TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**MÔN KHAI PHÁ WEB**

**BÁO CÁO BÀI TẬP LỚN**

**Đề Tài: Xây Dựng Hệ Gợi Ý Phim Sử Dụng**

**Item-Based Collaborative Filtering**

|  |  |
| --- | --- |
|  |  |
|  |  |

*Hà Nội, 06/2020*

**Mục Lục**

[1. Đặt vấn đề , lý do chọn đề tài 3](#_Toc42619853)

[1.1. Hệ gợi ý là gì? 3](#_Toc42619854)

[1.2. Lý do chọn đề tài 3](#_Toc42619855)

[2. Mô tả đề tài 3](#_Toc42619856)

[2.1. Tìm hiểu về Lọc cộng tác ( Collaborative Filtering ) 4](#_Toc42619857)

[2.2. Lọc cộng tác dựa trên vật phẩm ( Item-based collaborative filtering) 6](#_Toc42619858)

[2.2.1. Tính độ tương đồng 6](#_Toc42619859)

[2.2.2. Dự đoán: Phương pháp tổng trọng số (Weighted Sum) 7](#_Toc42619860)

[2.3. Các độ đo đánh giá 7](#_Toc42619861)

[2.3.1. Độ đo MAE (Mean Absolute Error) 7](#_Toc42619862)

[2.3.2. Độ đo NMAE 7](#_Toc42619863)

[2.3.3. Độ đo RMSE (Root Mean Squared Error) 8](#_Toc42619864)

[3. Dữ liệu 8](#_Toc42619865)

[4. Cài đặt 8](#_Toc42619866)

[4.1.1. Đọc dữ liệu và chuẩn hóa dữ liệu 8](#_Toc42619867)

[4.1.2. Tính similarity 10](#_Toc42619868)

[4.1.3. Hàm dự đoán 10](#_Toc42619869)

[4.1.4. Hàm đánh giá 11](#_Toc42619870)

[5. Kết quả 12](#_Toc42619871)

[6. Phân chia công việc 13](#_Toc42619872)

# Đặt vấn đề , lý do chọn đề tài

## Hệ gợi ý là gì?

Hệ gợi là một phần của một hệ thống thông tin . Mục đích của nó là hỗ trợ người dùng tìm kiếm được đúng thông tin cần thiết, dự đoán sở thích hay xếp hạng mà người dùng có thể dành cho một mục thông tin (item) nào đó mà họ chưa gặp trong quá khứ.

## Lý do chọn đề tài

Dưới đây là một số ví dụ về công dụng của hệ gợi ý trong các hệ thống cụ thể :

• Youtube tự động chuyển các clip liên quan đến clip bạn đang xem. Youtube cũng tự gợi ý những clip mà có thể bạn sẽ thích.

• Khi bạn mua một món hàng trên Amazon, hệ thống sẽ tự động gợi ý “Frequently bought together”, hoặc nó biết bạn có thể thích món hàng nào dựa trên lịch sử mua hàng của bạn.

• Facebook hiển thị quảng cáo những sản phẩm có liên quan đến từ khoá bạn vừa tìm kiếm.

• Facebook gợi ý kết bạn.

• Netflix tự động gợi ý phim cho người dùng.

Việc sử dụng hệ gợi ý giúp cho hiệu quả của việc Marketing tăng lên một cách đáng kể . Người dùng có được sự thuận tiện nhanh chóng phù hợp với nhu cầu của mình, người cung cấp sản phẩm tăng được doanh thu.

Trong học phần này nhóm em xin được chọn đề tài Xây dựng Hệ gợi ý phim dựa trên Item-Based Collaborative Filtering.

# Mô tả đề tài

Hệ gợi ý thường được chia thành 2 nhóm lớn:

* Collaborative filtering (Lọc cộng tác)
* Content-based filtering (Lọc dựa trên nội dung)

Đặc điểm của Content-based Recommendation Systems là việc xây dựng mô hình cho mỗi user không phụ thuộc vào các users khác mà phụ thuộc vào profile của mỗi items. Việc làm này có lợi thế là tiết kiệm bộ nhớ và thời gian tính toán. Đồng thời, hệ thống có khả năng tận dụng các thông tin đặc trưng của mỗi item như được mô tả trong bản mô tả (description) của mỗi item. Bản mô tả này có thể được xây dựng bởi nhà cung cấp hoặc được thu thập bằng cách yêu cầu users gắn tags cho items. Việc xây dựng feature vector cho mỗi item thường bao gồm các kỹ thuật Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP).

Content-based Recommendation Systems có hai nhược điểm cơ bản.

* Thứ nhất, khi xây dựng mô hình cho một user, các hệ thống Content-based không tận dụng được thông tin từ các users khác. Những thông tin này thường rất hữu ích vì hành vi mua hàng của các users thường được nhóm thành một vài nhóm đơn giản; nếu biết hành vi mua hàng của một vài users trong nhóm, hệ thống nên suy luận ra hành vi của những users còn lại.
* Thứ hai, không phải lúc nào chúng ta cũng có bản mô tả cho mỗi item. Việc yêu cầu users gắn tags còn khó khăn hơn vì không phải ai cũng sẵn sàng làm việc đó; hoặc có làm nhưng sẽ mang xu hướng cá nhân. Các thuật toán NLP cũng phức tạp hơn ở việc phải xử lý các từ gần nghĩa, viết tắt, sai chính tả, hoặc được viết ở các ngôn ngữ khác nhau.

Những nhược điểm phía trên có thể được giải quyết bằng Collaborative Filtering (CF).

Ảnh có chứa văn bản, bản đồ

Mô tả được tạo với mức tin cậy rất cao

## Tìm hiểu về Lọc cộng tác ( Collaborative Filtering )

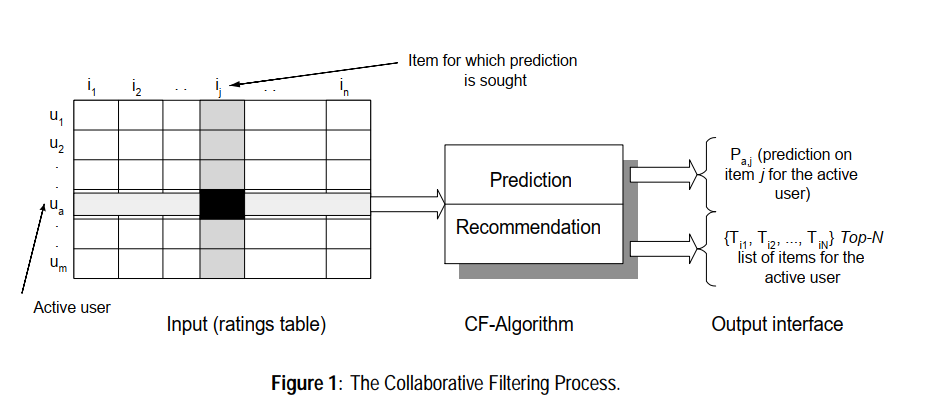
*Collaborative filtering*: hệ thống gợi ý *items* dựa trên sự tương quan (similarity) giữa các *users* hoặc *items*. Có thể hiểu rằng ở nhóm này một *item* được *recommended* tới một *user* dựa trên những *users* có *hành vi* tương tự.

Việc gần giống nhau giữa các users có thể được xác định thông qua mức độ quan tâm của các users này tới các items khác mà hệ thống đã biết. Ví dụ, A, B đều thích phim Cảnh sát hình sự, tức đều rate bộ phim này 5 sao. Ta đã biết A cũng thích Người phán xử, vậy nhiều khả năng B cũng thích bộ phim này.

Cho danh sách của m users U= {u1, u2, . . . , um }

và danh sách của n items I= {i1, i2, . . . , im }

Mỗi user ui có một danh sách các item Iui là mức độ quan tâm của người dùng ui tới item ij. Tồn tại một người dung ua có giá trị này rỗng và nhiệm vụ của việc lọc cộng tác để đề xuất các item mới hoặc dự đoán mức độ quan tâm của một item j nào đó.



Có hai thực thể chính trong Recommendation Systems là users và items.

Bài toán Lọc cộng tác được chia ra thanh 2 loại:

* Lọc cộng tác dựa trên user ( User-based collaborative filtering)
* Lọc cộng tác dựa trên vật phẩm ( Item-based collaborative filtering)

Ý tưởng cơ bản của User-based CF là xác định mức độ quan tâm của một user tới một item dựa trên các users khác gần giống với user này. Việc gần giống nhau giữa các users có thể được xác định thông qua mức độ quan tâm của các users này tới các items khác mà hệ thống đã biết.

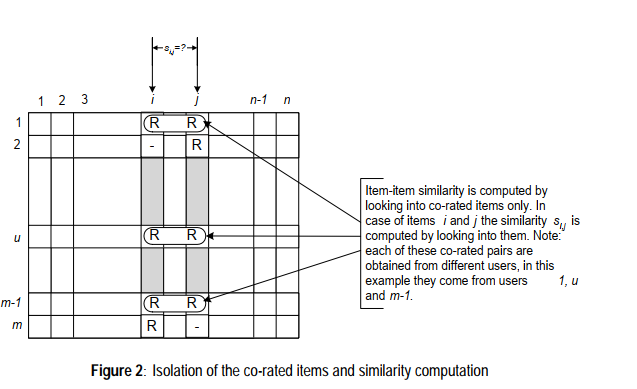
Item-based CF thay vì xác định user similarities, hệ thống sẽ xác định item similarities. Từ đó, hệ thống gợi ý những items gần giống với những items mà user có mức độ quan tâm cao.

Một số hạn chế của User-user CF:

* Trên thực tế, số lượng users luôn lớn hơn số lượng items rất nhiều. Kéo theo đó là Similarity matrix là rất lớn. Việc này, khiến cho việc lưu trữ ma trận này trong nhiều trường hợp là không khả thi.
* Ma trận Utility thường là rất thưa (sparse). Với số lượng users rất lớn so với số lượng items, rất nhiều cột của ma trận này sẽ rất sparse, tức chỉ có một vài phần tử khác 0. Lý do là users thường lười rating. Cũng chính vì việc này, một khi user đó thay đổi rating hoặc rate thêm items, trung bình cộng các ratings cũng như vector chuẩn hoá tương ứng với user này thay đổi nhiều. Kéo theo đó, việc tính toán ma trận Similarity, vốn tốn nhiều bộ nhớ và thời gian, cũng cần được thực hiện lại.

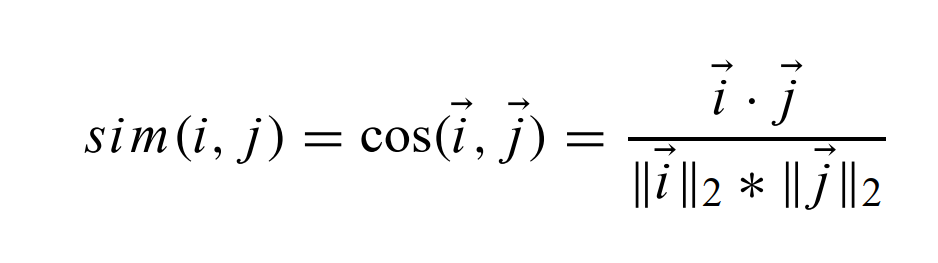
## Lọc cộng tác dựa trên vật phẩm ( Item-based collaborative filtering)

### Tính độ tương đồng

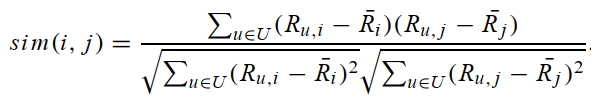


Độ tương tự của vật phẩm được tính bằng cách chỉ xét các vật phẩm được cùng xếp hạng. Trong trường hợp vật phẩm i và j độ tương tự si, j được tính bằng cách xem xét các xếp hạng của những người dùng cùng đánh giá i, j. Trong ví dụ này họ đến từ người dùng 1, u và m-1.

*Độ tương đồng dựa trên Cosin*



*Độ tương đồng Cosin dựa trên vật phẩm*

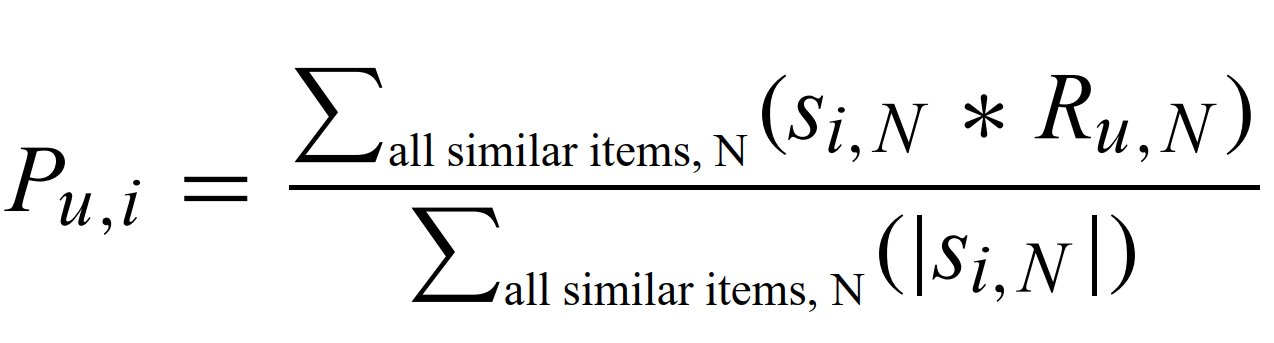


U là tập người dùng đánh giá sản phẩm i, j.

Ru,i: giá trị rating của user u cho item i.

Ri: giá trị rating trung bình của item i

### Dự đoán: Phương pháp tổng trọng số (Weighted Sum)



trong đó N là tập hợp k item trong neighborhood (tức có similarity cao nhất) của u mà đã rated i.

## Các độ đo đánh giá

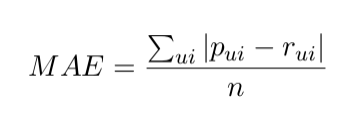
Cho:

* Tập người dùng U
* Tập sản phẩm I

Dữ liệu gồm có

* u: người dùng u ∈ U
* i: sản phẩm i ∈ I
* Rui: Đánh giá của người dùng u đối với sản phẩm i
* t: thời gian đánh giá

### Độ đo MAE (Mean Absolute Error)

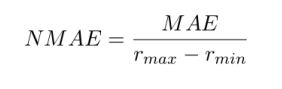


pui: dự đoán của người dùng u cho item i

rui: đánh giá của người dùng u cho item i

n: tổng số ví dụ trong tập test

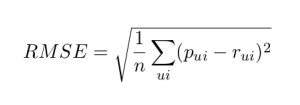
### Độ đo NMAE



rmax: giá trị dự đoán lớn nhất của người dùng

rmin: giá trị dự đoán bé nhất của người dùng

### Độ đo RMSE (Root Mean Squared Error)



Các tiêu chí dựa vào sai số trung bình phương nhạy hơn với chuỗi có nhiều ngoại lai (outliers)

# Dữ liệu

Bộ dữ liệu MovieLens với 100 000 xếp hạng (mỗi người dùng đánh giá ít nhất 20 bộ phim). Tập dữ liệu đã được chuyển đổi thành ma trận mục người dùng *A* có 943 hàng (nghĩa là 943 người dùng) và 1682 cột (tức là 1682 phim được đánh giá bởi ít nhất một trong số những người dùng).

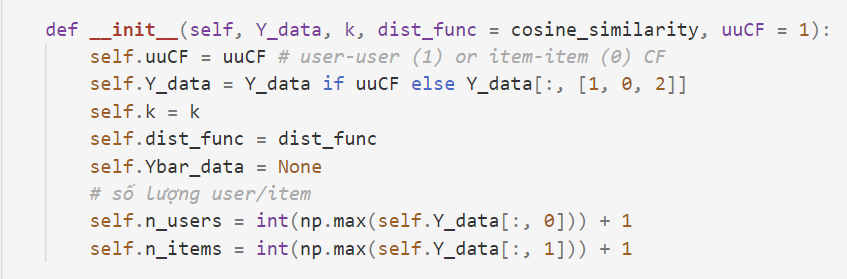
Độ thưa thớt của ma trận Utility = 1 - = 0,9369

# Cài đặt

Nguồn tham khảo:

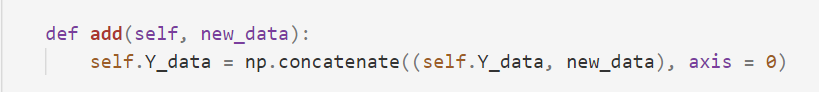
<https://github.com/tiepvupsu/tiepvupsu.github.io/blob/master/assets/24_collaborativefiltering/python/CF.ipynb>

### Đọc dữ liệu và chuẩn hóa dữ liệu

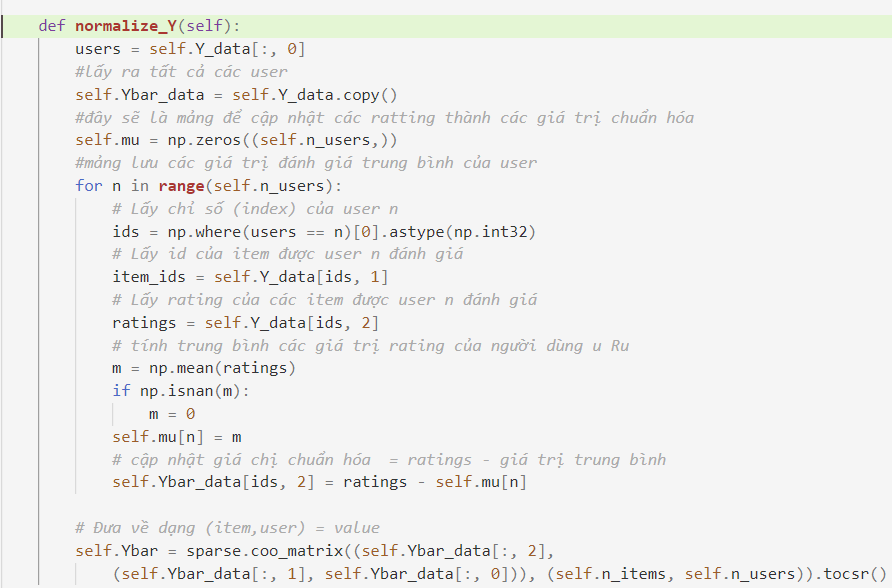


Khởi tạo các giá trị ban đầu, Dữ liệu đầu vào của hàm khởi tạo là ma trận Utility Y\_data được lưu dưới dạng một ma trận với 3 cột, k là số lượng các điểm lân cận được sử dụng để dự đoán kết quả. dist\_func là hàm đó similarity giữa hai vectors, mặc định là cosine\_similarity được lấy từ sklearn.metrics.pairwise.

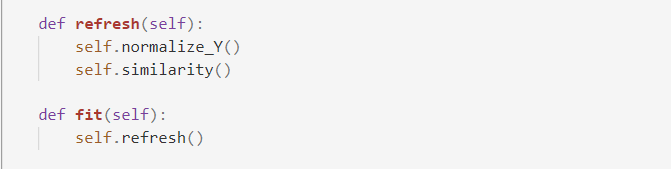
Biến uuCF thể hiện việc đang sử dụng User-user CF (1) hay Item-item CF(0).



**Khi có dữ liệu mới, cập nhận Utility matrix** bằng cách thêm các hàng này vào cuối Utility Matrix.

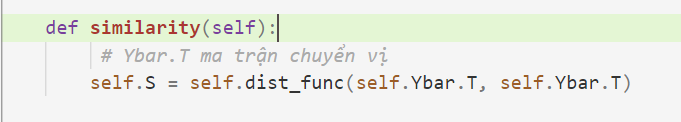


Chuẩn hóa các giá trị rating bằng cách trừ đi giá trị rating trung bình của user hoặc item.

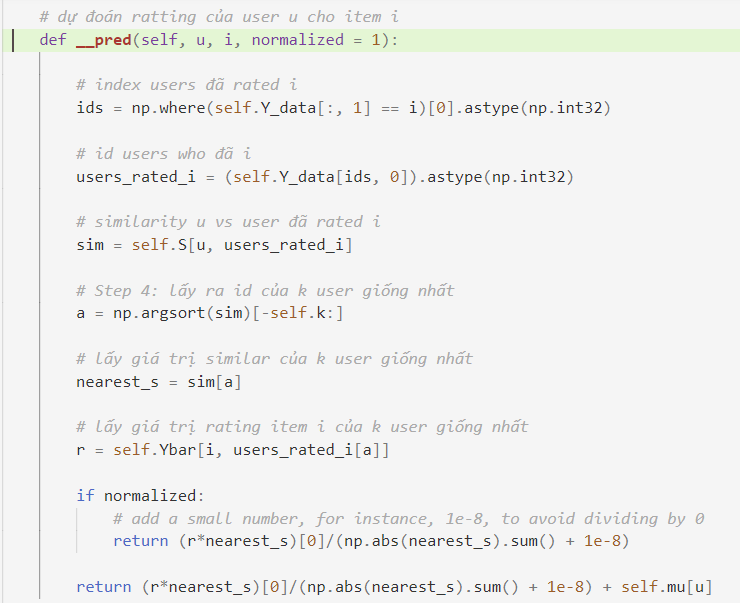


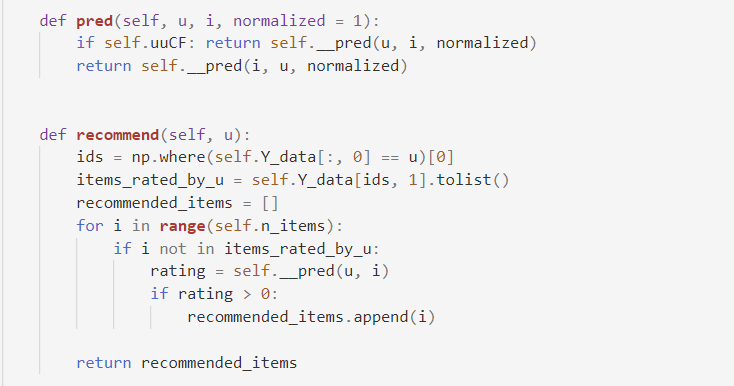
Thực hiện 2 hàm này khi có thêm dữ liệu

### Tính similarity



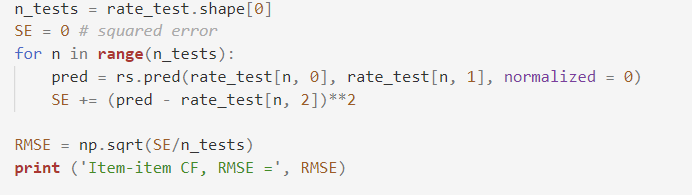
### Hàm dự đoán





Hàm recommend đối với user u, tìm ra cá item i chưa được user u đánh giá để tính ra giá trị rating bằng cách gọi hàm pred(u,i).

### Hàm đánh giá



# Kết quả

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/Test | 0.2 | 0.25 | 0.3 | 0.4 |
| User user CF | 0,958826701 | 0,956231118 | 0,967510435 | 0,973873031 |
| Item item CF | 0,954959186 | 0,95236597 | 0,959534485 | 0,967548857 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K | 10 | 20 | 30 | 40 | 50 | 100 | 125 | 150 |
| User user CF | 0,968193348 | 0,962942963 | 0,956231118 | 0,960966673 | 0,966896442 | 0,969291 | 0,970837 | 0,979125 |
| Item item CF | 0,959223487 | 0,95765452 | 0,95236597 | 0,960004259 | 0,964983458 | 0,96674 | 0,965878 | 0,968959 |

# Phân chia công việc

|  |  |
| --- | --- |
| Thành viên | Công việc |
| Dương | Làm báo cáo, Thuyết trình, Chạy code |
| Thắng | Chạy code, Tìm hiểu thuật toán KNN |
| Linh | Làm slide, Tìm hiểu thuật toán KNN |
| Sáng | Làm báo cáo, Tìm hiểu các độ đo đánh giá |