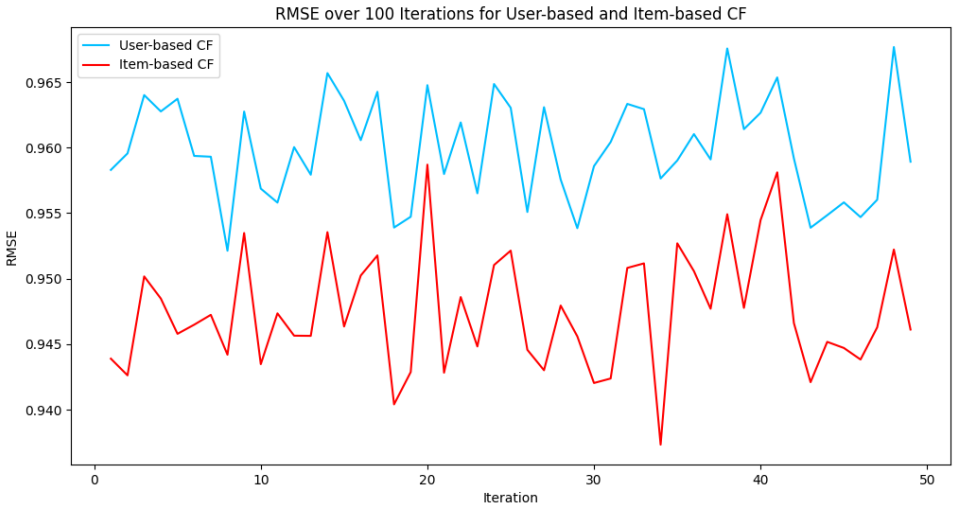
Bài kiểm tra thứ ba được thực hiện trên cơ sở dữ liệu đầy đủ lần thứ ba, nhưng lần này với một nửa (50%) của cơ sở dữ liệu làm ma trận huấn luyện và nửa còn lại làm ma trận kiểm tra.

Bài kiểm tra thứ tư sử dụng 75% của cơ sở dữ liệu làm dữ liệu huấn luyện và 25% làm dữ liệu kiểm tra. Thử nghiệm với các giá trị k-neighbor khác nhau, k=5, k=10, k=20, k=30.

Cả 4 bài kiểm tra đều chạy trong khoảng thời gian từ 5 phút đến 10 phút. Với các giá trị k nhỏ hơn thử nghiệm sẽ chạy nhanh hơn và ngược lại, việc chia tập dữ liệu gần như không ảnh hưởng tới tốc độ của thử nghiệm.

3.2 Kết quả

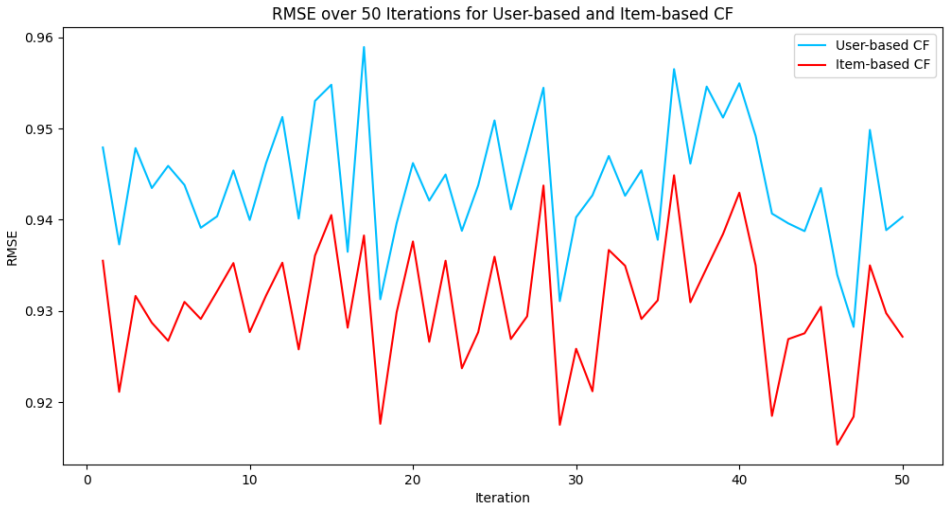
3.2.1 Thử nghiệm với tập dữ liệu 75% training, 25% testing



Hình 3.1: Đồ thị hiển thị kết quả của 50 lần thực thi trong bài kiểm tra với tập dữ liệu tỉ lệ 75/25. Trục x là số lần lặp hiện tại và trục y là giá trị RMSE. Đường màu đỏ đại diện cho kết quả từ ước lượng dựa trên item, và đường màu xanh đại diện cho kết quả dựa trên người dùng.

Lọc cộng tác dựa trên item vượt trội hơn so với lọc cộng tác dựa trên người dùng trong cả 50 lần thực thi.

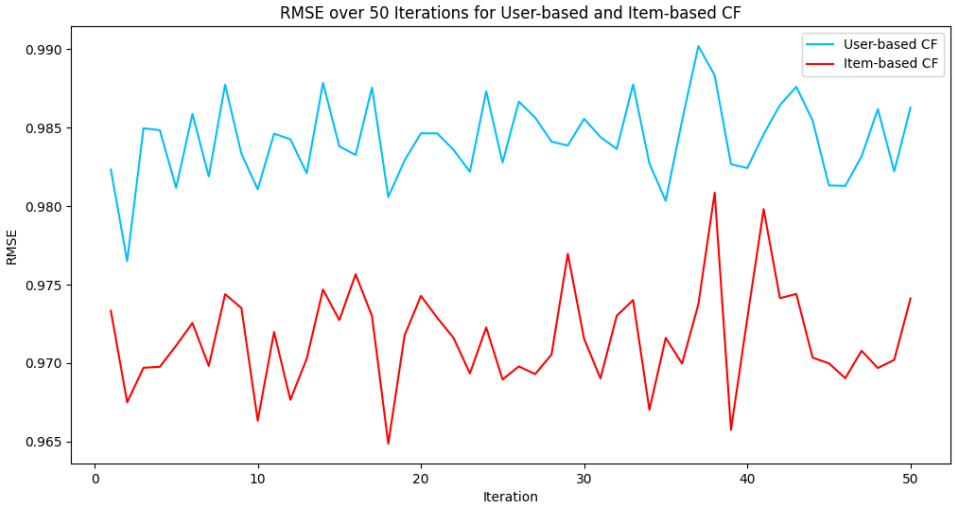
3.2.2 Thử nghiệm với tập dữ liệu 90% training, 10% testing



Hình 3.2: Đồ thị hiển thị kết quả của 50 lần thực thi trong bài kiểm tra với tập dữ liệu tỉ lệ 90/10. Trục x là số lần lặp hiện tại và trục y là giá trị RMSE. Đường màu đỏ đại diện cho kết quả từ ước lượng dựa trên item, và đường màu xanh đại diện cho kết quả dựa trên người dùng.

Lọc cộng tác dựa trên item đã vượt trội hơn so với lọc cộng tác dựa trên người dùng trong cả 50 lần thực thi. Dữ liệu huấn luyện 90% cho kết quả tốt hơn so với dữ liệu huấn luyện 75%.

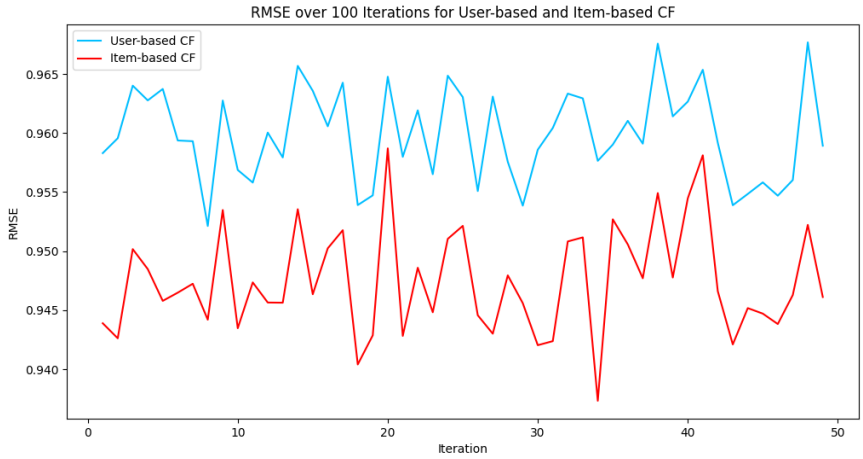
3.2.3 Thử nghiệm với tập dữ liệu 50% training, 50% testing



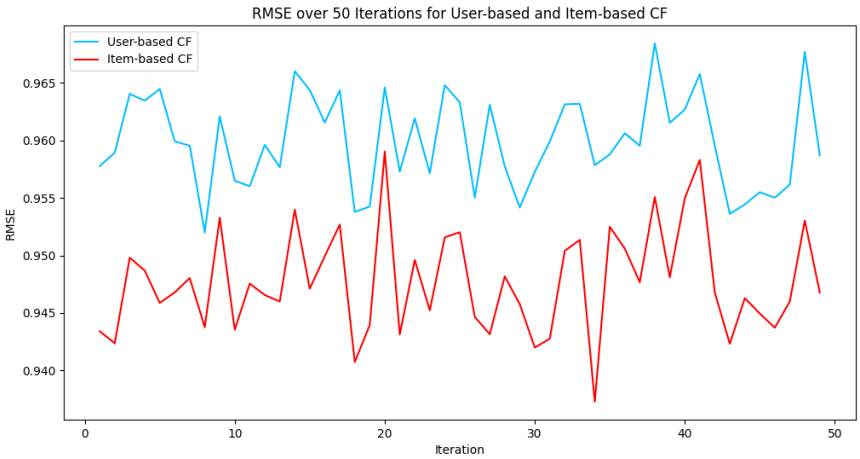
Hình 3.3: Đồ thị hiển thị kết quả của 50 lần thực thi trong bài kiểm tra với tập dữ liệu tỉ lệ 50/50. Trục x là số lần lặp hiện tại và trục y là giá trị RMSE. Đường màu đỏ đại diện cho kết quả từ ước lượng dựa trên item, và đường màu xanh đại diện cho kết quả dựa trên người dùng.

Lọc cộng tác dựa trên item vượt trội hơn so với lọc cộng tác dựa trên người dùng trong cả 50 lần thực thi. Biên độ dao động nằm trong phạm vi tương tự so với thử nghiệm tương ứng trên bộ dữ liệu đầu tiên.

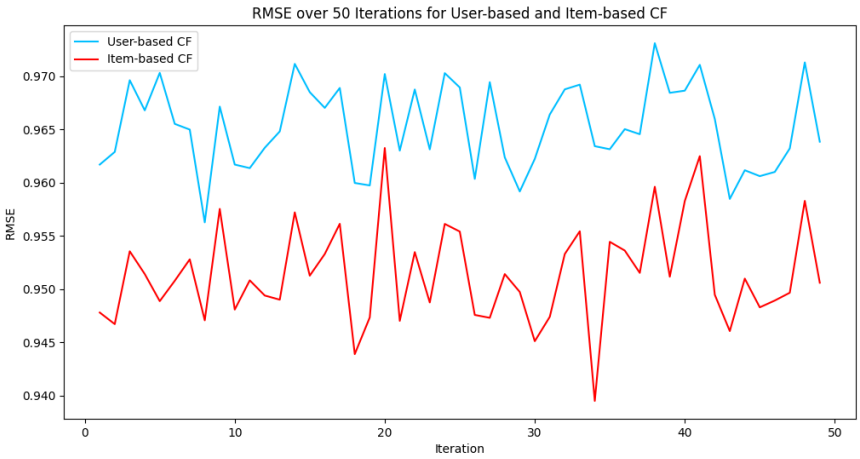
3.2.4 Thử nghiệm với giá trị k-neighbor khác nhau



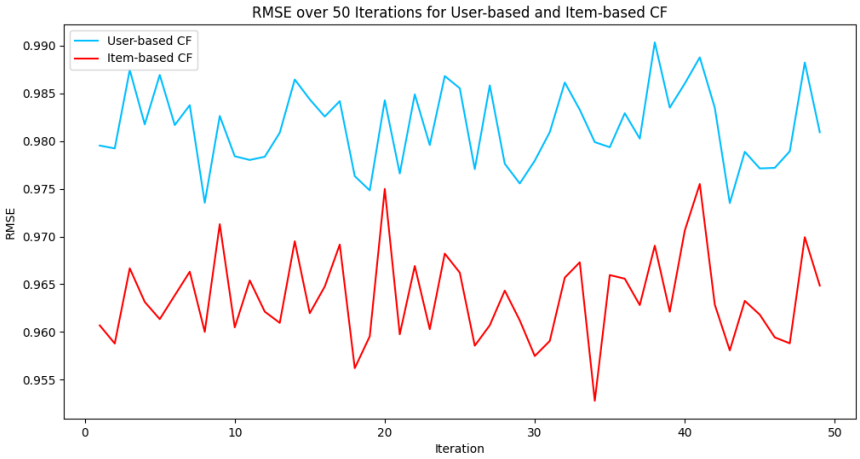
Hình 3.4: K=30



Hình 3.5: K=20



Hình 3.6: K=10



Hình 3.7: K=5

Giá trị RMSE giảm dần khi giá trị K tăng từ 10 đến 30. Điều này có nghĩa là hiệu suất dự đoán của thuật toán tăng khi giá trị K tăng. Sau khi đạt giá trị tối ưu tại K = 30, RMSE tăng nhẹ khi K tiếp tục tăng, có nghĩa là lợi ích của việc tăng K giảm dần khi K tăng.

3.3 Kết luận

Trong mỗi thử nghiệm trên toàn bộ cơ sở dữ liệu, lọc cộng tác dựa trên item tốt hơn, với độ lệch trung bình dao động từ khoảng 0.930 đến 0.965 gần hơn so với đánh giá thực tế. Các ước lượng dựa trên người dùng bị ảnh hưởng nhiều hơn bởi kích thước tập huấn luyện nhỏ hơn so với các ước lượng dựa trên mục. Lọc cộng tác dựa trên người dùng cũng có sự cải thiện nhẹ với các tập dữ liệu huấn luyện lớn hơn, nhưng mức độ cải thiện nhỏ hơn đáng kể.

Thử nghiệm số 3 cho cả hai phương pháp lọc cộng tác dựa trên item và người dùng, sử dụng 50% của cơ sở dữ liệu làm dữ liệu huấn luyện, có kết quả ổn định nhất. Điều này có thể là do ma trận kiểm tra lớn hơn, dẫn đến mỗi sự lệch lạc trong đó có ảnh hưởng nhỏ hơn đến bức tranh tổng thể. Điều này làm giảm các sự sai lệch phát sinh từ những người dùng "gray sheep". Cũng hợp lý rằng nó hoạt động kém hơn, vì có ít dữ liệu để huấn luyện, và các đánh giá lệch lạc trong ma trận kiểm tra có ảnh hưởng lớn hơn đến các ước lượng mà thuật toán này cung cấp so với hai phương pháp còn lại.

Khi các đỉnh và đáy của đường cong cho lọc cộng tác dựa trên mục và dựa trên người dùng dường như gần như giống hệt nhau cho mỗi thử nghiệm riêng lẻ, chúng ta có thể giả định rằng các thuật toán hoạt động giống nhau cho mỗi tập dữ liệu, và rằng chúng đơn giản là vượt trội so với nhau dựa trên kích thước của ma trận huấn luyện và kiểm tra, và rằng các giá trị cá nhân có ít hoặc không ảnh hưởng gì, ít nhất là đối với các tập dữ liệu đủ lớn.

Từ đây, chúng ta có thể kết luận rằng càng nhiều dữ liệu huấn luyện cung cấp kết quả tốt hơn trong khi dữ liệu kiểm tra lớn hơn cung cấp dao động nhỏ hơn. Kích thước của các dao động giữa việc có 50% dữ liệu kiểm tra và 25% hầu như không ảnh hưởng đến kết quả trong trường hợp này, và chúng ta kết luận rằng đối với các cơ sở dữ liệu lớn như thế này, lượng dữ liệu huấn luyện cao hơn sẽ vượt trội hơn so với lượng dữ liệu kiểm tra cao hơn, miễn là dữ liệu kiểm tra đủ lớn để không làm lệch kết quả quá nhiều bởi một đánh giá lẻ so với các đánh giá khác. Trong trường hợp này, 10% dữ liệu kiểm tra đã đủ.

Với thử nghiệm thứ 4, cả hai phương pháp User-based CF và Item-based CF đều cho thấy hiệu quả tương đương nhau trong việc dự đoán xếp hạng người dùng cho các mục. User-based CF có hiệu suất tốt hơn Item-based CF ở giai đoạn đầu,

nhưng sự khác biệt này không đáng kể. Số lượng lặp tối ưu cho cả hai phương pháp là khoảng 30. Giá trị K ảnh hưởng đến hiệu suất dự đoán của thuật toán theo cách không tuyến tính. Giá trị K tối ưu cho cả hai phương pháp là khoảng 30. Việc chọn giá trị K phù hợp có thể cải thiện đáng kể hiệu suất dự đoán của thuật toán.

3.4 Đánh giá

Chúng em đã chọn sử dụng Adjusted Cosine Similarity thay vì hệ số tương quan Pearson trong thuật toán của mình. Điều này là vì khi so sánh hai người dùng có sự khác biệt nhau, hệ số tương quan Pearson sẽ bỏ qua các mục mà chỉ một trong hai hoặc cả hai người dùng không đánh giá, trong khi Độ Tương Tự Cosine Điều Chỉnh vẫn giữ lại chúng nhưng đặt đánh giá là 0. Do đó, Độ Tương Tự Cosine Điều Chỉnh không chỉ xem xét dữ liệu giống nhau mà còn xem xét cả dữ liệu khác nhau, điều này sẽ mang lại cho chúng ta kết quả chính xác hơn .

3.5 Tổng kết

Báo cáo này cho thấy rằng các bộ dữ liệu đủ lớn để có thể sử dụng trong thực tế, lọc cộng tác dựa trên người dùng là ưu việt trong tất cả các trường hợp được kiểm tra. Số lượng lớn dữ liệu huấn luyện dẫn đến kết quả tốt hơn, và lượng dữ liệu kiểm tra không cần phải quá lớn để đạt được kết quả tốt. Các bộ dữ liệu có ít biến thiên về số lượng mục, và do đó có người dùng đồng nhất hơn dẫn đến kết quả ưu việt, đó là lý do tại sao các bài kiểm tra trong bộ dữ liệu thứ hai cho kết quả tốt hơn so với các bài kiểm tra trong bộ dữ liệu thứ nhất.

Lọc cộng tác dựa trên người dùng và dựa trên mục cung cấp kết quả có vẻ tương tự nhau, nhưng lọc cộng tác dựa trên item lại tốt hơn trên các kích thước được kiểm tra trong báo cáo này. Lọc cộng tác dựa trên item cũng cải thiện nhanh hơn khi lượng dữ liệu huấn luyện tăng lên.

Trong bộ dữ liệu thứ hai, kết quả dựa trên item cũng được cải thiện hơn (khoảng 0.1) so với dựa trên người dùng (được cải thiện khoảng 0.05).

CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Hệ tư vấn không phải là một đề tài mới, nhưng đã trở thành một phần quan trọng của các hệ thống thông tin hiện đại. Trong quá trình thực hiện đề tài "Sử dụng phương pháp lọc cộng tác cho bài toán gợi ý sản phẩm", chúng em đã đạt được một vài kết quả chính như sau:

- Trình bày khái quát về phương pháp lọc cộng tác
- Giới thiệu về phương pháp lọc công tác dựa trên người dùng và dựa trên sản phẩm
- Thử nghiệm và đưa ra đánh giá về phương pháp lọc cộng tác trên hai cách tiếp cận user-based CF và item-based CF

4.2 Hướng phát triển

Hướng mở của đề tài tiếp tục phát triển:

- Nghiên cứu phương pháp lọc cộng tác bằng phương pháp đa nhiệm và ứng dụng
- Phương pháp kết hợp lọc cộng tác và lọc nội dung dựa trên mô hình đồ thị
- Nghiên cứu thêm về các phương pháp phát triển hệ thống gợi ý khác

TÀI LIỆU THAM KHẢO

[1] Hà Thị Thanh Ngà, Nguyễn Đình Cường, *Xây dựng hệ thống gợi ý bằng thuật toán người láng giềng và thử nghiệm trên Movielens Dataset*, Hội nghị khoa học Công nghệ thông tin và Truyền thông ICT, Đà Lạt, 2017.

[2] Nguyễn Hùng Dũng, Nguyễn Thái Nghe, *Hệ thống gợi Ý sản phẩm trong bán hàng trực tuyến sử dụng kỹ thuật lọc cộng tác*, Tạp chí Khoa học Trường Đại học Cần Thơ, 2014.

[3] Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan, *Collaborative Filtering Recommender Systems*, Foundations and Trends in Human–Computer Interaction - Vol.4 - No.2, 2011.

ĐÁNH GIÁ ĐÓNG GÓP CÁC THÀNH VIÊN

Họ và tên	**Mã sinh viên**	**Công việc**
Nguyễn Minh Giang	B21DCCN304	Phần thực nghiệm và kết quả
Nguyễn Khánh Linh	B21DCCN484	Phần thực nghiệm và kết quả
Nguyễn Văn Hùng	B21DCCN417	Tạo slide, thuyết trình
Trần Công Hiếu	B21DCCN369	Implement thuật toán
Phạm Quỳnh Chi	B21DCCN177	Phần cơ sở lý thuyết, viết báo cáo