;

TRƯỜNG ĐẠI HỌC SƯ PHẠM TPHCM

KHOA CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN NHÓM**

**HỆ THỐNG TƯ VẤN THÔNG TIN**

**MATRIX FACTORIZATION COLLABORATIVE FILTERING**

**AND**

**CONTENT BASED RECOMMENDATION (CBR)**

;

TRƯỜNG ĐẠI HỌC SƯ PHẠM TPHCM

KHOA CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN NHÓM**

**HỆ THỐNG TƯ VẤN THÔNG TIN**

**Matrix Factorization Collaborative Filtering**

**And**

**Content Based Recommendation (CBR)**

DANH SÁCH NHÓM:

-Nguyễn Văn An 43.01.104.002

-Trần Minh Trường 43.01.104.192

- Nguyễn Doãn Tứ 43.01.104.196

Mục lục

[I. Cơ sở lý thuyết 4](#_Toc43310187)

[1. Giới thiệu: 4](#_Toc43310188)

[2. Ý tưởng: 5](#_Toc43310189)

[3. Thuật toán 6](#_Toc43310190)

[II. Thực hành (6đ) 6](#_Toc43310191)

[1. Hướng dẫn cài đặt: 6](#_Toc43310192)

[2. ... 6](#_Toc43310193)

[III. Kết luận 6](#_Toc43310194)

[1. Kết quả đạt được 6](#_Toc43310195)

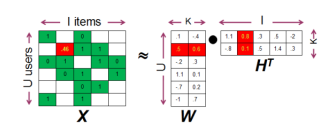
[2. Khó khăn và thuận lợi 6](#_Toc43310196)

# Cơ sở lý thuyết

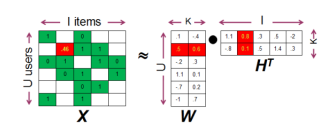
## Giới thiệu:

Kỹ thuật phân rã ma trận (MF) là việc chia một ma trận lớn **X** thành 2 ma trận có kích thước nhỏ hơn rất nhiều so với ma trận ban đầu **W** và **H**, sao cho X có thể được xây dựng lại từ hai ma trận nhỏ hơn này càng chính xác càng tốt, nghĩa là

như hình minh hoạ (Hình I.1):

Hình I.1:

Trong đó, **X** là tập hợp tất cả các đánh giá (rating) của người dùng (user) với mục tin (item), bao gồm cả những giá trị chưa biết cần được dự đoán tạo nên một ma trận gọi là Utility Matrix. là một ma trận mà mỗi dòng *u* là một véc tơ bao gồm *K* nhân tố tiềm ẩn (latent factors) mô tả cho user *u*, vàlà một ma trận mà mỗi dòng *i* là một véc tơ bao gồm K nhân tố tiềm ẩn mô tả cho item *I (*Hình I.2).

Hình I.2

## Ý tưởng:

Ý tưởng chính đằng sau Matrix Factorization cho Recommendation Systems là tồn tại các *latent features* (tính chất ẩn) mô tả sự liên quan giữa các

*items* và *users*. Ví dụ với hệ thống gợi ý các bộ phim, tính chất ẩn có thể là *hình sự*, *chính trị*, *hành động*, *hài*, …; cũng có thể là một sự kết hợp nào đó của các thể loại này; hoặc cũng có thể là bất cứ điều gì mà chúng ta không thực sự cần đặt tên. Mỗi *item* sẽ mang tính chất ẩn ở một mức độ nào đó tương ứng với các hệ số trong vector **x** của nó, hệ số càng cao tương ứng với việc mang tính chất đó càng cao. Tương tự, mỗi *user* cũng sẽ có xu hướng thích những tính chất ẩn nào đó và được mô tả bởi các hệ số trong vector **w** của nó. Hệ số cao tương ứng với việc *user* thích các bộ phim có tính chất ẩn đó. Giá trị của biểu thức **xw** sẽ cao nếu các thành phần tương ứng của **x** và **w** đều cao. Điều này nghĩa là *item* mang các tính chất ẩn mà *user* thích, vậy thì nên gợi ý *item* này cho *user* đó.

**Sử dụng hàm mất mát:** Giả sử rằng số *users* là **N**, số *items* là **M**, ***utility maxtrix***được mô tả bởi ma trận **Y**. Thành phần ở hàng thứ **m**, cột thứ **n** của **Y** là *mức độ quan tâm* (ở đây là số sao đã *rate*) của *user* thứ **n** lên sản phẩm thứ **m** mà hệ thống đã thu thập được. Ma trận **Y** bị khuyết rất nhiều thành phần tương ứng với các giá trị mà hệ thống cần dự đoán. Thêm nữa, gọi R là ma trận *rated or not* thể hiện việc một *user* đã *rated* một *item* hay chưa. Cụ thể, **rij** bằng 1 nếu *item* thứ i đã được *rated* bởi *user* thứ j, bằng 0 trong trường hợp ngược lại.

**Content Based Recommendation (CBR)**: Để xây dựng đề xuất cá nhân hóa, nhóm chúng tôi tính toán sự giống nhau giữa các bộ phim dựa trên các số liệu nhất định. Sau đó, chúng ta có thể suy luận / dự đoán một cách an toàn những bộ phim mà người dùng sẽ thích dựa trên bộ phim mà người dùng yêu thích trước đó. Điều này được thực hiện bằng cách sử dụng số liệu tương tự dựa trên các thuộc tính / nội dung của phim. Do đó nó được gọi là Content Based Recommentdation. Chúng tôi có thể xây dựng mô hình CBR dựa trên tất cả hoặc một tập hợp con các thuộc tính của phim.

Giả sử, chúng tôi xây dựng một mô hình CBR với tên phim là một thuộc tính. Ví dụ: một bộ phim ‘Love Actually’ và ‘Love Story’ sẽ rơi vào thể loại 'Tình cảm'. Do đó, dựa trên tên phim, chúng ta có thể phân cụm các phim có tên tương tự. Tuy nhiên, đây KHÔNG phải là cách tiếp cận tốt nhất vì nó cũng có thể gây hiểu nhầm. Ví dụ: một bộ phim Gone With The Wind' (Lãng mạn) có thể không liên quan gì đến ' Wind Chill' (Kinh dị).

**Collaborative Filtering (CF):** Qua Content Based Recommendation (CBR),Chúng tôi thấy rằng các dự đoán không quá chính xác vì một số thể loại khác nhau rất nhiều. Do đó, cách tiếp cận khai thác văn bản sẽ hữu ích hơn, nếu chúng tôi được cung cấp thêm chi tiết về các bộ phim như tóm tắt phim, diễn viên, đoàn làm phim, v.v. Mô hình này không đưa ra khuyến nghị cụ thể cho người dùng. Do đó, để khắc phục giới hạn này phát sinh do dữ liệu mô tả hạn chế có sẵn trong bộ dữ liệu MovieLens 100k, chúng tôi tính toán tùy chọn của user1 dựa trên tùy chọn khác của user2 trên các phim mà phim sau đã xem nhưng phim trước không. Cách tiếp cận này được gọi là Collaboảtive Filtering (CF).

## Thuật toán

# Thực hành

## Hướng dẫn cài đặt:

* Tài nguyên cần chuẩn bị:
  + Dataset: [Movielens 100k](https://www.kaggle.com/prajitdatta/movielens-100k-dataset/download)
  + Python: [Python 3.7.4](https://www.python.org/downloads/release/python-374/)
  + Môi trường: Jupyter notebook
  + Source dự án: [GitHub](https://github.com/tund-hcmue/Matrix_Factorization)
* Các bước cài đặt:
  + Cài đặt thư viện: pip install -r requirements.txt

## ...

# Kết luận

## Kết quả đạt được

## Khó khăn và thuận lợi

# Tài liệu tham khảo

Matrix Factorization, “Matrix Factorization Collaborative Filtering”, [Online], Available:

<https://machinelearningcoban.com/2017/05/31/matrixfactorization/?fbclid=IwAR2_t4PU60HtTQqUz5_hZvtOFzEmRoY3fi29TK8OSBbOZLo7Bs9aZXsav-g>