**NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN**

.......................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN

.......................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

**TÓM TẮT**

Tên đề tài: Xây dựng hệ thống phân tích và gợi ý du lịch thông minh

Sinh viên thực hiện: Tăng Văn Quốc Chí

Số thẻ SV: 102140060 Lớp: 14T2

Tại thời điểm hiện nay, các website mua bán sản phẩm xuất hiện rất nhiều và cạnh tranh lẫn nhau nên việc hiểu ý người dùng sớm và hợp lý là một điều vô cùng cần thiết. Thêm nữa, với lượng sản phẩm nhiều với số lượng hàng ngàn đến hàng triệu sản phẩm thì người dùng sẽ rất khó tiếp cận đến sản phẩm mình thích và cuối cùng là bỏ cuộc. Đồ án này xây dựng một hệ thống giúp phân tích và tạo ra các gợi ý để người dùng chính là người quản lý website có thể thử nghiệm độ tối ưu của thuật toán và áp dụng các thuật toán đó vào website của mình để tạo ra các gợi ý thích hợp với mỗi khách hàng của họ.

|  |  |
| --- | --- |
| ĐẠI HỌC ĐÀ NẴNG  **TRƯỜNG ĐẠI HỌC BÁCH KHOA**  KHOA CÔNG NGHỆ THÔNG TIN | **CỘNG HÒA XÃ HÔI CHỦ NGHĨA VIỆT NAM**  Độc lập - Tự do - Hạnh phúc |

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: Tăng Văn Quốc Chí Số thẻ sinh viên: 102140060

Lớp:14T2 Khoa: Công nghệ thông tin Ngành: Mạng và truyền thông

1. *Tên đề tài đồ án:*

Xây dựng hệ thống phân tích và gợi ý du lịch thông minh.

1. *Đề tài thuộc diện:*  *Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
2. *Các số liệu và dữ liệu ban đầu:*

Dữ liệu đầu vào được sinh ngẫu nhiên có quy luật (sử dụng thuật toán để sinh) kết hợp với dữ liệu thao tác của người dùng từ một đồ án khác.

1. *Nội dung các phần thuyết minh và tính toán:*

Lý thuyết và công thức được trích từ nhiều tài liệu báo cáo khoa học và được trích dẫn tại cuối báo cáo đồ án.

1. *Các bản vẽ, đồ thị ( ghi rõ các loại và kích thước bản vẽ ):*

Bản vẽ bao gồm các bản vẽ thuật toán được trích dẫn, sơ đồ hệ thống. Đồ thị có đồ thị so sánh độ chính xác giữa các thuật toán.

1. *Họ tên người hướng dẫn:* PGS. TS. Nguyễn Tấn Khôi
2. *Ngày giao nhiệm vụ đồ án:*  *06/03/2019*
3. *Ngày hoàn thành đồ án: 06/06/2019*

|  |  |
| --- | --- |
|  | *Đà Nẵng, ngày 05 tháng 06 năm 2019* |
| **Trưởng Bộ môn** …………………….. | **Người hướng dẫn** |

**LỜI NÓI ĐẦU**

Tại thời điểm hiện nay, các website mua bán sản phẩm xuất hiện rất nhiều và cạnh tranh lẫn nhau nên việc hiểu ý người dùng sớm và hợp lý là một điều vô cùng cần thiết và cấp bách. Từ đó mà việc sử dụng hệ gợi ý cho người dùng với các hệ thống đó là một việc vô cùng cần thiết.

Được sự đồng ý của khoa Công nghệ thông tin trường Đại học Bách khoa, Đại học Đà Nẵng và cán bộ hướng dẫn, em đã thực hiện đồ án “xây dựng hệ thống phân tích và gợi ý du lịch thông minh” để người dùng có thể dễ dàng đánh giá thuật toán và áp dụng nó vào hệ gợi ý một cách dễ dàng. Ứng dụng sử dụng các thuật toán không có ngữ cảnh và có ngữ cảnh, và đưa ra các gợi ý cho từng khách hàng của website.

Đồ án bao gồm 4 chương:

Chương 1: Tổng quan đề tài

Chương 2: Cơ sở lý thuyết

Chương 3: Thiết kế và cài đặt hệ thống

Chương 4: Kết quả thực nghiệm

Em xin chân thành cảm ơn thầy Nguyễn Tấn Khôi đã chỉ bảo và tận tình hỗ trợ để em có thể hoàn thành đồ án này. Vì thời gian có hạn và trình độ có hạn nên em sẽ không tránh khỏi thiếu xót. Em mong nhận được nhận xét và đóng góp của thầy cô để em có thể hoàn thiện hơn đồ án cũng như kiến thức ạ.

Đà Nẵng, ngày 05 tháng 06 năm 2019

Sinh viên thực hiện

Tăng Văn Quốc Chí

**CAM ĐOAN**

*Tôi xin cam đoan :*

1. *Nội dung trong luận văn này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của Thầy Nguyễn Tấn Khôi.*
2. *Các tham khảo dùng trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.*
3. *Nếu có những sao chép không hợp lệ, vi phạm, tôi xin chịu hoàn toàn trách nhiệm.*

Sinh viên thực hiện

Tăng Văn Quốc Chí

**MỤC LỤC**

Tóm tắt

Nhiệm vụ đồ án

Lời nói đầu và cảm ơn i

Lời cam đoan liêm chính học thuật ii

Mục lục iii

Danh sách các bảng biểu vi

Danh sách hình vii

Danh sách các cụm từ viết tắt viii

[Chương 1: TỔNG QUAN ĐỀ TÀI 2](#_Toc10674763)

[1.1. Giới thiệu đề tài 2](#_Toc10674764)

[1.2. Mục đích đề tài 2](#_Toc10674765)

[1.3. Yêu cầu đề tài 2](#_Toc10674766)

[Chương 2: CƠ SỞ LÝ THUYẾT 3](#_Toc10674767)

[2.1. Hệ gợi ý 3](#_Toc10674768)

[2.1.1. Dự đoán trong hệ gợi ý truyền thống 3](#_Toc10674769)

[2.1.2. Dự đoán trong hệ gợi ý có ngữ cảnh 4](#_Toc10674770)

[2.2. Ngữ cảnh 5](#_Toc10674771)

[2.2.1. Nhận biết 5](#_Toc10674772)

[2.2.2. Phương pháp 6](#_Toc10674773)

[2.3. Các phương pháp truyền thống 7](#_Toc10674774)

[2.3.1. Gợi ý dựa trên nội dung 8](#_Toc10674775)

[2.3.2. Gợi ý dựa trên cộng tác 9](#_Toc10674776)

[2.3.3. Gợi ý dựa trên cách tiếp cận kết hợp 10](#_Toc10674777)

[2.4. Các phương pháp tiếp cận gợi ý hiện đại 10](#_Toc10674778)

[2.4.1. Phương pháp nhận biết ngữ cảnh 10](#_Toc10674779)

[2.4.2. Phương pháp dựa trên ngữ nghĩa 11](#_Toc10674780)

[2.5. Các phương pháp tiếp cận phân rã ma trận 11](#_Toc10674781)

[2.5.1. Phân rã ma trận 11](#_Toc10674782)

[2.5.2. Phân rã ma trận nhận biết ngữ cảnh 12](#_Toc10674783)

[2.5.3. Phân rã ma trận nhận biết ngữ cảnh dựa trên tương quan 13](#_Toc10674784)

[2.6. Các phương pháp tiếp cận theo tương quan ngữ cảnh 14](#_Toc10674785)

[2.6.1. Bối cảnh độc lập tương tự nhau 14](#_Toc10674786)

[2.6.2. Bối cảnh tiềm ẩn tương tự nhau 15](#_Toc10674787)

[2.6.3. Bối cảnh đa chiều tương tự nhau 16](#_Toc10674788)

[2.7. Các phương pháp đánh giá 17](#_Toc10674789)

[2.7.1. Recall-At-k (Rec) 19](#_Toc10674790)

[2.7.2. Precision-At-k (Pre) 19](#_Toc10674791)

[2.7.3. Độ chính xác trung bình (MAP) 19](#_Toc10674792)

[2.7.4. Khoảng dưới đường cong ROC (AUC) 20](#_Toc10674793)

[2.7.5. Xếp hạng đối ứng trung bình (MRR) 20](#_Toc10674794)

[2.7.6. Chuẩn hóa chiết khấu tích lũy (NDCG) 20](#_Toc10674795)

[Chương 3: THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG 22](#_Toc10674796)

[3.1. Phân tích chức năng hệ thống 22](#_Toc10674797)

[3.2. Các sơ đồ hệ thống 23](#_Toc10674798)

[3.3. Thiết kế hệ thống 24](#_Toc10674799)

[3.3.1. Xây dựng các lớp nền 25](#_Toc10674800)

[3.4. Thuật toán 27](#_Toc10674801)

[3.4.1. Lớp mẫu của mọi thuật toán: 27](#_Toc10674802)

[3.4.2. Các thuật toán trung bình 27](#_Toc10674803)

[3.4.3. Các thuật toán lọc cộng tác 28](#_Toc10674804)

[3.4.4. Thuật toán xếp hạng 28](#_Toc10674805)

[3.4.5. Thuật toán biến dổi 29](#_Toc10674806)

[3.4.6. Thuật toán thích ứng không phụ thuộc 29](#_Toc10674807)

[3.4.7. Thuật toán thích ứng phụ thuộc 29](#_Toc10674808)

[Chương 4: KẾT QUẢ THỰC NGHIỆM 31](#_Toc10674809)

[4.1. Quy trình sinh dữ liệu 31](#_Toc10674810)

[4.1.1. Dữ liệu thực tế 31](#_Toc10674811)

[4.1.2. Dữ liệu được chuẩn bị 32](#_Toc10674812)

[4.1.3. Dữ liệu được biến đổi sang dạng nhị phân 32](#_Toc10674813)

[4.1.4. Kết quả thu được sau khi chạy thuật toán 32](#_Toc10674814)

[4.2. Triển khai hệ thống 33](#_Toc10674815)

[4.2.1. Yêu cầu hệ thống 33](#_Toc10674816)

[4.2.2. Tùy chọn tệp cấu hình 33](#_Toc10674817)

[4.2.3. Thực thi thuật toán 33](#_Toc10674818)

[4.2.4. Điều chỉnh thông số cho hệ 34](#_Toc10674819)

[4.2.5. So sánh giữa kết quả 34](#_Toc10674820)

[4.2.6. Lấy các kết quả gợi ý 35](#_Toc10674821)

[4.3. Hướng đẫn sử dụng 36](#_Toc10674822)

[4.3.1. Lần đầu khởi chạy ứng dụng 36](#_Toc10674823)

[4.3.2. Lần khởi chạy sau đó 37](#_Toc10674824)

[4.4. Thực nghiệm 37](#_Toc10674825)

[4.4.1. Chọn bộ dữ liệu 37](#_Toc10674826)

[4.4.2. Thực thi thuật toán 37](#_Toc10674827)

[4.4.3. Xem lịch sử 39](#_Toc10674828)

[4.4.4. Xem biểu đồ 40](#_Toc10674829)

[4.4.5. Nhận xét 40](#_Toc10674830)

[Kết luận 41](#_Toc10674831)

**DANH SÁCH HÌNH VẼ**

[Hình 2.1: Ma trận xếp hạng hai chiều 3](#_Toc10673266)

[Hình 2.2: Ma trận xếp hạng 3 chiều thêm chiều ngữ cảnh 4](#_Toc10673267)

[Hình 2.3: Chiến thuật sử dụng ngữ cảnh 6](#_Toc10673268)

[Hình 2.4: Các thành phần cơ bản của tiến trình gợi ý truyền thống 7](#_Toc10673269)

[Hình 2.5: Lịch sử đánh giá của người dùng 8](#_Toc10673270)

[Hình 2.6: Danh sách các bộ phim 8](#_Toc10673271)

[Hình 2.7: Ví dụ hệ gợi ý hợp tác 9](#_Toc10673272)

[Hình 2.8: Ví dụ về hệ gợi ý hợp tác trên wiki 9](#_Toc10673273)

[Hình 2.9: Cold-start problem 10](#_Toc10673274)

[Hình 2.10: Ví dụ của hệ tọa độ đa chiều 16](#_Toc10673275)

[Hình 3.1: Sơ đồ phân cấp chức năng của ứng dụng 23](#_Toc10673276)

[Hình 3.2: Sơ đồ luồng dữ liệu của ứng dụng 23](#_Toc10673277)

[Hình 3.3: Sơ đồ dữ liệu thuật toán 24](#_Toc10673278)

[Hình 3.4: Các nhóm thuật toán 27](#_Toc10673279)

[Hình 4.1: Dữ liệu thu được sau khi chạy thuật toán CAMF\_C 32](#_Toc10673280)

[Hình 4.2: Nơi nhập tệp cấu hình và tệp dữ liệu 33](#_Toc10673281)

[Hình 4.3: Các nhóm thuật toán 34](#_Toc10673282)

[Hình 4.4: Cấu hình chung cho toàn bộ thuật toán 34](#_Toc10673283)

[Hình 4.5: Cấu hình riêng cho thuật toán 34](#_Toc10673284)

[Hình 4.6: Lịch sử thực thi trên từng thuật toán 34](#_Toc10673285)

[Hình 4.7: Biểu đồ so sánh kết quả các thuật toán đã chạy 35](#_Toc10673286)

[Hình 4.8: Tệp kết quả của quá trình chạy thuật toán CAMF\_C và CAMF\_ICS 35](#_Toc10673287)

[Hình 4.9: Tệp gợi ý trong một thuật toán bất kỳ 36](#_Toc10673288)

[Hình 4.10: Tập tin ban đầu 36](#_Toc10673289)

[Hình 4.11: Tập tin sau khi khởi chạy lần đầu 36](#_Toc10673290)

[Hình 4.12: Chọn cấu hình và chọn tệp dữ liệu vào 37](#_Toc10673291)

[Hình 4.13: Thực thi thuật toán trung bình toàn cục 37](#_Toc10673292)

[Hình 4.14: Thực thi thuật toán trung bình ngữ cảnh 37](#_Toc10673293)

[Hình 4.15: Thực thi thuật toán User KNN 38](#_Toc10673294)

[Hình 4.16: Thực thi thuật toán Slope One 38](#_Toc10673295)

[Hình 4.17: Thực thi thuật toán SLIM 38](#_Toc10673296)

[Hình 4.18: Thực thi thuật toán LRMF 38](#_Toc10673297)

[Hình 4.19: Thực thi thuật toán Item Splitting kết hợp Biased MF 38](#_Toc10673298)

[Hình 4.20: Thực thi thuật toán Item User Spliting kết hợp Biased MF 38](#_Toc10673299)

[Hình 4.21: Thực thi thuật toán CAMF Context 38](#_Toc10673300)

[Hình 4.22: Thực thi thuật toán CSLIM Context User 39](#_Toc10673301)

[Hình 4.23: Thực thi thuật toán GCSLIM ICS 39](#_Toc10673302)

[Hình 4.24: Thực thi thuật toán GCSLIM MCS 39](#_Toc10673303)

[Hình 4.25: Thực thi thuật toán Matrix Factorization 39](#_Toc10673304)

[Hình 4.26: Lịch sử thực thi thuật toán 39](#_Toc10673305)

[Hình 4.27: Biểu đồ kết quả tốt nhất của mỗi thuật toán 40](#_Toc10673306)

**DANH SÁCH BẢNG**

[Bảng 2.1: Các thuộc tính của các nguồn ngữ cảnh 5](#_Toc10673248)

[Bảng 2.2: Phân loại các chiều của ngữ cảnh 5](#_Toc10673249)

[Bảng 2.3: Xếp hạng phim trong ngữ cảnh 12](#_Toc10673250)

[Bảng 2.4: Ví dụ của một ma trận tương quan 15](#_Toc10673251)

[Bảng 2.5: Bảng dummy\_truth 17](#_Toc10673252)

[Bảng 2.6: Bảng dummy\_rec 18](#_Toc10673253)

[Bảng 2.7: Kết quả dự đoán 18](#_Toc10673254)

[Bảng 4.1: Bảng người dùng 31](#_Toc10673255)

[Bảng 4.2: Bảng địa điểm 31](#_Toc10673256)

[Bảng 4.3: Bảng đánh giá của người dùng cho địa điểm 31](#_Toc10673257)

[Bảng 4.4: Dữ liệu chuẩn bị để đưa vào hệ thống 32](#_Toc10673258)

[Bảng 4.5: Dữ liệu đã được hệ thống biến đổi sang dạng nhịn phân 32](#_Toc10673259)

**DANH SÁCH CÁC CHỮ VIẾT TẮT**

RS: Recommender Systems

CARS: Context Aware Recommender Systems

CAMF: Context Aware Matrix Factorization

CSLIM: Contextual Sparse Linear Method

TF: Tensor Factorization

MF: Matrix Factorization

ICS: Independent Context Similarity

LCS: Latent Context Similarity

MCS: Multidimensional Context Similarity

**MỞ ĐẦU**

Hiện nay, việc mua bán online đang diễn ra vô cùng mạnh mẽ. Mỗi website có hàng ngàn, chục ngàn đến trăm ngàn sản phẩm với nhiều loại và công dụng khác nhau. Tương tự như vậy thì các website hướng dẫn du lịch cũng như vậy, khách hàng của họ chỉ ghé qua trong thời gian ngắn nhưng họ lại muốn thấy ngay những gì họ cần, nếu không họ sẽ cảm thấy thật khó khăn trong việc tìm cái mình cần ở một website. Từ đó, việc đưa ra gợi ý một cách vô cùng hợp lý là điều mà website nào cũng cần. Ngay khi khách hàng truy cập vào website và thực hiện một vài thao tác đầu tiên, việc đưa ra ngay gợi ý cho khách hàng mặt hàng tiếp theo là vô cùng cần thiết. Hiểu được khách hàng là một điều không phải ai cũng làm được nhất là đối với lập trình viên nhưng khách hàng lại chính là người có thể hiểu rõ nhau nhất. Chúng ta hoàn toàn có thể tìm điểm chung giữa họ và dự đoán cho những người có cùng điểm chung là họ cũng sẽ thích những mặt hàng mà những người có chung sở thích với họ cũng thích. Từ đó mà việc có một hệ gợi ý là một điều vô cùng quan trọng.

Với dữ liệu lịch sử khách hàng và các yếu tố khách quan, hệ gợi ý sẽ đưa ra các mặt hàng tiếp theo mà dự đoán rằng khách hàng sẽ quan tâm đến nó. Và đưa các gợi ý đó lên đầu trang để thu hút khác hàng về website của mình. Mục tiêu của đề tài là giúp cho bên nhà cung cấp dịch vụ có thể tìm ra hệ gợi ý phù hợp nhất với dữ liệu vào (lịch sử khách hàng) của mình và từ đó đưa ra các gợi ý phù hợp cho các khách hàng trong tương lai. Hệ gợi ý phù hợp bao gồm thuật toán phù hợp, thông số cài đặt phù hợp, thời gian chạy phù hợp và kết quả ra với xác suất cao nhất có thể của hệ.

Hệ thống nhắm đến mọi website có hình thức dịch vụ mua bán và giới thiệu sản phẩm cho người dùng. Để người dùng nhận được gợi ý hợp lý hay chính nhà cung cấp có thể tìm được người dùng thích hợp để giới thiệu. Và cũng như mọi hệ gợi ý khác, gợi ý không chỉ quan trọng mà còn cần phải đúng và đủ cho từng người dùng vì vậy mà tìm ra được hệ phù hợp là việc không phải dễ và cũng không dễ để áp dụng. Từ đó mà việc xây dựng một ứng dụng giúp bên thứ ba dễ dàng giải quyết bài toán đau đầu này là điều cần thiết.

# TỔNG QUAN ĐỀ TÀI

* 1. Giới thiệu đề tài

Hiện nay, việc lựa chọn nơi du lịch thích hợp theo sở thích, giá tiền và thời tiết (có thể xem như là thời điểm) thường sẽ tốn thời gian, ngoài ra có một số người không nhận ra rằng có 1 số địa điểm mà họ cũng có thể thích (địa điểm tương tự) và nhờ các phương pháp gợi ý mà các website có thể dễ dàng phán đoán nơi nào là lựa chọn tốt nhất cho khách hàng và thêm nữa là những nơi khách hàng có thể đi cùng lúc với nơi mình chọn. Nhưng không phải website nào cũng có hệ gợi ý và lựa chọn hệ gợi ý tốt nhất, có thể vì họ không có sự lựa chọn và cũng không dễ dàng để lựa chọn hệ tốt nhất cho mình. Vì vậy, đề tài mong giúp đỡ các website mua bán và quảng cáo có thể dễ dàng áp dụng và đưa ra các gợi ý thích hợp cho người dùng.

* 1. Mục đích đề tài

Với mục đích ban đầu là giúp cho người dùng (những người cần tìm ra hệ gợi ý hợp lí) dễ dàng đánh giá các thuật toán gợi ý và tìm ra thuật toán tốt nhất, ứng dụng hệ gợi ý hướng ngữ cảnh người dùng ra đời với hi vọng sẽ giúp người dùng có thể thử nghiệm từng thuật toán, đánh giá so sánh tạo ra được gợi ý cho hệ thống của mình.

* 1. Yêu cầu đề tài

Với dữ liệu sẵn có từ người, phần mềm sẽ tạo ra 1 giao diện thao tác để người dùng có cơ hội thử và đánh giá từng phương pháp gợi ý trong hệ gợi ý và đặc biệt có thể thử các phương pháp gợi ý mới nhất trong đó có hệ gợi ý theo hướng ngữ cảnh. Sau cùng, người dùng có thể tùy chỉnh thông số phù hợp để đạt được kết quả gợi ý tốt nhất và trích ra gợi ý dựa trên dữ liệu đầu vào thành gợi ý cho hệ thống của mình.

Yêu cầu của hệ thống:

* Đánh giá của người dùng về một sản phẩm nào đó (một chuyến đi) và các ngữ cảnh xunh quanh và từ chính người dùng.
* Thuật toán tìm ra gợi ý và các thông số cài đặt.

Kết quả của hệ thống:

* Kết quả độ chính xác của thuật toán xét trên nhiều thuật toán đánh giá khác nhau.
* Một loạt danh sách về các gợi ý mà hệ thống đưa ra cho từng người dùng thông qua bộ dữ liệu ban đầu.

# CƠ SỞ LÝ THUYẾT

* 1. Hệ gợi ý

Hệ gợi ý (Recommender system - RS) là lớp con của hệ lọc thông tin, là những công cụ phần mềm và kỹ thuật cung cấp đề xuất các sản phẩm sẽ được sử dụng cho người dùng. Hệ gợi ý cố gắng dự đoán sự "xếp hạng" hay "độ ưu tiên" mà người dùng sẽ dành cho các sản phẩm nào đó mà họ chưa từng tương tác.

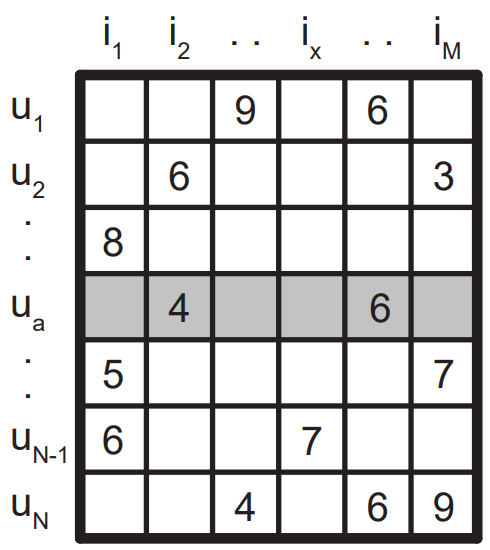
Hệ gợi ý là một cách thuận tiện để giảm thiểu lượng thông tin thừa trong nhiều lĩnh vực ứng dụng bằng cách điều chỉnh các khuyến nghị cho người dùng theo sở thích cá nhân. Bằng cách thu thập các tương tác của người dùng với hệ thống và tìm ra điểm tương đồng giữa người dùng hoặc sản phẩm, một thuật toán sẽ cố gắng đưa ra các sản phẩm cho người dùng mà nó dự đoán rằng họ sẽ hứng thú với chúng.

Ngoài ra còn có một số loại hệ thống gợi ý xuất hiện trong những năm gần đây, chẳng hạn như hệ thống gợi ý hướng ngữ cảnh (Context Aware Recommender Systems – CARS) và gợi ý bối cảnh (Context Recommender). CARS nổi lên vượt ra ngoài sở thích người dùng và cũng tính đến tình huống bối cảnh của người dùng trong việc tạo ra đề xuất. CARS đã chứng minh hiệu quả trong một số lĩnh vực, chẳng hạn như du lịch, âm nhạc, phim và nhà hàng.

* + 1. Dự đoán trong hệ gợi ý truyền thống

Công thức tiêu chuẩn để giải quyết vấn đề gợi ý bắt đầu với một ma trận xếp hạng hai chiều, được tạo nên từ người dùng và sản phẩm:

R: Users × Items → Ratings.

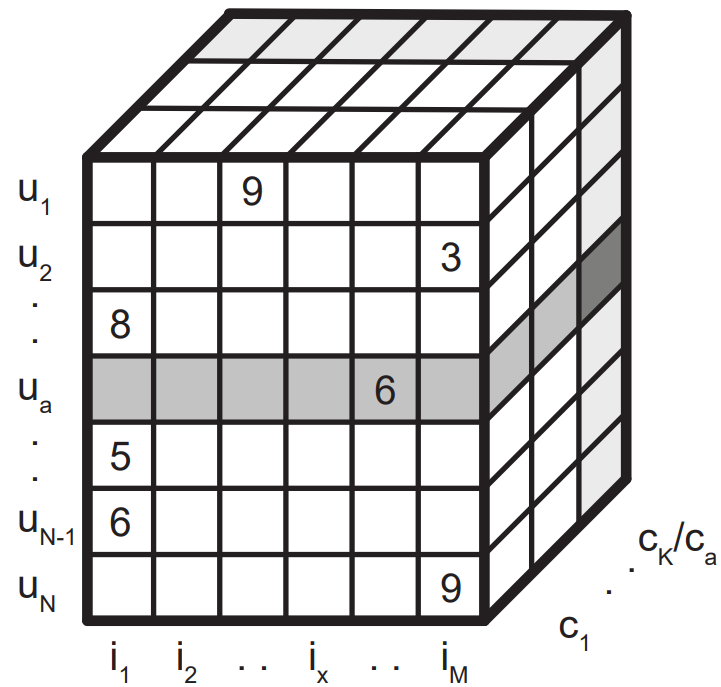


Hình 2.1: Ma trận xếp hạng hai chiều

* + 1. Dự đoán trong hệ gợi ý có ngữ cảnh

Chìa khóa nằm bên trong của các hệ thống gợi ý nhận biết ngữ cảnh là các ưa thích của người dùng cho các sản phẩm có thể bị bối cảnh tác dụng một phần lên các sản phẩm đó. Việc kết hợp bối cảnh yêu cầu cần ước tính sở thích của người dùng bằng cách sử dụng hàm xếp hạng đa chiều (a multidimensional rating function):

R: Users × Items × Contexts → Ratings



Hình 2.2: Ma trận xếp hạng 3 chiều thêm chiều ngữ cảnh

* 1. Ngữ cảnh

Ngữ cảnh là "bất kỳ thông tin nào mà có thể sử dụng được để đặc tả một tình huống của một thực thể. Một thực thể là một người, một nơi hay một số đối tượng được xem là có liên quan đến tương tác giữa người dùng và ứng dụng, bao gồm cả chính người dùng và ứng dụng đó".

Trong một tương tác giữa người dùng và ứng dụng có rất nhiều thông tin có thể được coi là ngữ cảnh. Ta gọi số loại thông tin ngữ cảnh là "chiều ngữ cảnh" (contextual dimension), mỗi loại thông tin ngữ cảnh ứng với một miền giá trị gọi là "tình trạng ngữ cảnh" (contextual condition) thì tập hợp các thông tin ngữ cảnh trong một tương tác được gọi là một "tình huống ngữ cảnh" (contextual situation).

* + 1. Nhận biết
       1. Các thuộc tính của ngữ cảnh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kiểu** | **Nguồn** | **Tính lâu dài** | **Chất lượng** | **Nguyên nhân lỗi** |
| Cảm biến | Cảm biến vật lý và logic | Ngắn | Tùy vào chất lượng hệ thống | Lỗi thiết bị cảm biến, mất mạng, ... |
| Đặc tả | Do người dùng đặc tả trực tiếp hoặc gián tiếp | Dài | Không đảm bảo tính chính xác | Do cảm xúc, khả năng mô tả, lỗi ghi chép, ... |

Bảng 2.1: Các thuộc tính của các nguồn ngữ cảnh

* + - 1. Phân loại ngữ cảnh

|  |  |
| --- | --- |
| **Ngữ cảnh** | **Tham số ngữ cảnh** |
| Ngữ cảnh tĩnh của người dùng | Tên, tuổi, thói quen, sở thích, ... |
| Ngữ cảnh động của người dùng | Hành động, cảm xúc, trạng thái, ... |
| Ngữ cảnh môi trường | Thời tiết, thời điểm, địa điểm, ... |

Bảng 2.2: Phân loại các chiều của ngữ cảnh

* + 1. Phương pháp

Trong hệ thống gợi ý nhận biết ngữ cảnh, ngữ cảnh luôn luôn được xem là “bất cứ thông tin nào có thể được sử dụng để đặc trưng hóa tình hình của một thực thể”, bao gồm *thời gian* và *người đồng hành* có thể là hai điều có ảnh hưởng đến ngữ cảnh, bởi vì sở thích của người dùng có thể thay đổi khi hai ngữ cảnh kia thay đổi.

Giới thiệu về phân loại hai phần của thông tin theo ngữ cảnh. Phân loại của nó dựa trên hai sự cân nhắc: những điều mà hệ gợi ý nhận biết về một yếu tố ngữ cảnh (contextual factor) và cách mà yếu tố ngữ cảnh đó thay đổi theo thời gian. Hiểu biết của hệ thống (system knowledge) có thể được chia thành: *quan sát được hoàn toàn* (fully observable), *quan sát được một phần* (partially observable) và *không quan sát được* (unobservable). Khía cạnh thời gian của một yếu tố có thể chia thành *tĩnh* (static) và *động* (dynamic).

Một vài thuật toán gợi ý nhận biết ngữ cảnh đã được phát triền trong vài thập kỷ trước. Điển hình là, ngữ cảnh có thể áp dụng trong ba chiến thuật cơ bản sử dụng gợi ý: *lọc trước* (pre-filtering), *lọc sau* (post-filtering) và *mô hình ngữ cảnh* (contextual modeling). Kỹ thuật pre-filtering, như là cách tiếp cận phân tách nhận biết ngữ cảnh (Context-Aware Splitting Approaches – CASA), áp dụng ngữ cảnh làm bộ lọc trước để lọc ra các hồ sơ xếp hạng không liên quan. Post-filtering, mặc khác, áp dụng ngữ cảnh như bộ lọc cho hệ gợi ý sau quá trình ra gợi ý. Trong contextual modeling, các mô hình dự đoán được học cách sử dụng toàn bộ ngữ cảnh. Hầu hết chúng vẫn còn hoạt động, như là phân rã ma trận nhận diện ngữ cảnh (CAMF), phân rã nhân tử (TF) và phương pháp tuyến tính ngữ cảnh thưa thớt (CSLIM).

C:\Users\sihc\Desktop\Sơ đồ CARS.png

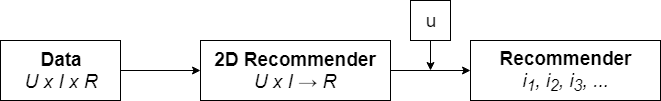
Hình 2.3: Chiến thuật sử dụng ngữ cảnh

* 1. Các phương pháp truyền thống

Gọi U(users) là tập người dùng, I(items) là tập các sản phẩm có thể được gợi ý. Để dự đoán xếp hạng (hay tính tiện ích) của sản phẩm i đối với người dùng u thì người ta đưa ra hàm xếp hạng (rating) r:

U x I → R

Trong đó: R(recommendations) là tập các giá trị xếp hạng được thứ tự toàn phần (ví dụ: số nguyên dương hoặc số thực trong tập xác định). Mô hình này còn được gọi với tên gọi là mô hình dự đoán 2 chiều (two-dimensional recommendation framework).



Hình 2.4: Các thành phần cơ bản của tiến trình gợi ý truyền thống

Với mỗi người dùng u ∈ U, chúng ta có thể chọn được sản phẩm i ∈ I sao cho hàm xếp hạng của người dùng u đối với item i là lớn nhất.

∀𝑢 ∈ 𝑈, 𝑖 ′ 𝑢 = arg max 𝑟̂(𝑢, 𝑖)

𝑖∈𝐼

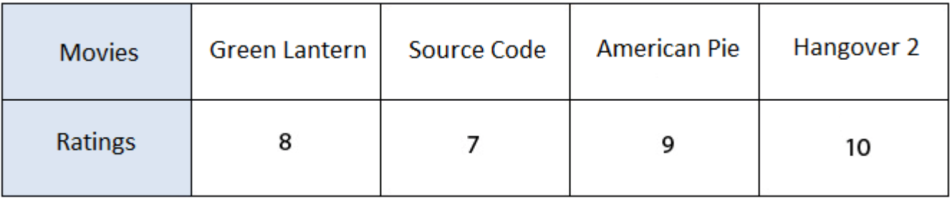
Có rất nhiều phương pháp tiếp cận trong hệ gợi ý, tuy nhiên có thể chia thành 3 nhóm kỹ thuật chính như sau:

* Gợi ý dựa trên nội dung (content-based filtering)
* Gợi ý dựa trên cộng tác (collaborative filtering - CF)
* Gợi ý dựa trên cách tiếp cận kết hợp (hybrid approach)
  + 1. Gợi ý dựa trên nội dung

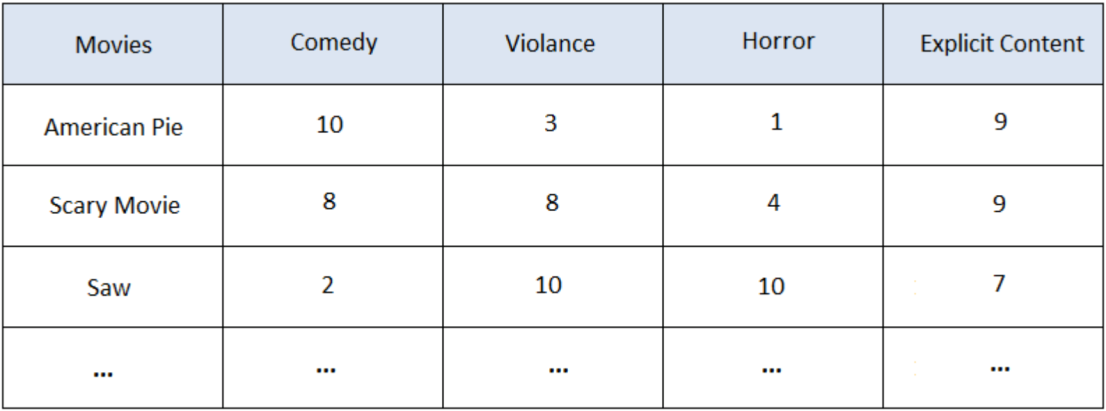
Hệ thống gợi ý dựa trên nội dung làm việc với hồ sơ của người dùng được tạo ngay từ lúc ban đầu. Một hồ sơ có thông tin về một người dùng và “khẩu vị”(taste) của người đó. Khẩu vị được dựa trên cách người dùng đánh giá các sản phẩm. Thông thường, khi tạo một hồ sơ, hệ thống gợi ý tạo một khảo sát, để lấy thông tin khởi đầu về một người dùng để tránh những vấn đề với người dùng mới.

Trong quá trình đưa ra gợi ý, hệ thống so sánh những sản phẩm đã được đánh giá tích cực bởi người dùng với những sản phẩm người đó không đánh giá và tìm kiếm điểm tương đồng. Những sản phẩm mà gần như tương đồng với cái được đánh giá tích cực, sẽ được gợi ý cho người dùng.

Hình 2.4 cho thấy một ví dụ cho một hồ sơ của người dùng với những bộ phim mà người đó đã xem và người đó đã đánh giá. Hình 2.5 cho thấy danh sách những bộ phim và tính chất của chúng. Một hệ thống gợi ý dựa trên nội dung sẽ tìm ra những bộ phim từ danh sách (hình 2.5) và tìm kiếm những điểm tương đồng. Những bộ phim tương đồng sẽ được gợi ý cho người dùng. Trong ví dụ này chúng ta có thể thấy rằng bộ phim “Scary Movie” tương tự với bộ phim “American Pie” mà người xe đã đánh giá tích cực vậy nên nó sẽ được gợi ý cho người dùng này.



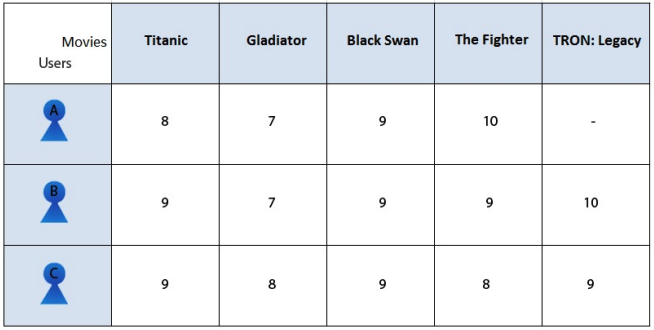
Hình 2.5: Lịch sử đánh giá của người dùng



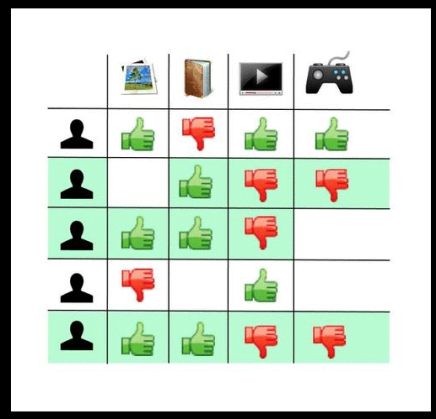
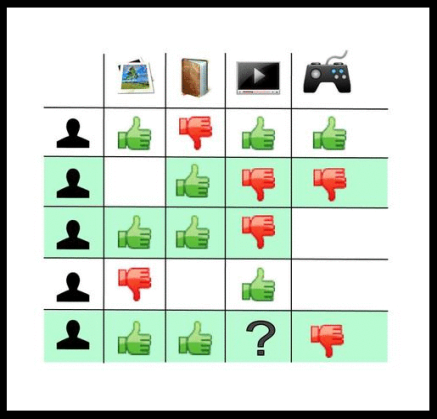
Hình 2.6: Danh sách các bộ phim

* + 1. Gợi ý dựa trên cộng tác

Gợi ý cộng tác trở thành một trong hầu hết các công nghệ của hệ thống gợi ý được nghiên cứu từ khi cách tiếp cận này được đề cập đến và mô tả bởi Paul Resnick và Hal Varian vào năm 1007. Ý tưởng về lọc cộng tác là tìm kiếm người dùng trong một cộng đồng chia sẻ đánh giá. Nếu hai người dùng có cùng hoặc có cùng hầu hết đánh giá về những sản phẩm chung, thì họ có cùng “khẩu vị”. Giống như những người dùng tạo nên một nhóm hoặc gọi là những người hàng xóm. Một người dùng nhận được gợi ý đến những sản phẩm mà họ chưa đánh giá trước đó, nhưng đã được đánh giá tích cực bởi hàng xóm của họ. Hình 2.7 cho thấy rằng cả 3 người dùng đều đánh giá các bộ phim tích cực với cùng số điểm. Có nghĩa là họ có cùng khẩu vị và được nhóm vào 1 nhóm hàng xóm. Người dùng A chưa đánh giá bộ phim “TRON: Legacy”, mà có lẽ người đó chưa xem. Vì bộ phim đã được đánh giá cao bởi những hàng xóm khác, anh ấy sẽ được gợi ý bộ phim đó. Trái ngược với với hệ gợi ý đơn giản nơi mà gợi ý dựa trên sản phẩm đã được đánh giá nhiều và sản phẩm phổ biến nhất, gợi ý cộng tác quan tâm đến khẩu vị của người dùng. Khẩu vị được đánh giá là không đổi hoặc đổi rất ít.



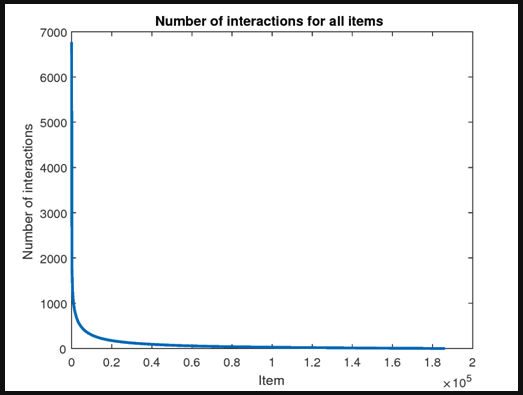
Hình 2.7: Ví dụ hệ gợi ý hợp tác



Hình 2.8: Ví dụ về hệ gợi ý hợp tác trên wiki

* + 1. Gợi ý dựa trên cách tiếp cận kết hợp

Để kết có kết quả tốt hơn, vài hệ thống gợi ý kết hợp các công nghệ khác nhau của cách tiếp cận cộng tác và cách tiếp cận dựa trên nội dung. Sử dụng cách tiếp cận kết hợp, ta có thể bỏ qua giới hạn và các vấn đề của một hệ thống yếu, như là “cold-start problem”.



Hình 2.9: Cold-start problem

Sự kết hợp của các phương pháp có thể thực hiện theo các cách khác nhau:

* Thực hiện đầy đủ các thuật toán và gộp kết quả.
* Sử dụng một số quy tắc lọc dựa trên nội dung trong phương pháp hợp tác
* Sử dụng một số quy tắc lọc cộng tác theo cách tiếp cận lọc nội dung
* Tạo một hệ thống đề xuất thống nhất, kết hợp cả hai cách tiếp cận
  1. Các phương pháp tiếp cận gợi ý hiện đại

Trong vài thập kỷ đã qua, một số thuật toán gợi ý nhận diện ngữ cảnh đã được phát triển, như là:

* Mô hình bối cảnh khác biệt (Differential Context Modeling): kết hợp ngữ cảnh như là những bộ lọc trong lọc cộng tác (Collaborative Filtering).
* Phân rã ma trận nhận diện ngữ cảnh (Context-Aware Matrix Factorization - CAMF): sử dụng ngữ cảnh trong phân rã ma trận.
* Phân rã nhân tử (Tensor Factorization - TF): trực tiếp xem xét ngữ cảnh là chiều cá nhân trong không gian xếp hạng đa chiều.
  + 1. Phương pháp nhận biết ngữ cảnh

Ngữ cảnh là thông tin về môi trường của một người dùng và tình huống cụ thể của người đó. Những chi tiết như vậy có thể đáng kể hơn nhiều luật trong gợi ý hơn việc đánh giá các sản phẩm, vì việc đánh giá, một mình nó không có thông tin chi tiết về hoàn cảnh bên dưới mà nó được nhận bởi người dùng. Vài gợi ý có thể thích hợp hơn với người dùng trong buổi chiều và không hợp với sở thích của người đó vào buổi sáng, và người dùng muốn làm một việc khi trời lạnh nhưng lại không làm khi trời nóng. Hệ gợi ý chú ý và sử dụng thông tin đó trong việc đưa ra các khuyến nghị được gọi là hệ thống gợi ý nhận biết ngữ cảnh.

* + 1. Phương pháp dựa trên ngữ nghĩa

Hầu hết thông tin của sản phẩm, người dùng trong hệ gợi ý và phần còn lại của trang web được thể hiện ra trong trang web ở dạng văn bản. Sử dụng chủ đề và từ khóa mà không có bất kỳ ý nghĩa ngữ nghĩa nào sẽ không cải thiện tính chính xác của các đề xuất trong mọi trường hợp, vì một số từ khóa có thể là từ đồng âm. Đó là lý do tại sao việc hiểu và cấu trúc cho văn bản là một khuyến nghị phần rất quan trọng. Phương pháp khai thác văn bản truyền thống dựa trên phân tích từ vựng và cú pháp cho thấy các mô tả có thể hiểu được bởi người dùng chứ không phải máy tính hoặc hệ thống đề xuất. Đó là một lý do tạo ra các kỹ thuật xác định ý nghĩa văn bản mới dựa trên phân tích ngữ nghĩa. Các hệ thống đề xuất với các kỹ thuật như vậy được gọi là các hệ thống đề xuất ngữ nghĩa.

* 1. Các phương pháp tiếp cận phân rã ma trận
     1. Phân rã ma trận

Matrix factorization

Phân rã ma trận là một trong các thuật toán gợi ý hiệu quả nhất trong hệ thống gợi ý truyền thống. Đơn giản, cả người dùng và sản phẩm đều được biểu diễn dưới dạng vector. Ví dụ, được sử dụng để biểu thị vector người dùng, như là một vector sản phẩm. Các giá trị trong những vector đó có thể được biểu diễn bởi phương trình 1.

(1)

Đặc biệt hơn, trọng số của được sử dụng để biểu thị số lượng người dùng thích các nhân tố tiềm ẩn(latent factor) đó, và trọng số thể hiện cách mà những sản phẩm này có được các nhân tố tiềm ẩn đó. Do đó, tích vô hướng của hai vector này được sử dụng để biểu lộ mức độ người dùng thích sản phẩm này

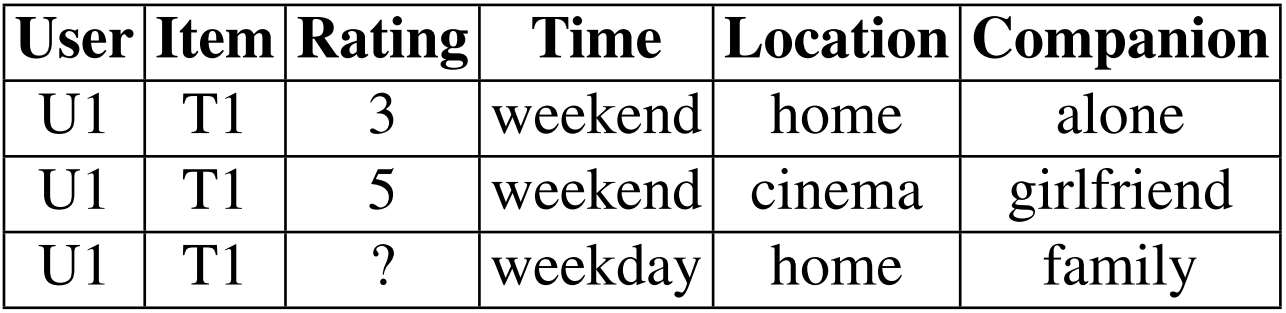
(2)

Bên cạnh đó, người dùng và thiên vị đánh giá sản phẩm cũng được thêm vào, như đã biểu diễn ở công thức số 2, nơi mà biểu lộ đánh giá trung bình toàn bộ trong bộ dữ liệu, và biểu diễn người dùng và thiên vị cho sản phẩm tương ứng.

* + 1. Phân rã ma trận nhận biết ngữ cảnh

Context-aware Matrix Factorization

Xét ví dụ gợi ý phim trong bảng 2.3. Ta thấy người dùng U1, một sản phẩm T1, và ba chiều ngữ cảnh - *Thời gian* (ngày trong tuần và cuối quẩn), *Vị trí* (ở nhà và rạp phim) và *Đồng hành* (một mình, người yêu, gia đình) . Trong thảo luận sau, chúng ta sử dụng *chiều ngữ cảnh* để chứng tỏ giá trị của ngữ cảnh, ví dụ như *Vị trí*. Thuật ngữ điều kiện ngữ cảnh đề cập đến một giá trị cụ thể trong một chiều, ví dụ như “nhà” và “rạp phim” là hai trạng thái của “Vị trí”. Vì lẽ trên, một ngữ cảnh hay một tình huống có ngữ cảnh là một tập hợp của các trạng thái ngữ cảnh.



Bảng 2.3: Xếp hạng phim trong ngữ cảnh

Thêm nữa, chúng ta sử dụng và để chứng tỏ hai tình huống ngữ cảnh khác nhau. Nói cách khác, được hợp thành bởi một tập của các điều kiện ngữ cảnh. Và chúng ta sử dụng để chứng tỏ trạng thái ngữ cảnh thứ l () trong ngữ cảnh . Ví dụ, giả sử = {cuối tuần, ở nhà, một mình}, là “nhà”. Trong những thảo luận dưới, chúng ta tiếp tục sử dụng những ký hiệu và biểu tượng này để mô tả không gian ngữ cảnh và các điều kiện tương trong thuật toán hay phương trình.

Hầu hết các thuật toán gợi ý nhận biết ngữ cảnh được xây dựng bằng cách kết hợp ngữ cảnh và trong các thuật toán gợi ý truyền thống, đặc biệt là những thuật toán dựa trên lọc cộng tác. Là một trong những thuật toán gợi ý phổ biến nhất, nó khá hợp lý cho việc sử dụng MF để xây dựng các thuật toán phân rã ma trận nhận biết ngữ cảnh.

Thuật toán CAMF đầu tiên được đề xướng và phát triển bởi Linas Baltrunas. Trong đó, phương trình dự đoán xếp hạng được biểu diễn trong Công thức 3.

(3)

Giả sử ta có tổng cộng L không gian ngữ cảnh, = {} được dùng để mô tả tình huống ngữ cảnh, với biểu thị tình trạng ngữ cảnh trong chiều ngữ cảnh thứ j (). Do đó, biểu thị độ lệch đánh giá theo ngữ cảnh (contextual rating deviation term) với sản phẩm i và trạng thái ngữ cảnh trong chiều thứ j ().

Một so sánh giữa công thức 2 và công thức 3 cho thấy rằng CAMF đơn giản thay thế thiên vị sản phẩm bới một công thức độ lệch đánh giá theo ngữ cảnh và nó giả định rằng thiên vị đánh giá theo ngữ cảnh là phụ thuộc vào sản phẩm, do đó, hướng tiếp cận này được đặt tên là CAMF\_CI (Context Item). Theo đó, độ lệch này cũng có thể được biểu diễn bị phụ thuộc vào người dùng, bằng cách thay thế bằng độ lệch đánh giá theo ngữ cảnh và hình thành CAMF\_CU (Context User). Thêm nữa, thuật toán CAMF\_C giả định rằng độ lệch đánh giá theo ngữ cảnh không phụ thuộc và người dùng hay sản phẩm.

Do đó, những biến số, như là người dùng và sản phẩm và vector sản phẩm, thiên vị người dùng và độ lệch đánh giá, có thể được học bằng phương pháp stochastic gradient descent (SGD) để giảm thiểu lỗi dự đoán xếp hạng.

* + 1. Phân rã ma trận nhận biết ngữ cảnh dựa trên tương quan

Correlation-Based Context-aware Matrix Factorization

Phương trình độ lệt đánh giá theo ngữ cảnh là một cách phổ biến và hiệu quả để xây dựng thuật toán gợi ý hướng ngữ cảnh. CARS hiện tại đã kết hợp độ lệch đánh giá theo ngữ cảnh vào phương pháp tuyến tính rời rạc (Sparse Linear Method - SLIM) và phát triển hướng ngữ cảnh cho SLIM (CSLIM) và CSLIM đã được chứng minh là vượt trội những thuật toán gợi ý hướng ngữ cảnh hiện tại, bao gồm cả CAMF đã được giới thiệu ở trên.

Như đã đề cập ở trên, tương quan ngữ cảnh là một cách thay thế để xây dựng thuật toán gợi ý hướng ngữ cảnh, ngoại trừ mô hình hóa sai lệch theo ngữ cảnh (modeling the contextual deviations). Tiếp theo, chúng ta khám phá thêm những cách để xây dựng CAMF dựa trên tương quan.

Giả định đằng sau khái niệm “ngữ cảnh tương quan” là: tăng thêm tính tương tự hay tương quan giữa hai ngữ cảnh, hai danh sách gợi ý cho cùng một người dùng cho hai tình huống tương tự cũng nên giống nhau. Trong phân rã ma trận, hàm dự đoán có thể được mô tả như phương trình 4.

(4)

thay cho tình huống có ngữ cảnh thiếu – giá trị trong không gian mẫu là trống (empty) hoặc không có sẵn (NA - Not Available), nghĩa là: . Do đó, hàm ước tính sự tương quan giữa và hình huống ngữ cảnh nơi mà ít nhất một tình huống ngữ cảnh không trống hoặc “NA”. Chú ý rằng công thức 3. Ngữ cảnh sai lệch xếp hạng theo ngữ cảnh có thể được xem như sai lệt từ tình huống ngữ cảnh thiếu với tình huống không thiếu ngữ cảnh.

(5)

Theo đó, các vector người dùng và sản phẩm, cũng như các mối tương quan ngữ cảnh có thệ được học dựa trên SGD bằng cách giảm thiểu các lỗi dự đoán xếp hạng, trong đó hàm mất mát (loss function) được biểu diễn bằng phương trình 5.

Thách thức còn lại là cách biểu diễn hay mô hình hóa hàm tương quan trong mô công thức số 4. Chúng ta sẽ sử dụng bốn chiến thuật: Bối cảnh độc lập tương tự nhau (Independent Context Similarity – ICS), Bối cảnh tiềm ẩn tương tự nhau (Latent Context Similarity – LCS), Bối cảnh đa chiều tương tự nhau (Multidimensional Context Similarity – MCS), Trọng số ngữ cảnh jaccard tương tự nhau (Weighted Jaccard Context Similarity – WJCS). Và những chiến thuật tiếp này có thể đưuọc sử dụng lại trong CAMF dựa trên tương quan. Do WJCS chỉ đếm chiều ngữ cảnh có cùng giá trị và sự tương quan được tính toán bằng điều đó giữa một ngữ cảnh và một ngữ cảnh trống, nó không áp dụng cho hàm tương quan được đề cập ở trên.

* 1. Các phương pháp tiếp cận theo tương quan ngữ cảnh
     1. Bối cảnh độc lập tương tự nhau

Independent Context Similarity (ICS)

Một ví của ma trận tương quan trong bảng 2.4. Với ICS, chúng ta chỉ tính toán sự tương quan giữa hai trạng thái dữ liệu khi chúng nằm trên cùng một chiều không gian ngữ cảnh, chúng ta không bao giờ tính toán tương quan giữa “Thời gian = Cuối tuần” và “Vị trí = Nhà”, bởi vì chúng từ hai chiều khác nhau. Mỗi cặp của chiều không quan được cho rằng không phụ thuộc vào nhau. Trong trường hợp này, sự tương quan giữa hai ngữ cảnh có thể được biểu diễn bằng tích của sự tương quan giữa các chiều khác nhau. Ví dụ, cho rằng là {Thời gian = Cuối tuần, Vị trí = Nhà}, và là {Thời gian = Giữa tuần, Vị trí = Rạp phim}, sự tương quan giữa và có thể được biểu diễn bằng sự tương quan của <Thời gian = Cuối tuần, Thời gian = Giữa tuần> nhân với tương quan của <Vị trí = Nhà, Vị trí = Rạp phim>, vì hai chiều chiều này được cho là không phụ thuộc.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | T = Cuối tuần | T = Giữa tuần | L = Nhà | L = Rạp phim |
| T = Cuối tuần | 1 | 0.54 | N/A | N/A |
| T = Giữa tuần | 0.54 | 1 | N/A | N/A |
| L = Nhà | N/A | N/A | 1 | 0.82 |
| L = Rạp phim | N/A | N/A | 0.82 | 1 |

Bảng 2.4: Ví dụ của một ma trận tương quan

Cho rằng có tất cả L chiều không gian ngữ cảnh, sự tương quan có thể được dự đoán bằng công thức 6, với được dùng để biểu đạt giá trị của trạng thái ngữ cảnh trong chiều không gian thứ *l* trong ngữ cảnh , và hàm “correlation” được dùng để biểu diễn sự tương quan giữa hai trạng thái ngữ cảnh, mà cũng là cái được học trong quá trình tối ưu. Nói cách khác, sự tương quan giữa hai ngữ cảnh được biểu diễn bằng tích của các tương quan không phụ thuộc giữa các trạng thái ngữ cảnh trong mỗi miền không gian.

(6)

Những tương giá trị tương quan này () có thể được học bởi quá trình tối ưu một cách hợp lý. Sự nguy rủi ro của biểu diễn này là vài thông tin có thể bị mất, nếu sự tương quan là không thật sự không phụ thuộc trong các chiều ngữ cảnh khác nhau. Ví dụ, nếu người dùng luôn luôn đi xem phim để xem phim tình cảm với gia đình của họ, thì “Vị trí” và “Đồng hành” có thể có ý nghĩa tương quan với kết quả.

* + 1. Bối cảnh tiềm ẩn tương tự nhau

Latent Context Similarity (LCS)

Như đã lưu ý trước đó, dữ liệu xếp hạng theo ngữ cảnh thường thưa thớt, vì việc nhiều người dùng xếp hạng các sản phẩm giống nhau trong nhiều tình huống theo ngữ cảnh là hơi bất thường. Điều này đặt ra một khó khăn khi bối cảnh mới gặp phải. Ví dụ , sự tương quan giữa một cặp ngữ cảnh mới <Thời gian = Cuối tuần, Thời gian = Kỳ nghỉ> có thể yêu cầu trong bộ dữ liệu thử, nhưng nó có thể khôn được học từ dữ liệu dạy vì vấn đề dữ liệu thưa thớt (sparsity problem). Nhưng, sự tương quan của hai cặp dữ liệu có sẵn <Thời gian = Giữa tuần, Thời gian = Cuối tuần> và <Thời gian = Giữa tuần, Thời gian = Kỳ nghỉ> có thể đã được học. Trong trường hợp này, biểu diễn này bị vấn đề thưa thớt xếp hạng theo ngữ cảnh (contextual rating sparsity problem). Việc xử lý từng không gian một cách độc lập sẽ ngăn thuật toán tận dụng ưu thế của việc so sánh mà có thể làm được thông qua nhiều chiều không gian.

Để giảm thiểu vấn đề này, chúng ta biểu diễn mỗi trạng thái ngữ cảnh bởi một vector trọng số thông qua một tập các hệ số tiềm ẩn (theo kinh nghiệm thì chúng ta sử dụng 5 hệ số tiềm ẩn), mà trọng số được khởi tạo vào thời điểm bắt đầu và được học qua quá trình tối ưu. Tích vô hướng giữa hai vector có thể được dùng để biểu đạt sự tương quan giữa mỗi cặp trạng thái ngữ cảnh. Kết quả là ngay cả khi có một gặp mới mà không tồn tại trong bộ dữ liệu dạy, trọng số trong vector để biểu diễn hai trạng thái (“Thời gian = Cuối tuần” và “Thời gian = Kỳ nghỉ”) sẽ được học và cập nhật bởi quá trình học thông qua các cặp đang tồn tại, và sự tương quan của cặp mới có thể được tính toán một cách dễ dàng bằng sử dụng tích vô hướng. Sự tương quan được cho bởi:

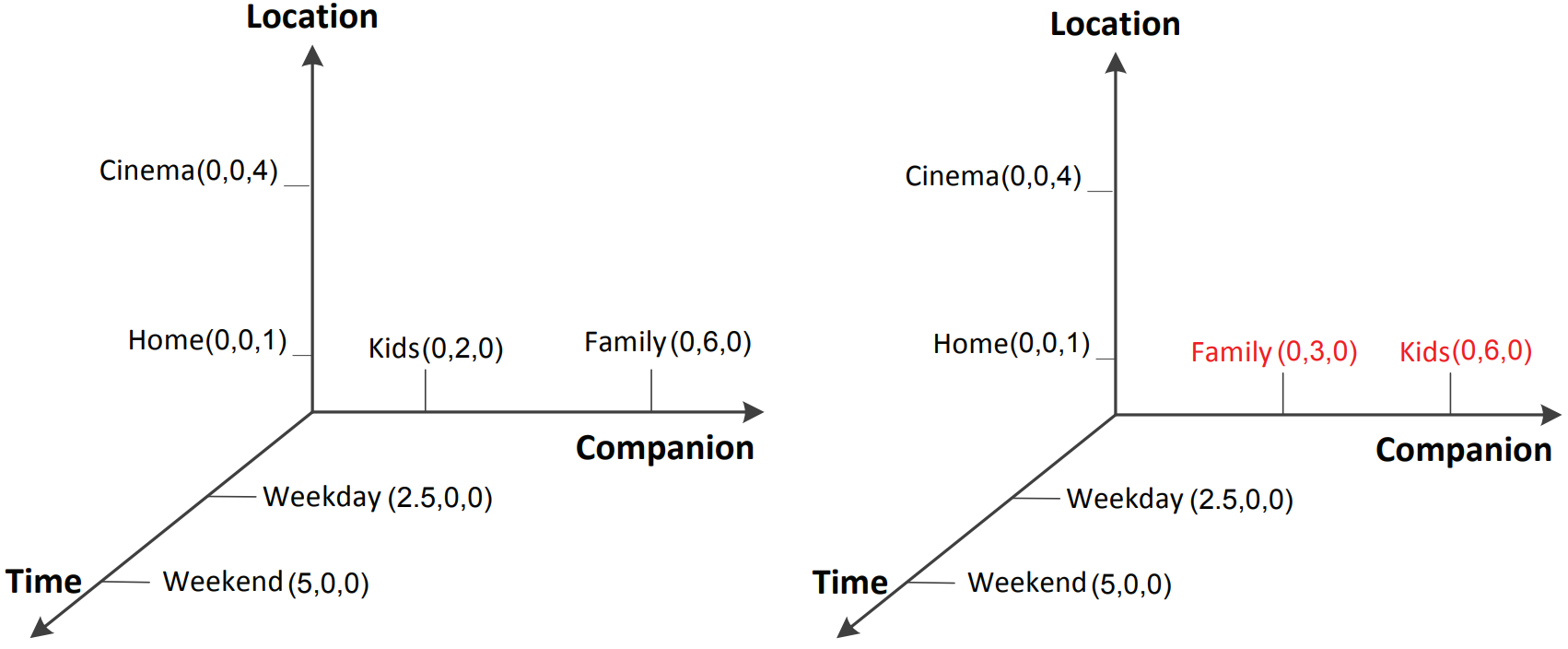
(7)

với và biểu diễn vector đại diện cho trạng thái ngữ cảnh và , tương ứng trong không gian của các yếu tố tiềm ẩn. Sau đó, chúng ta sử dụng phép tính tương quan tương tự như công thức (6). Chúng ta gọi phương pháp này là mô hình Latent Context Similarity (LCS) . Phương pháp này cũng cho phép tăng thêm hiệu suất của thuật toán SCLIM dựa trên sai lệch. Trái ngược với phương pháp tiếp cận bối cảnh độc lập tương tự nhau, LCS tăng thêm khả năng đáp ứng, nhưng cũng thêm hao tổn tính toán qua việc học các hệ số tiềm ẩn. Với LCS, những gì cần được học trong quá trình tối ưu là các vector trọng số đại diện cho từng trạng thái ngữ cảnh.

* + 1. Bối cảnh đa chiều tương tự nhau

Multidimensional Context Similarity (MCS)

Trong mô hình MCS, chúng ta giả định rằng các chiều ngữ cảnh tạo thành một hệ tọa độ đa chiều. Như ví dụ ở hình 2.10.



Hình 2.10: Ví dụ của hệ tọa độ đa chiều

Chúng ta giả định rằng có ba chiều không gian ngữ cảnh: Thời gian, Vị trí và Đồng hành. Chúng ta gán giá trị thực sự cho từng trạng thái ngữ cảnh trong những chiều này, do đó mỗi trạng thái được xác định một vị trí trong trục tương ứng. Trong trường hợp này, một ngữ cảnh (là một tập các trạng thái ngữ cảnh) có thể được biểu diễn như một điểm trong không gian đa chiều. Theo đó, khoảng cách giữa hai điểm có thể được sử dụng như nền tảng cho tính toán tương quan. Trong phương pháp này, những giá trị thực cho mỗi trạng thái ngữ cảnh là thông số được học trong quá trình tối ưu. Ví dụ, những giá trị “gia đình” và “trẻ em” được cập nhật trong phía bên phải của hình 2.10. Vậy nên vị trí của các điểm dữ liệu được cấp cho hai trạng thái ngữ cảnh này sẽ được thay đổi, tương tự với khoảng cách giữa tính tương quan của hai ngữ cảnh. Do đó, mối tương quan có thể được tính bằng nghịch đảo khoảng cách giữa hai điểm dữ liêu. Trong các thí nghiệm, chúng tôi sử dụng khoảng cách Euclidean để tính khoảng cách, mặc dù các cách tính khoảng cách khác có thể được sử dụng. Chi phí tính toán liên quan trực tiếp đến số lượng trạng thái ngữ cảnh trong bộ dữ liệu, mà có thể khiến cho phương pháp này là là mô hình hao phí nhất. Một lần nữa, số lượng trạng thái ngữ cảnh có thể được giảm bởi việc lựa chọn ngữ cảnh.

* 1. Các phương pháp đánh giá

Trong ngữ cảnh của một bài toán xếp hạng, hồi đáp nhị phân nghĩa là các nhãn nhị phân đã được gán cho các sản phẩm, và các sản phẩm có hồi đáp dương được coi như quan sát đúng đắn. Với bảng 2.5 (dummy\_truth), ta giả định rằng có 3 người dùng (userid = 1,2,3) có đúng ba đánh giá chân trị giống nhau cho ba mục (itemid = 1, 2, 4) được chọn từ sáu mục có sẵn:

|  |  |
| --- | --- |
| userid | itemid |
| 1 | 1 |
| 1 | 2 |
| 1 | 4 |
| 2 | 1 |
| 2 | 2 |
| 2 | 4 |
| 3 | 1 |
| 3 | 2 |
| 3 | 4 |

Bảng 2.5: Bảng dummy\_truth

|  |  |  |
| --- | --- | --- |
| userid | itemid | Score |
| 1 | 1 | 10.0 |
| 1 | 3 | 8.0 |
| 1 | 2 | 6.0 |
| 1 | 6 | 2.0 |
| 2 | 1 | 10.0 |
| 2 | 3 | 8.0 |
| 2 | 2 | 6.0 |
| 2 | 6 | 2.0 |
| 3 | 1 | 10.0 |
| 3 | 3 | 8.0 |
| 3 | 2 | 6.0 |
| 3 | 6 | 2.0 |

Bảng 2.6: Bảng dummy\_rec

Thêm nữa, chúng ta xem bảng 2.6 dummy\_rec à kết quả của dự đoán. Làm sao chúng ta so sánh bảng dummy\_rec với bảng dummy\_truth để đánh giá độ chính xác?

Gọi người dùng u U, tập sản phẩm I, một tập theo thứ tự của k sản phẩm gợi ý , và một tập các sản phẩm đúng . Do đó, khi chúng ta chạy gợi ý top-2 cho hai bảng trên. , và hai sản phẩm có xếp hạng cao nhất , và . Sau khi tính toán, ta có 6 kết quả khác nhau và kết quả là:

|  |  |  |
| --- | --- | --- |
| Ranking measure | top-4 (max\_k) | top-2 |
| Recall | 0.666666666666666 | 0.333333333333333 |
| Precision | 0.5 | 0.5 |
| MAP | 0.555555555555555 | 0.333333333333333 |
| AUC | 0.75 | 1.0 |
| MRR | 1.0 | 1.0 |
| NDCG | 0.703918089034139 | 0.613147192765485 |

Bảng 2.7: Kết quả dự đoán

Ở đây, chúng ta giới thiệu về 6 phương pháp đánh giá cho xếp hạng của một danh sách các sản phẩm. Quan trọng là mỗi số có một khái niệm khác nhau đằng sau công thức, và độ chính xác được tính toán bởi các công thức thể hiện một giá trị khác nhau cho cùng một bộ dữ liệu đầu vào ở trên.

* + 1. Recall-At-k (Rec)

**Recall-at-k (Recall@k)** chỉ ra phạm vi bao phủ của các mẫu đúng (truth samples) mà được xem như là kết quả của k gợi ý hàng đầu.

Ở đây, là số lượng các kết quả “đúng tích cực” (true positives).

Với và ,

* + 1. Precision-At-k (Pre)

Không giống như Recall@k, **Precision-at-k (Precision@k)** tính toán sự đúng đắn của k gợi ý hàng đầu trong danh sách dựa theo phần “đúng tích cực” trong danh sách.

Nói cách khác, Precision@k có nghĩa là bao nhiêu danh sách đề xuất bao gồm các cặp thực sự. Ở đây, với và .

* + 1. Độ chính xác trung bình (MAP)

Mean Average Precision (MAP)

Trong khi Precision@k nguyên bản cho thấy một mức điểm cho một danh sách gợi ý có độ dài cố định , **độ chính xác trung bình** tính toán một trung bình của các điểm thông qua kích thước của tất cả gợi ý từ 1 tới . MAP được phát biểu với một hàm chỉ định cho (sản phẩm thứ *n* cho .

Chú ý rằng, MAP không chỉ là một trunh bình cộng đơn giản của Precision@1, Precision@2, …, Precision@, và “đúng tích cực” được xếp hạng cao hơn dẫn đến một MAP tốt hơn. Ví dụ:

Với , trong khi:

* + 1. Khoảng dưới đường cong ROC (AUC)

Area Under the ROC Curve (AUC)

Đường cong ROC và **khoảng dưới đường cong ROC** được dùng phổ biến trong tính toán các bài toán phân lớp. Tuy nhiên, những khái niệm này cũng có thể được giải thích trong bối cảnh của bài toán xếp hạng.

Về cơ bản, số liệu AUC để xếp hạng được dùng để xem xét tất cả các cặp “đúng tích cực” và các sản phẩm khác mà nó được ký hiệu tương ứng bởi và , và kỳ vọng rằng những gợi ý được xếp hạng tốt nhất cao hơn . Điểm số cuối cùng là được tính từ các phần của các cặp có thứ tự đúng () trong tất cả các kết hợp có thể, được xác định bởi tập ký hiệu .

* + 1. Xếp hạng đối ứng trung bình (MRR)

Mean Reciprocal Rank (MRR)

Nếu chúng ta chỉ quan tâm đến cái “đúng tích cực” đầu tiên, **xếp hạng đối ứng trung bình** có thể được xem là một lựa chọn hợp lý để đánh giá định lượng cho các danh sách gợi ý. Cho , vị trí của “đúng tích cực” đầu tiên trong , MRR đơn giản là cho kết quả ngược lại:

MRR có thể bằng 0 khi và chỉ khi rỗng.

Trong các bảng giả đã miêu tả ở trên, giá trị đúng đầu tiên được đặt ở vị trí đầu tiên trong danh sách xếp hạng các sản phẩm. Do đó, MRR = 1/1 = 1.

* + 1. Chuẩn hóa chiết khấu tích lũy (NDCG)

Normalized Discounted Cumulative Gain (NDCG)

**Chuẩn hóa chiết khấu tích lũy** tính điểm cho trong đó nhấn mạnh vào các “đúng tích cực” được xếp hạng cao hơn. Ngoài việc là một công thức tính toán tốt hơn, sự khác biệt giữa NDCG và MPR là NDCG cho phép chúng ta chỉ định xếp hạng dự kiến trong , đó là, số liệu có thể kết hợp , một số điểm liên quan cho thấy khả năng mẫu thứ n được xếp hạng ở đầu danh sách gợi ý và tương ứng trực tiếp với xếp hạng dự kiến của các mẫu đúng.

Như kết quả của k gợi ý hàng đầu, NDCG được tính theo công thức:

Với:

Ở đây, cho biết mức độ phù hợp với hoán vị thật (the truth permutation), và là tốt nhất mà thật sự khớp với .

Bây giờ, chúng ta chỉ quan tâm đến hồi đáp nhị phân, vậy điểm nhị phân là:

Bởi vì gợi ý của chúng ta dùng là lấy 2 gợi ý tốt nhất, nên suy ra ta có: .

Trong khi đó, chỉ có mẫu đầu tiên trong là “đúng tích cực”, do đó: .

Do đó, .

# THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG

* 1. Phân tích chức năng hệ thống
     + 1. Tùy chỉnh các đường dẫn thư mục

Hệ thống yêu cầu 1 tập tin cấu hình chứa các thông số đầu vào cho hệ thống bao gồm đường dẫn tập tin đầu vào, tên thuật toán mặc định, và các thông số để chạy thuật toán.

* + - 1. Chạy các thuật toán

Người dùng có thể tùy chọn và chạy tất cả thuật toán được hỗ trợ trong ứng dụng và lịch sửa sẽ được lưu lại và

* + - 1. Điều chỉnh thông số cho hệ

Với từ hệ gợi ý sẽ có các thông số đầu vào, người dùng có thể tùy chỉnh để tối ưu kết quả đạt được của thuật toán với mỗi bộ dữ liệu khác nhau.

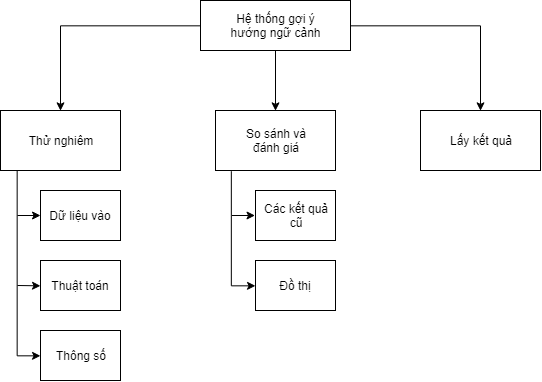
* + - 1. So sánh giữa kết quả của các thuật toán

Phần mềm sẽ lưu lại kết quả tốt nhất của thuật toán và từ đó cho người dùng cái nhìn tổng quan nhằm lựa chọn thuật toán tốt nhất cho bộ dữ liệu. Từ đó sử dụng bộ dữ liệu đó để áp dụng vào hệ gợi ý.

* + - 1. Lấy các kết quả gợi ý

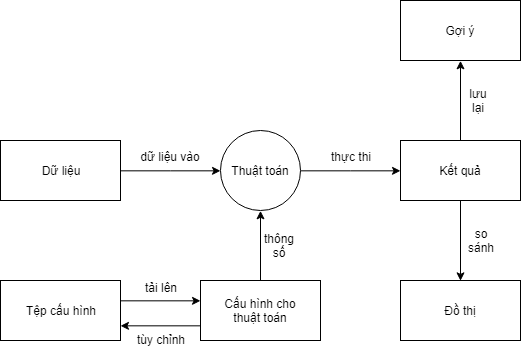
Với mỗi thuật toán sẽ sinh ra một loạt các kết quả gợi ý mặt hàng thích hợp cho người dùng được lưu vào các tệp để người dùng có thể sử dụng. Người dùng sẽ sử dụng nó để tạo ra bộ dữ liệu gợi ý cho khách hàng.

* 1. Các sơ đồ hệ thống
     + 1. Sơ đồ phân cấp chức năng



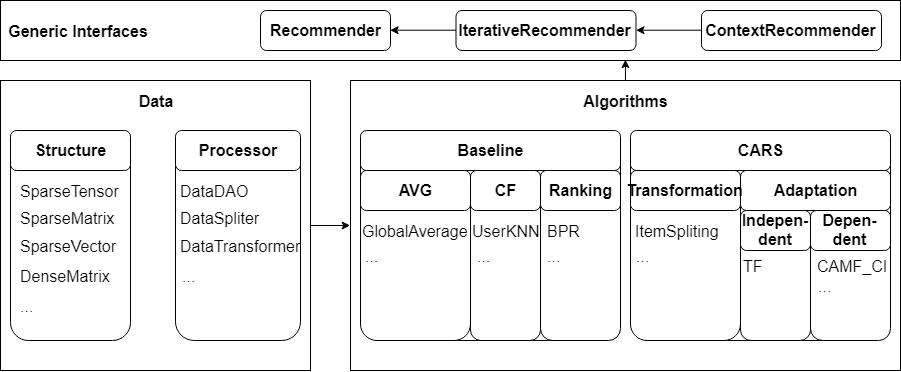
Hình 3.1: Sơ đồ phân cấp chức năng của ứng dụng

* + - 1. Sơ đồ luồng dữ liệu



Hình 3.2: Sơ đồ luồng dữ liệu của ứng dụng

* + - 1. Sơ đồ thuật toán

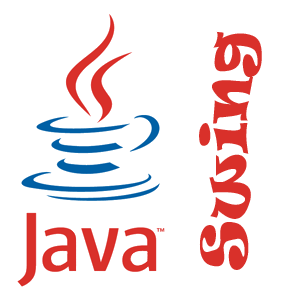
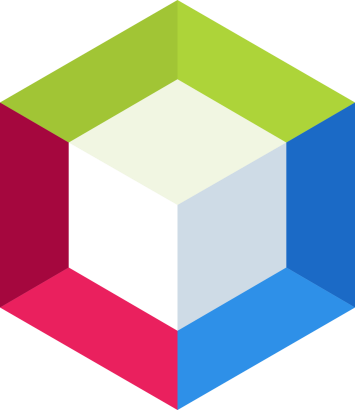


Hình 3.3: Sơ đồ dữ liệu thuật toán

* 1. Thiết kế hệ thống

Giao diện chương trình sử dụng Java Swing để tạo giao diện đồ họa cho chương trình.

Dùng IDE NetBeans để hỗ trợ việc lập trình giao diện đơn giản hơn.

Java Swing Logo NetBeans Logo

* + 1. Xây dựng các lớp nền
       1. Lớp quản lý tệp cấu hình

|  |
| --- |
| **public** **class** ManageConfig {  **private** **static** **boolean** *debug* = **true**;  **private** **static** **boolean** *show\_detail* = **false**;  **private** String config\_path;  **private** Vector<String> config\_lines = **new** Vector<String>();  **private** Vector<Boolean> is\_config = **new** Vector<Boolean>();  **private** **boolean** checked\_path\_user;  **public** Map<String, CARSConfig> configs = **new** Hashtable<String, CARSConfig>();  **private** Pattern re\_name = Pattern.*compile*("[\\w.]\*=");  **public** ManageConfig(String path) {}  **private** **void** check\_path() {}  **public** **void** read\_config() **throws** IOException {}  **public** **void** write\_config() {}  **public** **boolean** change\_config(String name, String conf, String value) {}  **public** **boolean** change\_config\_data\_path(String old\_path, String new\_path) {}  **public** String get\_path() {}  **public** **static** String default\_config() {}  **public** **static** String default\_sample() {}  **public** **static** **void** main(String[] args){}  } |

Lớp này có mục đích quản lý toàn bộ cấu hình có trong tệp cấu hình và tự động cập nhật tệp cấu hình khi người dùng thao tác tên giao diện.

* + - 1. Lớp quản lý từng cấu hình

|  |
| --- |
| **public** **class** CARSConfig {  **private** **static** **boolean** *debug* = **false**;  **public** **int** line\_number = 0;  **public** String name = "";  **public** String value = "";  **public** String line = "";  **public** Map<String, String> config = **new** Hashtable<String, String>();  **private** Pattern re\_value = Pattern.*compile*("=[^ \\n,]+");  **private** Pattern re\_details = Pattern.*compile*("-[a-zA-Z][\\w-]\* -?[.\\w]+");  **public** CARSConfig(String config\_name, String config\_line) {}  **public** String to\_string() {}  **public** **void** add\_config(String config\_string) {}  **public** **void** change\_line(String old\_config, String new\_config) {}  **public** **void** change\_config(String key, String value) {}  } |

Lớp này giúp lọc ra từng cấu hình theo từng dòng. Lưu trữ cấu hình, chỉnh sửa cấu hình và cuối cùng khi cần xuất ra tệp, nó sẽ trả về cấu hình đã được chỉnh sửa đúng theo ý người dùng.

* + - 1. Lớp quản lý lịch sử

|  |
| --- |
| **public** **class** CARSResult {  **private** String result\_path;  **private** String COMMA\_DELIMITER = ",";  **public** List<List<Double>> all\_value\_results;  **public** List<String> all\_algo\_names;  **public** Map<String, List<Double>> map\_result;    **public** CARSResult(String path) {}  **public** CARSResult(List<String> algos, List<List<Double>> values) {}  **public** **static** **void** main(String[] args) {}  **public** **void** read\_result() **throws** IOException {}  **public** **void** make\_map() {}  **public** **boolean** greater(List<Double> fi, List<Double> se) {}  **public** Double sum(List<Double> list) {}  } |

Lớp này giúp ứng dụng có thể đọc lại toàn bộ lịch sử mà người dùng đã thao tác với thuật toán, chọn ra kết quả tốt nhất trong lịch sử của mỗi thuật toán để đưa lên cho người dùng.

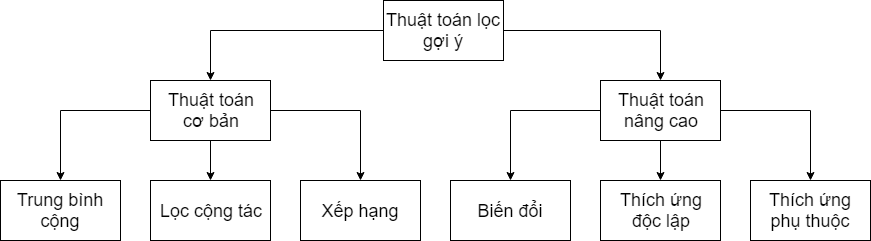
* + - 1. Lớp vẽ đồ thị

|  |
| --- |
| **public** **class** CARSGraph **extends** JPanel {  **private** **int** width = 800;  **private** **int** height = 600;  **private** **int** padding = 25;  **private** **int** labelPadding = 25;  **private** Color pointColor = **new** Color(100, 100, 100, 180);  **private** Color gridColor = **new** Color(200, 200, 200, 200);  **private** **static** **final** Stroke ***GRAPH\_STROKE*** = **new** BasicStroke(2f);  **private** **int** pointWidth = 4;  **private** **int** numberYDivisions = 10;  **private** **static** Map<String, Color> *map\_colors* = **new** HashMap<>();  **private** Map<String, List<Double>> map\_result;  **public** CARSGraph(List<List<Double>> values, List<String> algos) {}  **public** CARSGraph(Map<String, List<Double>> map) {}    **private** **void** init\_color() {}  @Override  **protected** **void** paintComponent(Graphics g) {}  **private** **double** getMinScore() {}  **private** **double** getMaxScore() {}  **public** **int** getWidth() {}  **public** **int** getHeight() {}  **public** **void** setSize(**int** Width, **int** Height) {}  **private** **static** **void** createAndShowGui() {}  **public** **static** **void** main(String[] args) {}  } |

Lớp này tạo ra một biểu đồ đường thẳng giúp người dùng dễ dàng so sánh các thuật toán khác nhau.

* 1. Thuật toán

Hệ thống có một loạt các thuật toán được chia thành nhiều loại:



Hình 3.4: Các nhóm thuật toán

Các thuật toán cơ bản đã được giait thích trong phương pháp tiếp cận truyền thống và hiện đại. Các thuật toán biến đổi sử dụng thêm một bộ phân chia dữ liệu trước khi thực thi các thuật toán gợi ý cơ bản. Các thuật toán thích ứng độc lập và thích ứng phụ thuộc được giải thích ở phần phương pháp tiếp cận phân rã ma trận và theo tương quan ngữ cảnh.

* + 1. Lớp mẫu của mọi thuật toán:

|  |
| --- |
| **public** **class** ClassName **extends** Recommender{  **public** ClassName() {}  @Override  **protected** **void** initModel() **throws** Exception {}  @Override  **protected** **void** buildModel() **throws** Exception {}  @Override  **protected** **double** predict(**int** u, **int** j, **int** c) **throws** Exception {}  @Override  **protected** **double** predict(**int** u, **int** j) **throws** Exception{}  } |

Mọi thuật toán đều được kế thừa từ lớp Recommender và viết chồng lên các hàm dự đoán (predict) và sẽ được gọi thực thi ngoài chương trình chính bằng cách gọi hàm execute

* + 1. Các thuật toán trung bình

Các thuật toán tính trung bình bao gồm: Global Average, User Average, User Average, Item Average, Context Average, User Item Average, User Context Average và Item Context Average. Ví dụ hàm dự đoán của lớp Context Average:

|  |
| --- |
| **protected** **double** predict(**int** u, **int** j, **int** c) {  **return** *rateDao*.getContextAvg(trainMatrix,c);  } |

* + 1. Các thuật toán lọc cộng tác

Các thuật toán lọc cộng tác bao gồm: Item KNN, User KNN, Slope One, Biased MF, PMF (Probabilistic Matrix Factorization), BPMF (Bayesian Probabilistic Matrix Factorization), NMF (None-negative Matrix Factorization), SVD++ (Singular Value Decomposition). Ví dụ lớp hàm dự đoán của lớp SlopOne:

|  |
| --- |
| **protected** **double** predict(**int** u, **int** j) **throws** Exception{  SparseVector uv = train.row(u, j);  **double** preds = 0, cards = 0;  **for** (**int** i : uv.getIndex()) {  **double** card = cardMatrix.get(j, i);  **if** (card > 0) {  preds += (devMatrix.get(j, i) + uv.get(i)) \* card;  cards += card;  }  }  **return** cards > 0 ? preds / cards : globalMean;  } |

* + 1. Thuật toán xếp hạng

Các thuật toán xếp hạng bao gồm: BPR (Bayesian personalized ranking), SLIM (Sparse Linear Methods), LRMF (Low-Rank Matrix Factorization), Rank ALS (Alternating Least Square), Rank SGD (Stochastic Gradient Descent). Ví dụ hàm dự đoán trong lớp SLIM:

|  |
| --- |
| **protected** **double** predict(**int** u, **int** j, **boolean** exclude, **int** excluded\_item) **throws** Exception {  Collection<Integer> nns = *knn* > 0 ? itemNNs.get(j) : allItems;  SparseVector Ru = userCache.get(u);  **double** pred = 0;  **for** (**int** k : nns) {  **if** (Ru.contains(k)) {  **if**(k!= excluded\_item) {  **double** ruk = Ru.get(k);  pred += ruk \* W.get(k, j);  }**else**{  **if**(!exclude){  **double** ruk = Ru.get(k);  pred += ruk \* W.get(k, j);  }  }  }  }  **return** pred;  } |

* + 1. Thuật toán biến dổi

Có 3 phương pháp biến đổi: User Splitting, Item Splitting, User Item Splitting.

Sau khi biến đổi dữ liệu, thuật toán sẽ chạy một trong các thuật toán cơ bản đã nêu ở trước.

* + 1. Thuật toán thích ứng không phụ thuộc

Adaptation Independent

Theo hướng thích ứng không phụ thuộc chỉ có một thuật toán là: Tensor Factorization.

|  |
| --- |
| **protected** **double** predict(**int** u, **int** j, **int** c) **throws** Exception {  **double** pred = 0;  **int**[] keys = getKeys(u,j,c);  **for** (**int** f = 0; f < *numFactors*; f++) {  **double** prod = 1;  **for** (**int** d = 0; d < *numDimensions*; d++) {  prod \*= M[d].get(keys[d], f);  }  pred += prod;  }  **if** (pred > *maxRate*)  pred = *maxRate*;  **if** (pred < *minRate*)  pred = *minRate*;  **return** pred;  } |

* + 1. Thuật toán thích ứng phụ thuộc

Thuật toán thích ứng phụ thuộc bao gồm 3 hướng phát triển: phụ thuộc dựa trên sự khác nhau, phụ thuộc dựa trên sự giống nhau và phân rã ma trận.

* + - 1. Thuật toán thích ứng phụ thuộc thiên vị (deviation base)

Các thuật toán phụ thuộc thiên vị bao gồm: CAMF\_C, CAMF\_CI, CAMF\_CU, CAMF\_CUCI, CSLIM\_C (Contextual SLIM using Context), CSLIM\_CI, CSLIM\_CU, CSLIM\_CUCI, CSLIM\_CC. Ví dụ hàm dự đoán trong lớp CAMF\_C:

|  |
| --- |
| **protected** **double** predict(**int** u, **int** j, **int** c) **throws** Exception {  **double** pred = globalMean + userBias.get(u) + itemBias.get(j) +  DenseMatrix.*rowMult*(P, u, Q, j);  **for**(**int** cond:getConditions(c)){  pred+=condBias.get(cond);  }  **return** pred;  } |

* + - 1. Thuật toán thích ứng phụ thuộc tương quan

Các thuật toán thích ứng phụ thuộc tương quan bao gồm: CAMF\_ICS, CAMF\_LCS, CAMF\_MCS, CSLIM\_ICS, CSLIM\_LCS, CSLIM\_MCS, GCSLIM\_ICS (Improve General Context SLIM), GCSLIM\_LCS, GCSLIM\_MCS. Ví dụ hàm dự đoán của lớp CAMF\_ICS:

|  |
| --- |
| **protected** **double** predict(**int** u, **int** j, **int** c) **throws** Exception {  **double** pred=DenseMatrix.*rowMult*(P, u, Q, j);  List<Integer> conditions=getConditions(c);  **for**(**int** i=0;i<conditions.size();++i)  pred=pred\*ccMatrix\_ICS.get(conditions.get(i),  *EmptyContextConditions*.get(i));  **return** pred;  } |

* + - 1. Thuật toán máy nhân tố

Factorization Machines

|  |
| --- |
| **protected** **double** predict(**int** u, **int** j, **int** c) **throws** Exception {  DenseVector fs = getFeatureVector(u,j,c);  **double** pred=w0;  **for**(j=0;j<p;++j)  pred+=w.get(j)\*fs.get(j);  **double** sum = 0.0;  **for**(**int** f=0;f<k;++f){  **double** sum1=0.0, sum2=0.0;  **for**(j=0;j<p;++j){  **double** dot = V.get(j,f)\*fs.get(j);  sum1 += V.get(j,f)\*fs.get(j);  sum2 += Math.*pow*(dot, 2);  }  sum += Math.*pow*(sum1,2) - sum2;  }  pred+=0.5\*sum;  **return** pred;  } |

# KẾT QUẢ THỰC NGHIỆM

* 1. Quy trình sinh dữ liệu

Ứng dụng sử dụng dữ liệu từ một website có sẵn do một đồ án của người khác về du lịch. Tuy dữ liệu ban đầu là random sinh ra một bộ dữ liệu có logic nhưng sau đó chúng tôi sử dụng dữ liệu thật sự tự nhập. Và dữ liệu vào là kết hợp cả 2 bộ dữ liệu để dưa ra gợi ý.

* + 1. Dữ liệu thực tế

|  |  |
| --- | --- |
| user\_id | name |
| 1 | Trương Quang Ngữ |
| 2 | Tăng Văn Quốc Chí |
| 3 | Nguyễn Thanh Bằng |

Bảng 4.1: Bảng người dùng

|  |  |
| --- | --- |
| travel\_id | name |
| 1 | Bà Nà núi chúa |
| 2 | Trượt thác Hòa Phú Thành |
| 3 | Bãi Rạng |

Bảng 4.2: Bảng địa điểm

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| comment\_id | user\_id | travel\_id | content | rating | partner | time |
| 1 | 1 | 1 | Giá vé đắt quá | 2.0 | Một mình | Giữa tuần |
| 2 | 1 | 2 | Cuối tuần đi tốt nhất với bạn gái | 5.0 | Bạn gái | Cuối tuần |
| 3 | 2 | 3 | Tôi thích đến đây với bạn bè | 4.0 | Bạn bè | Cuối tuần |
| 4 | 3 | 2 | Hè tránh nóng ở đây thật tuyệt | 5.0 | Gia đình | Cuối tuần |
| 5 | 1 | 3 | Đi cùng bạn bè đến đây dịp lễ thì còn gì bằng | 4.0 | Bạn bè | Lễ |

Bảng 4.3: Bảng đánh giá của người dùng cho địa điểm

Đây là cách lưu trữ dữ liệu mà đa số các hệ thống du lịch sử dụng.

* + 1. Dữ liệu được chuẩn bị

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| user\_id | travel\_id | rating | partner | time |
| 1 | 1 | 2.0 | alone | weekday |
| 1 | 2 | 5.0 | GF/BF | weekend |
| 2 | 3 | 4.0 | friends | weekend |
| 3 | 2 | 5.0 | family | weekend |
| 1 | 3 | 4.0 | friends | holiday |

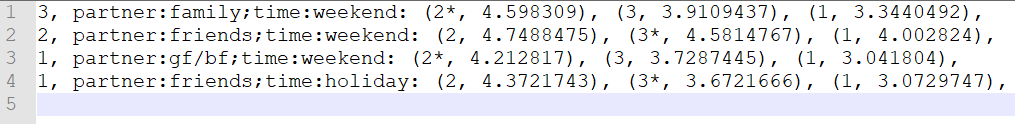
Bảng 4.4: Dữ liệu chuẩn bị để đưa vào hệ thống

* + 1. Dữ liệu được biến đổi sang dạng nhị phân

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| user | item | rate | partner: alone | partner: friends | partner: family | partner: gf/bf | time: weekday | time: weekend | time: holiday |
| 1 | 1 | 2.0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 2 | 5.0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 2 | 3 | 4.0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 3 | 2 | 5.0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 3 | 4.0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

Bảng 4.5: Dữ liệu đã được hệ thống biến đổi sang dạng nhịn phân

* + 1. Kết quả thu được sau khi chạy thuật toán



Hình 4.1: Dữ liệu thu được sau khi chạy thuật toán CAMF\_C

Dữ liệu trên dòng thứ 1 có nghĩa là:

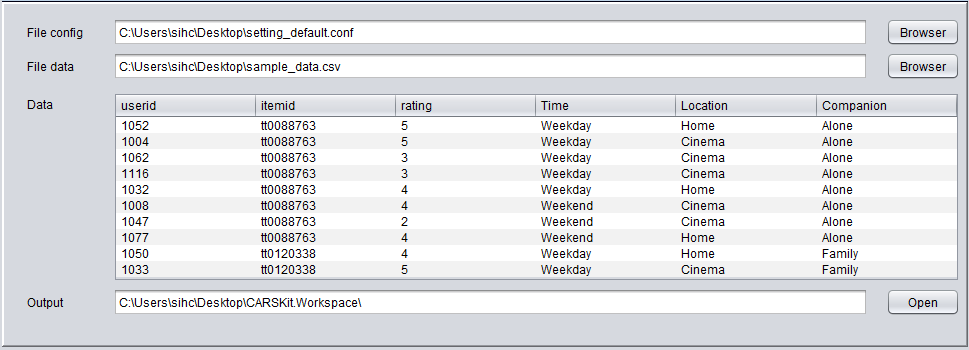
Với người dùng 3, trong ngữ cảnh đi với gia đình vào cuối tuần, họ sẽ đánh giá chuyến số 2: 4.6 và chuyến số 3: 3.9 và chuyến số 1: 3.3 và chuyến số 2 (chuyến có dấu \*) là chuyến nên gợi ý đầu tiên

Tương tự với các người dùng còn lại, ta sẽ lấy ra danh sách các điểm nên gợi ý cho người dùng và hệ số xếp hạng và sau đó, ta sắp xếp lại dữ liệu đó ta sẽ có bộ dữ liệu gợi ý riêng cho mỗi người dùng.

* 1. Triển khai hệ thống
     1. Yêu cầu hệ thống
* Java: Yêu cầu jdk và jre từ 8 trở lên để có thể sử dụng được chương trình.
* OS: Hệ thống được lập trình trên cả hai hệ điều hành Windows và MacOS (Unit) nên có thể chạy trên hầu hết hệ điều hành kể cả Linux
* IDE: Có thể sử dụng cả Eclipse và NetBeans tuy nhiên vì hệ thống yêu cầu java 8 nên cầu điều chỉnh lại jdk và jre.
  + 1. Tùy chọn tệp cấu hình

Hệ thống yêu cầu 1 tập tin setting.conf chứa các thông số đầu vào cho hệ thống bao gồm đường dẫn tập tin dữ liệu đầu vào, tên thuật toán mặc định, và các thông số để chạy thuật toán. Tệp cấu hình này ban đầu sẽ được tạo ra mặc định, người dùng trong quá trình thử nghiệm thuật toán mà tinh chỉnh để đạt được kết quả tốt nhất.

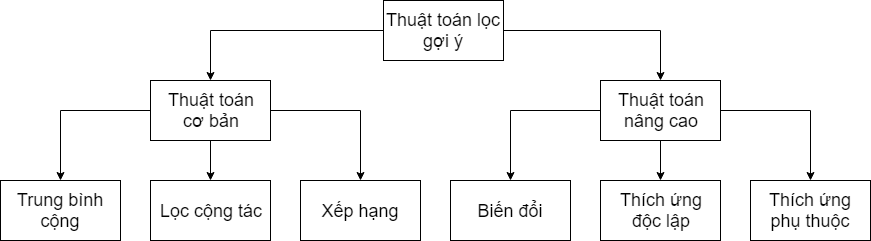
Việc thay đổi được đường tệp cấu hình cho phép cùng lúc người dùng có thể thử nghiệm với nhiều bộ dữ liệu khác nhau tại các vị trí khác nhau với các bộ cấu hình khác nhau.



Hình 4.2: Nơi nhập tệp cấu hình và tệp dữ liệu

* + 1. Thực thi thuật toán

Với mỗi bộ dữ liệu khác nhau thì tùy thuộc vào bộ dữ liệu, mục đích người dùng, mục đích trang web và cấu hình hệ thống mà việc lựa chọn thuật toán thích hợp nhất trở nên khó khăn vì thông thường người dùng sẽ khó có cơ hội có thể thử nghiệm hết. Thế nào là thuật toán thích hợp? Với mỗi trường hợp khác nhau thì hệ thống sẽ ưu tiên một mục đích khác nhau: tốc độ hơn chính xác, hay chính xác hơn tốc độ, hay trung bình. Từ đó, việc tạo cho người dùng cơ hội có thể thực thi tất cả thuật toán là mục đích sâu xa của hệ thống.



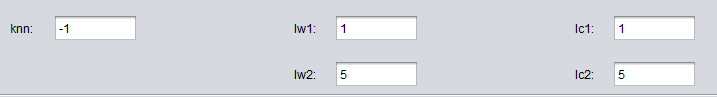
Hình 4.3: Các nhóm thuật toán

* + 1. Điều chỉnh thông số cho hệ

Một số thuật toán có các thông số đầu vào có thể là ngưỡng, tốc độ học và các thông số giới hạn trên dưới của bộ dữ liệu. Việc thay đổi thông số sẽ thay đổi chất lượng của gợi ý.



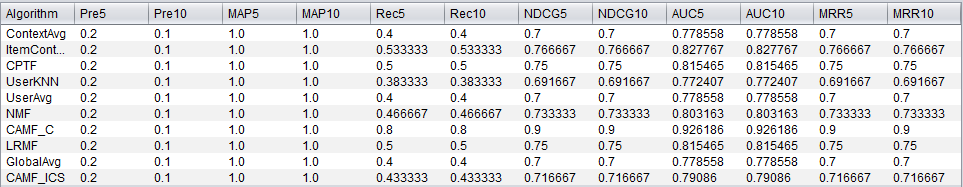
Hình 4.4: Cấu hình chung cho toàn bộ thuật toán



Hình 4.5: Cấu hình riêng cho thuật toán

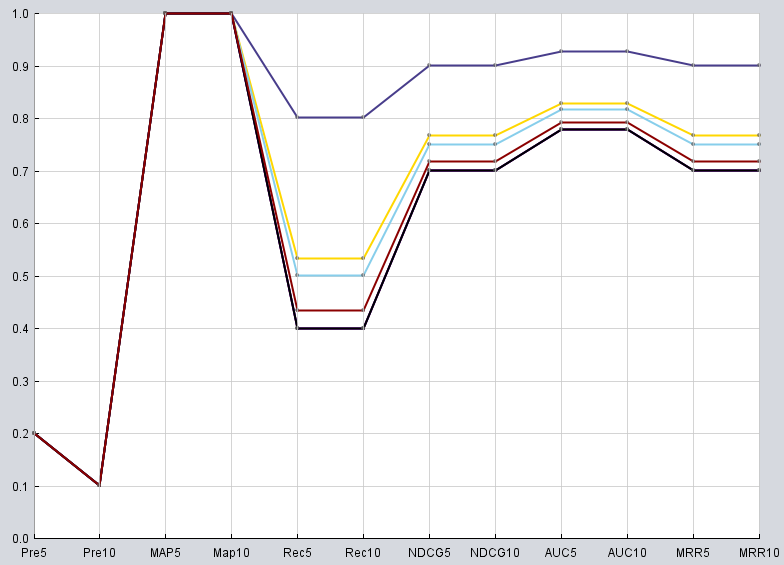
* + 1. So sánh giữa kết quả

Phần mềm sẽ lưu lại mọi kết quả trong quá trình thực thi vào một tệp result\_all.txt (được thiết lập trong tệp cấu hình) và trích xuất nó ra dưới dạng csv trong thư mục CARSKit.Workspace với tên all\_results.csv. Ứng dụng sẽ đọc thông số ngược từ các tệp lịch sử đó lên và lọc ra lần chạy tốt nhất với mỗi thuật toán và hiển thị lên một bảng lịch sử để giúp người dùng dễ dàng xem lại và đưa ra quyết định.



Hình 4.6: Lịch sử thực thi trên từng thuật toán

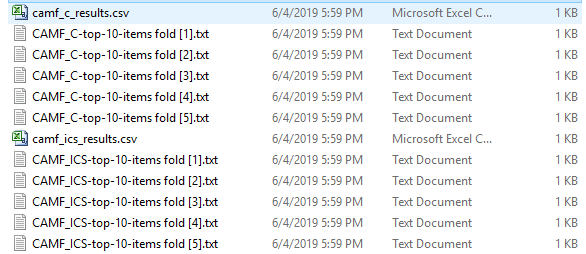
Vì số liệu rất khó nhìn nên hệ thống bổ sung tính năng vẽ biểu đồ để hỗ trợ người dùng trong việc so sánh nhiều thuật toán khác nhau.



Hình 4.7: Biểu đồ so sánh kết quả các thuật toán đã chạy

* + 1. Lấy các kết quả gợi ý

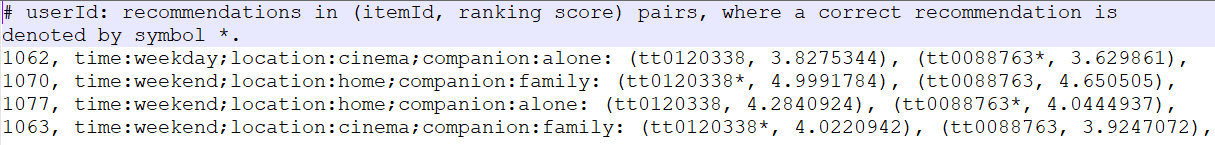
Với mỗi thuật toán sẽ sinh ra một loạt các kết quả gợi ý sản được lưu vào các tệp và đặt tên theo thuật toán để người dùng có thể sử dụng nó như là gợi ý cho khách hàng. Sẽ có 5 tệp lặp lại vì mặc định trong tệp cấu hình sẽ cho mọi thuật toán chạy 5 lần. Tùy chỉnh số lần lặp thông qua cài đặt evaluation.setup= -k 5 trong tệp cấu hình.



Hình 4.8: Tệp kết quả của quá trình chạy thuật toán CAMF\_C và CAMF\_ICS

Trong mỗi tệp sẽ được định dạng theo kiểu:

Id\_người\_dùng, ngữ\_cảnh\_1:giá\_trị\_1;ngữ\_cảnh\_n:giá\_trị\_n: (id\_sản\_phẩm\_1, xếp\_hạng\_sản\_phẩm\_1), (id\_sản\_phẩm\_m, xếp\_hạng\_sản\_phẩm\_m),



Hình 4.9: Tệp gợi ý trong một thuật toán bất kỳ

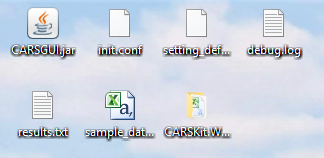
Với hệ thống CARSGUI xem gợi ý có xếp hạng cao nhất không phải là điều mà người dùng thích nhất, nên hệ sẽ đánh dấu \* vào gợi ý để đánh dấu gợi ý tốt nhất.

* 1. Hướng đẫn sử dụng
     1. Lần đầu khởi chạy ứng dụng

Chương trình đã được cài đặt nếu chưa dữ liệu ban đầu, nó sẽ khởi động với bộ dữ liệu mẫu tự sinh và một tệp cấu hình mẫu. Người dùng chỉ cần chạy là có thể sử dụng ngay.



Hình 4.10: Tập tin ban đầu



Hình 4.11: Tập tin sau khi khởi chạy lần đầu

* sample\_data.csv để tham khảo cách trình bày dữ liệu để nạp vào chương trình.
* Thư mục CARSKit.Workspace sẽ là nơi chứa mọi kết quả của việc thử nghiệm thuật toán.
* debug.log sẽ chứa mọi log của hệ thống trong quá trình chạy
* results.txt chứa lịch sử mọi lần bạn đọc dữ liệu vào với bộ thông số và kết quả của thuật toán.
* init.conf chứa đường dẫn đến tệp setting.conf, bạn có thể thay thế bằng 1 tệp config khác bất kỳ ở đâu.
* setting\_default.conf là cài đặt mặc định của hệ thống, người dùng nên sao chép ra thành 1 tệp setting\_user.conf để sau này có thể khôi phục.

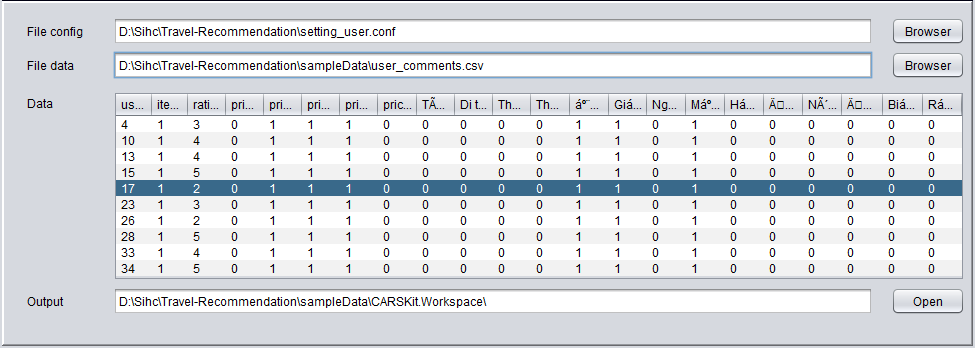
Bộ dữ liệu mặc định đầu đã đủ để cho phép người dùng có thể thao tác thử với dữ liệu mẫu thông qua bộ cấu hình mẫu.

* + 1. Lần khởi chạy sau đó

Hệ thống sẽ tự lưu lại vị trí tệp cấu hình cuối cùng thực thi để sẵn sàng cho người dùng thao tác tiếp tục sau khi đã thoát khỏi ứng dụng.

* 1. Thực nghiệm
     1. Chọn bộ dữ liệu

Với bộ dữ liệu mẫu từ một website từ đồ án khác cung cấp

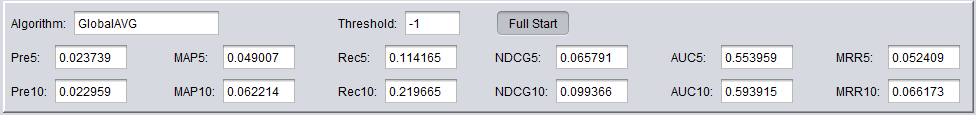


Hình 4.12: Chọn cấu hình và chọn tệp dữ liệu vào

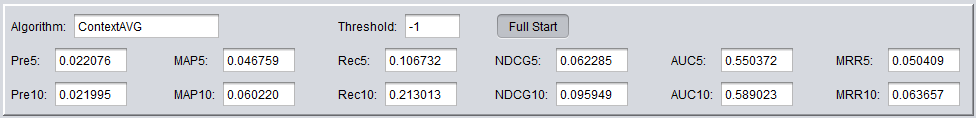
* + 1. Thực thi thuật toán

Với bộ dữ liệu đầu vào, ta thực thi tất cả thuật toán có thể để có thể xem xét thời gian thực thi cùng với so sánh độ chuẩn xác của thuật toán.

* + - 1. Các thuật toán tính trung bình

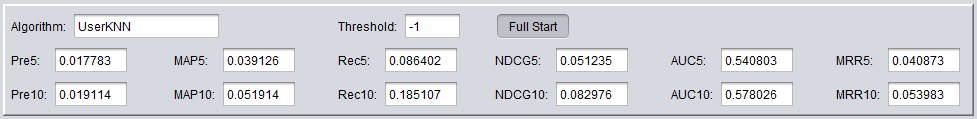


Hình 4.13: Thực thi thuật toán trung bình toàn cục

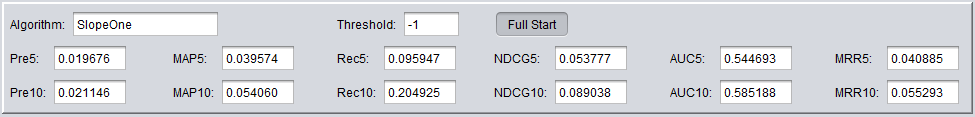


Hình 4.14: Thực thi thuật toán trung bình ngữ cảnh

* + - 1. Các thuật toán lọc kết hợp

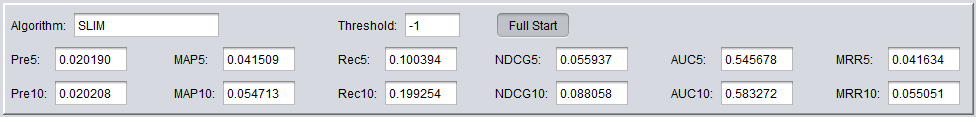


Hình 4.15: Thực thi thuật toán User KNN

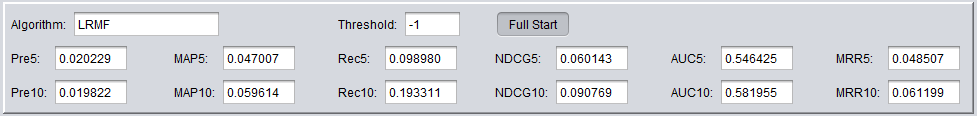


Hình 4.16: Thực thi thuật toán Slope One

* + - 1. Các thuật toán xếp hạng

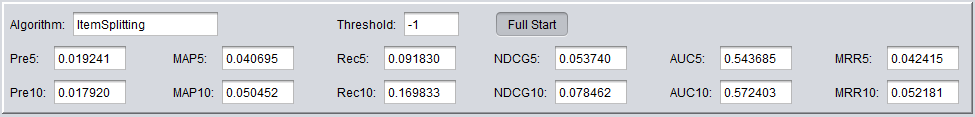


Hình 4.17: Thực thi thuật toán SLIM

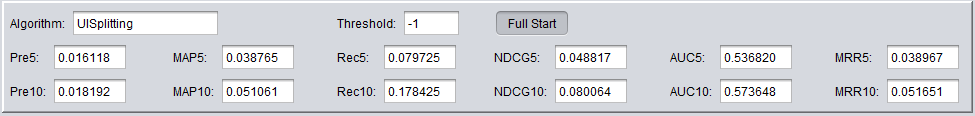


Hình 4.18: Thực thi thuật toán LRMF

* + - 1. Các thuật toán biến đổi kết hợp

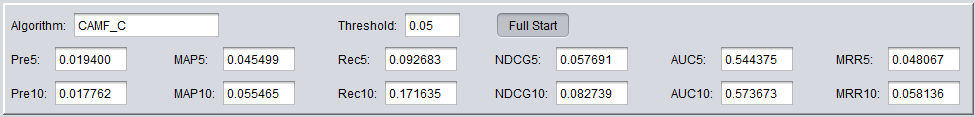


Hình 4.19: Thực thi thuật toán Item Splitting kết hợp Biased MF

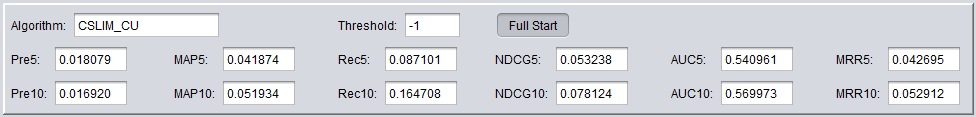


Hình 4.20: Thực thi thuật toán Item User Spliting kết hợp Biased MF

* + - 1. Các thuật toán dựa trên sai lệch

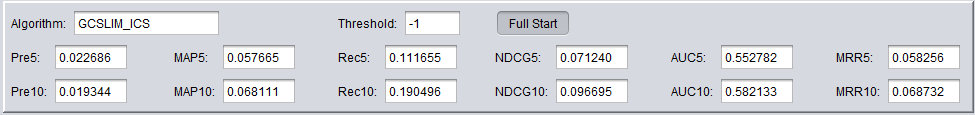


Hình 4.21: Thực thi thuật toán CAMF Context

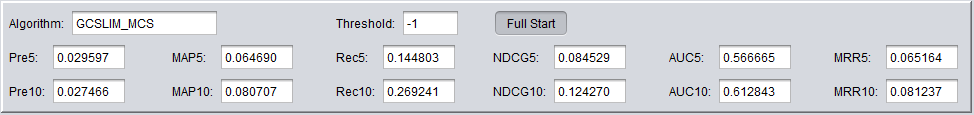


Hình 4.22: Thực thi thuật toán CSLIM Context User

* + - 1. Các thuật toán dựa trên điểm giống nhau

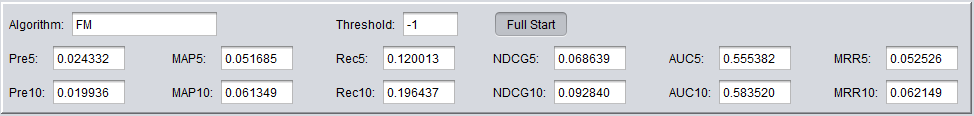


Hình 4.23: Thực thi thuật toán GCSLIM ICS



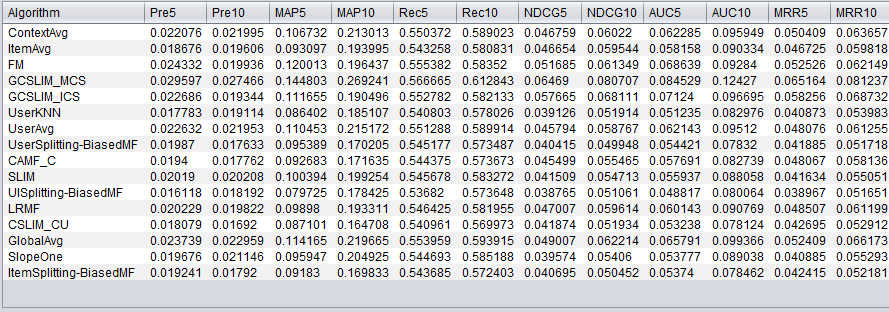
Hình 4.24: Thực thi thuật toán GCSLIM MCS

* + - 1. Thuật toán phân rã ma trận



Hình 4.25: Thực thi thuật toán Matrix Factorization

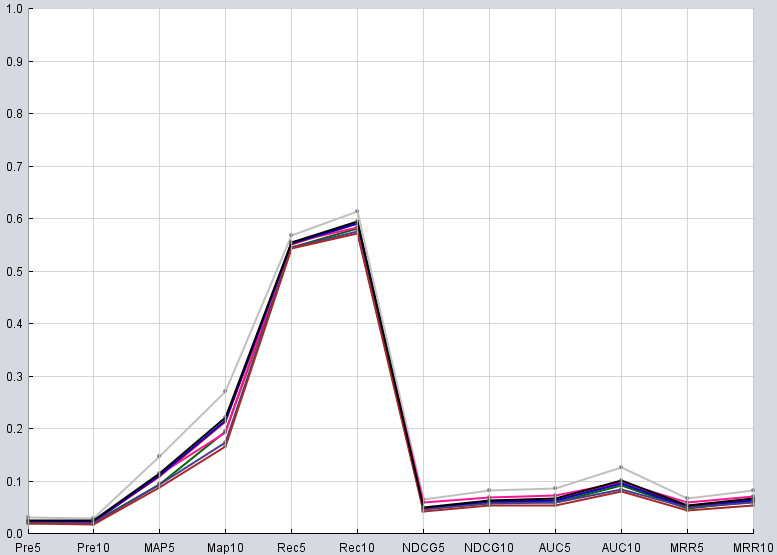
* + 1. Xem lịch sử



Hình 4.26: Lịch sử thực thi thuật toán

Sau khi thực thi một loạt các thuật toán, ta sẽ xem được toàn bộ lịch sử với giá trị tốt nhất của mỗi thuật toán từng thực thi.

* + 1. Xem biểu đồ



Hình 4.27: Biểu đồ kết quả tốt nhất của mỗi thuật toán

Với biểu đồ trên, ta dễ dàng đưa ra nhận xét có một thuật toán đang chạy với kết quả chính xác cao nhất và nhìn lại lịch sử, ta sẽ nhận ra đó là thuật toán GCSLIM\_MSC. Tuy nhiên trong lúc thực thi ta cũng nhận ra rằng vấn đề của thuật toán này là tốc độ thực thi lâu (nhất) so với các thuật toán khác tuy nhiên độ chính xác chênh lệch không quá lớn.

* + 1. Nhận xét

Nếu hệ thống hướng tới việc gợi ý chính xác nhất, ta sẽ dùng thuật toán GCSLIM\_MCS hay các thuật toán MCS vì đây là hướng phát triển mới nhất trong hệ gợi ý hiện nay.

Tuy nhiên nếu nhắm vào tốc độ thực thi, thì các thuật toán trung bình có áp dụng ngữ cảnh như là UserAverge hay các thuật toán gợi ý dựa trên sai lệch như CAMF\_C cũng là một lựa chọn tốt nhất về tốc độ.

**Kết luận**

* **Thành công**
* Hệ thống sử dụng tốt ngữ cảnh nhưng thu thập ngữ cảnh lại là một bài toán hoàn toàn khác cần được giải quyết.
* Hệ thống rất dễ sử dụng, không phụ thuộc vào hệ điều hành, chỉ cần máy ảo Java có phiên bản 8 trở lên.
* Hệ thống đã áp dụng thành công vào website du lịch thông minh và có kết quả.
* **Hạn chế**
* Dữ liệu vào ban đầu cho hệ gợi ý vẫn là một vấn đề hết sức trầm trọng. “Garbage in garbage out”. Đối với một hệ gợi ý thì dữ liệu gần như là nguồn sống.
* Ngữ cảnh còn phụ thuộc quá nhiều vào người dùng, từ đó dữ liệu sẽ lỗi nhiều.
* Hệ thống chỉ đưa ra thuật toán tốt nhất, nhưng cách tạo ra kết quả gợi ý còn chậm, chưa thật sự trở thành một hệ thống có thể đưa ra gợi ý tức thời.
* Hệ thống còn một số lỗi trên một số thuật toán, vì các thuật toán đó yêu cầu bộ dữ liệu đặc biệt (phải có dữ liệu khuyết – NA – chẳng hạn).
* **Hướng phát triển trong tương lai**
* Sửa toàn bộ lỗi cho tất cả thuật toán.
* Tăng tốc độ học, học tiếp tục từ dữ liệu thay vì học lại từ đầu
* Nếu có lỗi sẽ in lỗi rõ ràng ra màn hình (hiện nay thông qua 1 tệp log)
* Viết mô tả rõ hơn cho mỗi thuật toán.

TÀI LIỆU THAM KHẢO

1. Yong Zheng, Bamshad Mobasher, Robin Burke, “CARSKit: A Java-Based Context-aware Recommendation Engine,” Proceedings of the 15th IEEE International Conference on Data Mining (ICDM) Workshops, Atlantic, NJ, Mỹ, Tháng 11, 2015.
2. G. Adomavicius, B. Mobasher, F. Ricci, A. Tuzhilin, “Context-aware recommender systems,” AI Magazine 32 (3) (2011) 67–80.
3. Francesco Ricci, Lior Rokach và Bracha Shapira, “Recommender Systems Handbook,” Springer, 2011.
4. B. van Kortenhof, “Context-Aware Recommender Systems in the E-commerce Domain,” TU Delft, 2017.
5. G. A. v. A. Tuzhilin, “Context-Aware Recommender Systems,” Springer, Boston, 2015.
6. Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, và Pete Steggles, “Towards a better understanding of context and contextawareness,” International Symposium on Handheld and Ubiquitous Computing, Springer, 1999.
7. R. W. Belk, “An exploratory assessment of situational effects in buyer behavior,” Journal of marketing research, pp. 156-163, 1974.
8. Yong Zheng, Bamshad Mobasher, và Robin Burke, “Correlation-Based Contextaware Matrix Factorization,” trong Proceedings of School of Computing Research Symposium, Chicago, 2015.
9. L. Baltrunas, B. Ludwig, S. Peer, F. Ricci, “Context-aware places of interest recommendations for mobile users,” in: Design, User Experience, and Usability. Theory, Methods, Tools and Practice, Springer, 2011, pp. 531–540.
10. Y. Zheng, R. Burke, B. Mobasher, “Differential context relaxation for context-aware travel recommendation,” in: E-Commerce and Web Technologies, Springer, 2012, pp. 88–99.
11. Y. Zheng, R. Burke, B. Mobasher, “Differential context modeling in collaborative filtering,” in: Proceedings of School of Computing Research Symposium, DePaul University, Chicago IL, USA, 2013.