# NHẬP MÔN KHOA HỌC DỮ LIỆU

## HỆ KHUYẾN NGHỊ

ThS. Vũ Hoài Thư

# Nội dung

# Giới thiệu

# Tại sao cần hệ gợi ý

- Người dùng bị quá tải thông tin trong môi trường web
- Nhà bán hàng cần đưa ra sản phẩm phù hợp để:
  - Tăng doanh số bán hàng
  - Nâng cao chất lượng dịch vụ
- Xu hướng cá nhân hóa và số hóa là tất yếu

# Định nghĩa hệ khuyến nghị

- Hệ khuyến nghị là một hệ thống dựa trên học máy giúp cho người dùng khám phá những sản phẩm và dịch vụ mới.
- Hệ khuyến nghị đang được sử dụng để hướng người dùng đến những sản phẩm mà họ muốn mua nhất dựa trên những dữ liệu về hành vi người dùng đã thu thập từ trước.

# Lĩnh vực ứng dụng

- Thương mại điện tử
- Giải trí trực tuyến
- Tin tức trực tuyến
- Forum, mạng xã hội
- Nghiên cứu khoa học
- Hẹn hò trực tuyến

# Lĩnh vực ứng dụng (Tiếp)

1. Amazon:
   - Gợi ý sản phẩm
   - Tăng hơn 30% doanh thu
2. Netflix:
   - Gợi ý phim, chương trình TV
   - Mang về $1B mỗi năm
3. Google News::
   - Gợi ý tin tức
   - Tăng gần 40% lưu lượng truy cập

# Các phương pháp gợi ý

1, Gợi ý dựa trên nội dung: Gợi ý dựa trên lịch sử giao dịch của người dùng.
2, Lọc cộng tác: Gợi ý dựa trên người dùng có sở thích tương tự.
3, Gợi ý dựa trên phiên: Gợi ý dựa trên chuỗi giao dịch.
4, Các phương pháp lai

# Những thách thức của hệ gợi ý

- Số giao dịch rất nhỏ so với số lượng người dùng và sản phẩm thực tế
- Không đủ thông tin về người dùng và sản phẩm mới
- Người dùng và sản phẩm thay đổi theo thời gian, theo mùa
- Thói quen tiêu dùng thay đổi theo thời gian, theo mùa
- Gợi ý theo thời gian thực

# Hệ khuyến nghị hoạt động như thế nào

- Các hệ khuyến nghị hoạt động dựa trên nguyên tắc hiểu rõ các quan hệ của các thực thể trong hệ thống.
- Có 3 loại quan hệ hay được sử dụng trong hệ khuyến nghị:
  (1) quan hệ giữa người dùng và sản phẩm;
  (2) quan hệ giữa sản phẩm và sản phẩm;
  (3) quan hệ giữa người dùng và người dùng.
- Mục đích chính của các hệ khuyến nghị là dự đoán mức độ quan tâm của một người dùng tới một sản phẩm nào đó.

# Dữ liệu cho bài toán gợi ý

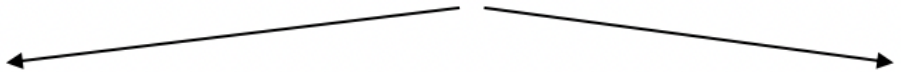
- Cho dữ liệu:
  - Tập người dùng $U$
  - Tập sản phẩm $I$
- Dữ liệu gồm các giao dịch $(u, i, r_{ui}, t)$. Trong đó:
  - $u$: người dùng $u \in U$
  - $i$: sản phẩm $i \in I$
  - $r_{ui}$: đánh giá của người dùng $u$ đối với sản phẩm $i$.
  - $t$: thời gian đánh giá

# Các phương pháp đánh giá hệ gợi ý

- $r_{ui}$ có thể theo thang đo 5 bậc (1,2,3,4,5) hoặc theo thang đo nhị phân (0,1).
- Dữ liệu được chia thành hai tập huấn luyện (train) và kiểm thử (test).
- Hệ gợi ý được huấn luyện trên tập train.
- Trên tập test, hệ gợi ý dự đoán đánh giá $p_{ui}$ của người dùng $u$ với sản phẩm $i$.

# Ví dụ

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ |
|---|---|---|---|---|
| $u_1$ | - | 5 | 3 | - |
| $u_2$ | 4 | - | 2 | 3 |
| $u_3$ | 4 | 1 | - | 5 |



| | $i_1$ | $i_2$ | $i_3$ | $i_4$ |
|---|---|---|---|---|
| $u_1$ | - | 5 | | - |
| $u_2$ | | - | 2 | 3 |
| $u_3$ | 4 | 1 | - | |

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ |
|---|---|---|---|---|
| $u_1$ | - | | 3 | - |
| $u_2$ | 4 | - | | |
| $u_3$ | | | - | 5 |

# Các độ đo đánh giá hệ gợi ý

Các độ đo được sử dụng để đánh giá hệ gợi ý:

- MAE
- NMAE
- RMSE
- Precision/Recall/ F-score

# MAE

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_{ui} - r_{ui}|$$

Trong đó:

- $p_{ui}$: Dự đoán của mô hình đối với đánh giá của người dùng $u$ với sản phẩm $i$
- $r_{ui}$: Đánh giá của người dùng $u$ đối với sản phẩm $i$
- $n$: Tổng số mẫu trong tập test

# NMAE

$$NMAE = \frac{MAE}{r_{\max} - r_{\min}}$$

Trong đó:

- $r_{\max}$: Giá trị dự đoán lớn nhất của người dùng
- $r_{\min}$: Giá trị dự đoán bé nhất của người dùng

# RMSE

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_{ui} - r_{ui})^2}$$

Trong đó:

- $p_{ui}$: Dự đoán của mô hình đối với đánh giá của người dùng $u$ với sản phẩm $i$
- $r_{ui}$: Đánh giá của người dùng $u$ đối với sản phẩm $i$
- $n$: Tổng số mẫu trong tập test

# Lọc dựa trên nội dung

# Lọc dựa trên nội dung

- Lọc dựa trên nội dung là một trong hai nhóm lớn của các hệ khuyến nghị.
- Lọc dựa trên nội dung đánh giá đặc tính của các sản phẩm được khuyến nghị.
- Ví dụ: Một người xem nhiều phim hành động, vậy hệ thống sẽ gợi ý một bộ phim có chung đặc tính *hành động* tới người dùng trong cơ sở dữ liệu phim.

# Utility matrix

- Có hai thực thể chính tồn tại ở trong các hệ khuyến nghị là người dùng (user) và sản phẩm (item).
- Mức độ quan tâm của người dùng đến sản phẩm được đo bằng giá trị mà người dùng đánh giá cho sản phẩm (rating).
- Tập hợp tất cả các ratings, bao gồm cả giá trị chưa biết cần phải dự đoán, tạo nên một ma trận gọi là utility matrix.

# Utility matrix (Tiếp)

Giả sử có 4 người dùng P1, P2, P3, P4 và 5 bộ phim. Utility matrix cho ví dụ này thể hiện ở bảng dưới đây:

| | P1 | P2 | P3 | P4 |
|---|---|---|---|---|
| **Nhiệm vụ bất khả thi** | 4 | 5 | 1 | 1 |
| **Avatar: Dòng chảy của nước** | 5 | 4 | | |
| **La la land** | | | 4 | 5 |
| **Me before you** | 1 | 2 | 5 | |
| **Vệ binh dải ngân hà** | 4 | | 2 | 3 |

# Utility matrix (Tiếp)

- Trong thực tế, số lượng users và items là rất lớn trong hệ thống, mỗi user thường đưa ra rất ít đánh giá về các items => Utility matrix là ma trận thưa.
- Tuy nhiên, việc xây dựng ma trận này thường có gặp nhiều khó khăn.
- Có hai hướng tiếp cận phổ biến để xác định giá trị rating cho mỗi vị trí trong Utility Matrix:
  - Nhờ người dùng đánh giá càng nhiều càng tốt
  - Dựa vào hành vi của người dùng để điền thêm vào các vị trí chưa có ratings.

# Hồ sơ của sản phẩm (item profiles)

- Trong các hệ thống khuyến nghị dựa trên nội dung, cần xây dựng một bộ hồ sơ của từng sản phẩm (item profiles)
- Hồ sơ này được biểu diễn dưới dạng vector đặc trưng.
- Ví dụ: các đặc trưng của một bộ phim có thể được sử dụng như diễn viên, đạo diễn, năm phát hành, thể loại, quốc gia.
- Một bộ đặc trưng tốt là bộ đặc trưng có thể mô tả chính xác được các items

# Ví dụ về item profiles

Giả sử xây dựng một vector đặc trưng 3 chiều cho các phim trong ví dụ trên, chiều thứ nhất là mức độ *hành động*, chiều thứ hai là mức độ *tình cảm*, chiều thứ ba là mức độ *hài hước*.

| | P1 | P2 | P3 | P4 | film's feature vector |
|---|---|---|---|---|---|
| **Nhiệm vụ bất khả thi** | 4 | 5 | 1 | 1 | $x_1 = [0.99, 0.02, 0.5]$ |
| **Avatar: Dòng chảy của nước** | 5 | 4 | | | $x_2 = [0.99, 0.1, 0.4]$ |
| **La la land** | | | 4 | 5 | $x_3 = [0.02, 0.98, 0.4]$ |
| **Me before you** | 1 | 2 | 5 | | $x_4 = [0.01, 0.99, 0.6]$ |
| **Vệ binh dải ngân hà** | 4 | | 2 | 3 | $x_5 = [0.99, 0.2, 0.9]$ |

# Content-based sử dụng độ đo cosine

- Dựa trên các sản phẩm mà người dùng đánh giá, xây dựng hồ sơ người dùng là trung bình (hoặc trọng số) của các vector sản phẩm đã đánh giá:

$$\mathbf{P}_u = \frac{1}{T} \sum_i (r_{ui} \times \mathbf{x}_i)$$

- Trong đó:
  - $\mathbf{P}_u$: Vector hồ sơ người dùng $u$
  - $T$: Tổng rating của người dùng $u$
  - $r_{ui}$: Điểm đánh giá của người dùng $u$ cho sản phẩm $i$

# Tính độ tương đồng và gợi ý

- Tính độ tương đồng cosine giữa hồ sơ người dùng $u$ và từng sản phẩm chưa đánh giá $j$:

$$sim(\mathbf{P}_u, \mathbf{x}_j) = \frac{\mathbf{P}_u.\mathbf{x}_j}{||\mathbf{P}_u||.||\mathbf{x}_j||}$$

- Gợi ý những sản phẩm có độ tương đồng cao nhất với hồ sơ người dùng.

# Ví dụ

Cho item profiles và rating tương ứng của người dùng về các sản phẩm. Hãy tính độ tương đồng cosine của người dùng P2 với các sản phẩm chưa đánh giá.

| | P1 | P2 | P3 | P4 | film's feature vector |
|---|---|---|---|---|---|
| **Nhiệm vụ bất khả thi** | 4 | 5 | 1 | 1 | $x_1 = [0.99, 0.02, 0.5]$ |
| **Avatar: Dòng chảy của nước** | 5 | 4 | | | $x_2 = [0.99, 0.1, 0.4]$ |
| **La la land** | | | 4 | 5 | $x_3 = [0.02, 0.98, 0.4]$ |
| **Me before you** | 1 | 2 | 5 | | $x_4 = [0.01, 0.99, 0.6]$ |
| **Vệ binh dải ngân hà** | 4 | | 2 | 3 | $x_5 = [0.99, 0.2, 0.9]$ |

# Content-Based Filtering có huấn luyện (learning-based)

Giả sử xây dựng được item profiles như hình:

| | P1 | P2 | P3 | P4 | film's feature vector |
|---|---|---|---|---|---|
| **Nhiệm vụ bất khả thi** | 4 | 5 | 1 | 1 | $x_1 = [0.99, 0.02, 0.5]$ |
| **Avatar: Dòng chảy của nước** | 5 | 4 | | | $x_2 = [0.99, 0.1, 0.4]$ |
| **La la land** | | | 4 | 5 | $x_3 = [0.02, 0.98, 0.4]$ |
| **Me before you** | 1 | 2 | 5 | | $x_4 = [0.01, 0.99, 0.6]$ |
| **Vệ binh dải ngân hà** | 4 | | 2 | 3 | $x_5 = [0.99, 0.2, 0.9]$ |

- Nhiệm vụ của bài toán là đi tìm từng mô hình người dùng $\theta_i$ để tối ưu hóa trên những ratings đã có trong bảng.
- Bài toán đi tìm mô hình $\theta_i$ cho mỗi user có thể được coi là bài toán hồi quy trong trường hợp ratings là một dải giá trị, hoặc bài toán phân loại trong trường hợp ratings là một vài trường hợp cụ thể.

# Xây dựng hàm mất mát

Giả sử có các giá trị sau:

- Số users là $N$
- Số items là $M$
- Ma trận utility là $\mathbf{Y} = \{y_{mn}\}_{M \times N}$, với $y_{mn}$ là rating của user $n$ cho sản phẩm $m$.
- Ma trận $\mathbf{R} = \{r_{ij}\}$, trong đó $r_{ij} = 1$ nếu sản phẩm $i$ được đánh giá bởi người dùng $j$ và ngược lại.

# Xây dựng hàm mất mát

- Giả sử mỗi user có thể tìm được một mô hình $\theta_i$ được minh họa bởi vector cột hệ số $\mathbf{w}_i$, và độ lệch $b_n$.
- Khi đó, mức quan tâm của người dùng tới một sản phẩm có thể biểu diễn được bằng một hàm tuyến tính:

$$\hat{y}_{mn} = \mathbf{x}_m \mathbf{w}_n + b_n$$

- Trong đó:
  - $\mathbf{x}_m$: vector đặc trưng của sản phẩm thứ $m$ (vector hàng)
  - $\mathbf{w}_n$: vector sở thích của người dùng (vector cột).

# Xây dựng hàm mất mát

- Với một user thứ $n$ bất kỳ, ta có tập hợp các thành phần đã được đánh giá tương ứng của $y_n$, hàm mất mát có thể được xây dựng như sau:

$$\mathscr{L}_n = \frac{1}{2} \sum_{m:r_{mn}=1} (x_m w_n + b_n - y_{mn})^2 + \frac{\lambda}{2} \|w_n\|_2^2$$

- Trong đó, thành phần thứ hai là regularization để ngăn cho mô hình không bị overfit.

# Xây dựng hàm mất mát

- Thông thường hàm mất mát sẽ được tính trên trung bình cộng của tất cả các vị trí có đánh giá của user nên hàm mất mát được viết lại:

$$\mathscr{L}_n = \frac{1}{2s_n} \sum_{m:r_{mn}=1} (x_m w_n + b_n - y_{mn})^2 + \frac{\lambda}{2s_n} \|w_n\|_2^2$$

Trong đó $s_n$ là số lượng các sản phẩm mà người dùng thứ $n$ đã đánh giá.

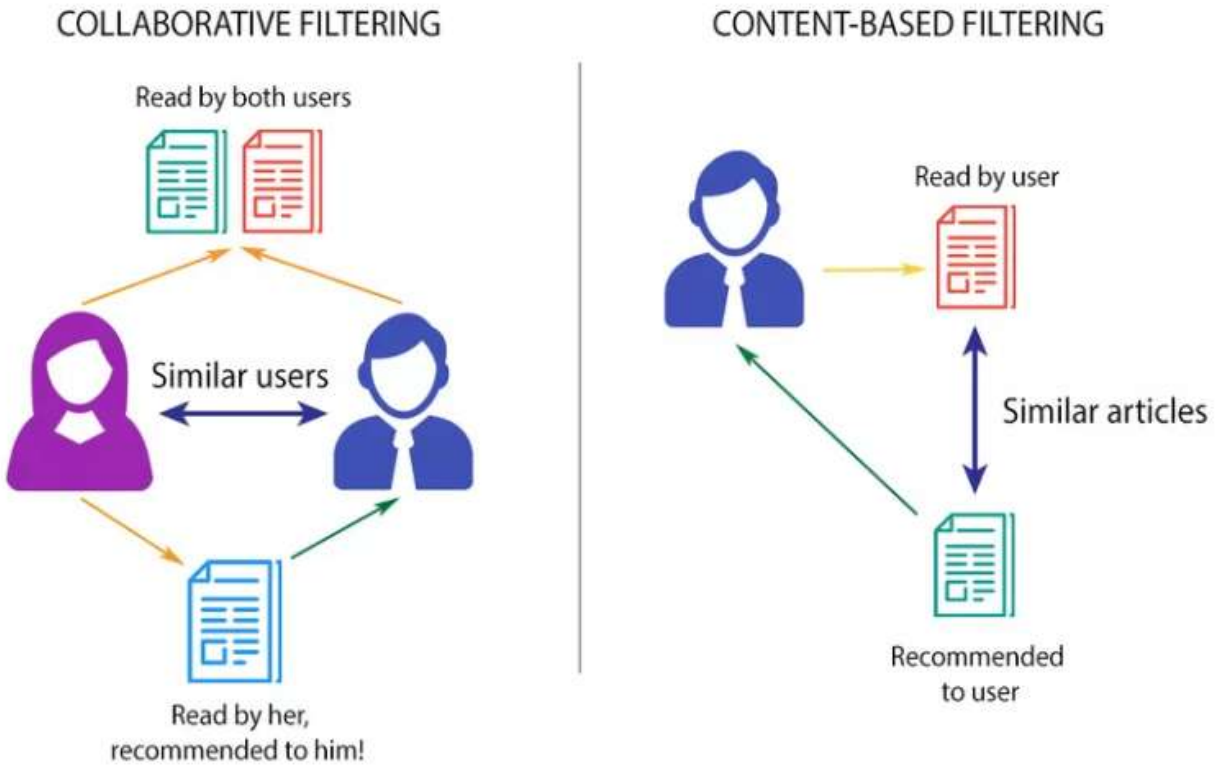
Cặp nghiệm $\mathbf{w}_n$, $b_n$ tương ứng với mô hình được tìm ra bằng việc áp dụng các phương pháp như Gradient Descent (GD), Stochastic Gradient Descent (SGD).

# Lọc cộng tác

# Lọc dựa trên nội dung và Lọc cộng tác

- Lọc dựa trên nội dung
  - Xây dựng các mô hình cho từng người dùng với dữ liệu huấn luyện là các cặp (hồ sơ sản phẩm, đánh giá) mà người dùng đã đánh giá.
  - Mô hình được xây dựng cho từng người dùng và không tận dụng được thông tin của các người dùng khác.
  - Không phải lúc nào việc trích xuất đặc trưng cho các item cũng thực hiện được.
- Lọc cộng tác
  - Lọc cộng tác được xây dựng với ý tưởng là xác định mức độ quan tâm của một người dùng tới một sản phẩm dựa trên các người dùng khác gần đặc điểm tương tự.
  - Việc xác định sự tương đồng giữa các người dùng dựa trên mức độ quan tâm của các người dùng và các sản phẩm mà hệ thống đã ghi nhận.

# Lọc dựa trên nội dung và Lọc cộng tác



# Lọc cộng tác

Các phương pháp tiếp cận:

- **Lọc cộng tác dựa trên người dùng (user-user collaborative filtering)**: Xác định sự tương đồng giữa các người dùng, sau đó dự đoán mức độ quan tâm của một người dùng tới một sản phẩm.
- **Lọc cộng tác dựa trên sản phẩm (item-item collaborative filtering)**: Tìm sự tương đồng giữa các sản phẩm sau đó gợi ý cho người dùng các sản phẩm tương tự với các sản phẩm mà người dùng có mức độ quan tâm cao.

# Lọc cộng tác dựa trên người dùng (user-user collaborative filtering)

**Ý tưởng**: Người dùng $u$ sẽ được gợi ý các sản phẩm mà những người dùng giống $u$ đã thích.
Độ tương đồng Pearson được sử dụng để đánh giá tương tự giữa người dùng $u$ và người dùng $v$:

$$sim(u,v) = \frac{\sum_{i \in C}(r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i \in C}(r_{ui} - \overline{r}_u)^2} . \sqrt{\sum_{i \in C}(r_{vi} - \overline{r}_v)^2}}$$

Trong đó:

- $C$: Tập những sản phẩm mà người dùng $u$ và $v$ đều đánh giá
- $\overline{r}_u$: Đánh giá trung bình của người dùng $u$ (chỉ tính trên các sản phẩm mà người dùng $u$ đã đánh giá)

# Dự đoán đánh giá

Khi đó, điểm dự đoán cho người dùng $u$ với sản phẩm $i$:

$$p_{ui} = \overline{r}_u + \frac{\sum_{v \in V} sim(u, v)(r_{vi} - \overline{r}_v)}{\sum_{v \in V} |sim(u, v)|}$$

Trong đó $V$: top $k$ người dùng tương tự với người dùng $u$.

# Ví dụ

Cho Utility matrix cho các người dùng từ $u_0$ đến $u_5$ đánh giá các sản phẩm từ $i_0$ đến $i_5$ như sau:

| | $u_0$ | $u_1$ | $u_2$ | $u_3$ | $u_4$ | $u_5$ |
|---|---|---|---|---|---|---|
| $i_0$ | 5 | 5 | 2 | 1 | 1 | |
| $i_1$ | 3 | | | 0 | | |
| $i_2$ | | 5 | 1 | | 1 | 2 |
| $i_3$ | 1 | 1 | 5 | 4 | 5 | 4 |
| $i_4$ | 2 | 2 | 3 | | | 5 |
| $i_5$ | 2 | 0 | 4 | 5 | | |

Hãy dự đoán rating của người dùng $u_0$ đối với sản phẩm $i_2$?

# Một số hạn chế của Lọc cộng tác dựa trên người dùng

Một số hạn chế của User-user CF:

- Thường xuyên phải cập nhật lại vector người dùng khi người dùng có giao dịch mới
- Phải tính toán trên toàn bộ tập người dùng

=> Sử dụng các tiếp cận thứ hai: Lọc cộng tác dựa trên sản phẩm (Item-item Collaborative filtering)

# Lọc cộng tác dựa trên sản phẩm (Item-item Collaborative filtering)

- Biểu diễn sản phẩm dựa trên ma trận tương tác người dùng - sản phẩm
- Phù hợp với hệ thống có số sản phẩm « số người dùng
- Ít phải cập nhật lại véc-tơ sản phẩm
- Có thể tính toán trước độ tương tự sản phẩm - sản phẩm

# Độ tương đồng giữa các sản phẩm

Hệ số tương quan Pearson được sử dụng để đánh gía tương đồng giữa hai sản phẩm $i$ và $j$:

$$sim(i,j) = \frac{\sum_{u \in U}(r_{ui} - \overline{r}_u)(r_{uj} - \overline{r}_u)}{\sqrt{\sum_{u \in U}(r_{ui} - \overline{r}_u)^2}.\sqrt{\sum_{u \in U}(r_{uj} - \overline{r}_u)^2}}$$

Trong đó:

- $U$: Tập người dùng đã đánh giá cả hai sản phẩm $i, j$
- $r_{ui}$: rating của người dùng $u$ cho sản phẩm $i$
- $\overline{r}_u$: trung bình rating của người dùng $u$

# Dự đóan rating cho sản phẩm chưa đánh giá

Dự đoán điểm mà người dùng $u$ có thể cho sản phẩm $i$:

$$p_{ui} = \frac{\sum_{j \in J} r_{uj} \times sim(i,j)}{\sum_{j \in J} |sim(i,j)|}$$

Trong đó $J$ là tập top $k$ sản phẩm tương tự với sản phẩm $i$

# Ví dụ

Giả sử có thông tin bảng dữ liệu lưu trữ xếp hạng của các người dùng đối với các sản phẩm khác nhau như sau:

| | **Sản phẩm1** | **Sản phẩm2** | **Sản phẩm3** | **Sản phẩm4** | **Sản phẩm5** |
|---|---|---|---|---|---|
| *Người A* | 5 | 4 | - | 2 | 3 |
| *Người B* | - | 3 | 5 | 4 | - |
| *Người C* | 3 | 2 | 5 | 4 | - |
| *Người D* | 4 | - | 3 | - | 5 |
| *Người E* | 3 | - | 5 | 4 | - |

Hãy dự đoán xếp hạng của người dùng D cho Sản phẩm 2?

# Lọc cộng tác dựa trên phân rã ma trận (matrix factorization)

# Giới thiệu

- Mỗi item được mô tả bởi một vector đặc trưng $\mathbf{x}$ (item profile).
- Mỗi user có một vector hệ số $\mathbf{w}$ mô tả sở thích.
- Dự đoán rating: $\hat{y} = \mathbf{xw}$
- Mục tiêu: Tìm ra $\mathbf{x}$ và $\mathbf{w}$ sao cho tổng thể các dự đoán khớp tốt nhất với rating thực tế trong utility matrix $\mathbf{Y}$
- Khác với content-based: ở đây item profile và user model được học đồng thời thay vì xây dựng thủ công.

# Nguyên lý phân rã ma trận

- Gỉa sử ma trận Utility $\mathbf{Y} \in \mathbf{R}^{M \times N}$ với $M$ là số lượng item, $N$ là số lượng user
- Tiến hành xấp xỉ ma trận $\mathbf{Y}$:

$$\mathbf{Y} \approx \mathbf{X}\mathbf{W}$$

- Trong đó:
  - $\mathbf{X} \in \mathbf{R}^{M \times K}$: ma trận đặc trưng của các items
  - $\mathbf{W} \in \mathbf{R}^{K \times N}$: ma trận đặc trưng của user
  - $K$: số chiều "tính chất ẩn" (latent features), $K << M, N$

# Ý tưởng về các đặc trưng ẩn (Latent Features)

- Giả định tồn tại các tính chất ẩn (latent features) mô tả mối liên hệ giữa users và items.
- 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 có ý nghĩa.
- Mỗi item mang các đặc tính ẩn ở mức độ khác nhau (vector $\mathbf{x}$)
- Mỗi user có xu hướng thích các đặc tính ẩn nào đó (vector $\mathbf{w}$)
- Dự đoán cao khi vector $\mathbf{x}$ và $\mathbf{w}$ "phù hợp hướng":

$$\hat{y}_{ui} = \mathbf{x}_i^T \mathbf{w_u}$$

=> Giải thích được tại sao user thích item, dựa trên các yếu tố tiềm ẩn học được từ dữ liệu.

# Quá trình huấn luyện và dự đoán

- Bài toán được tối ưu bằng cách cực tiểu hàm mất mát:

$$\mathcal{L} = \sum_{(u,i)} (r_{ui} - \mathbf{x}_i^T \mathbf{w_u})^2 + \lambda(||\mathbf{x}_i||^2 + ||\mathbf{w}_u||^2)$$

- Quá trình huấn luyện:
  1. Khởi tạo ngẫu nhiên $\mathbf{X}$ và $\mathbf{W}$
  2. Lặp lại cập nhật xen kẽ: Cố định $\mathbf{X}$, tối ưu $\mathbf{W}$ và ngược lại
- Sau huấn luyện, dự đoán rating:

$$p_{ui} = \mathbf{x}_i^T \mathbf{w_u}$$

# Ưu điểm

- Mô hình tổng quát và hiệu quả cao, khai thác được các mối quan hệ ẩn.
- Giảm yêu cầu lưu trữ: chỉ cần lưu $K(M+N)$ phần tử thay vì toàn bộ ma trận $M \times N$
- Phân rã ma trận giúp mô hình hóa sở thích tiềm ẩn của người dùng và đặc trưng ngầm của sản phẩm, từ đó tạo ra gợi ý mang tính cá nhân hóa sâu sắc và hiệu quả tính toán cao.