TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
NGÀNH CÔNG NGHỆ THÔNG TIN

BÁO CÁO TIẾN ĐỘ -TUẦN 03
MÔN: ĐỒ ÁN 3
ĐỀ TÀI: TÌM HIỂU VỀ THUẬT TOÁN RECOMMENDATION

GVHD : Thầy Huỳnh Xuân Phụng
SVTH :
Nguyễn Thành Như **17110202**
Võ Ngọc Thuận **17110234**

TP. Hồ Chí Minh, tháng 10 năm 2020

1. Collaborative filtering - CF (lọc cộng tác)

Có nhiều kỹ thuật lọc cộng tác và được chia thành hai dạng chính:

- Memory–based: lọc cộng tác dựa trên việc ghi nhớ toàn bộ dữ liệu.
- Model–based: Lọc cộng tác dựa trên các mô hình phân lớp, dự đoán.

1.1. Ma trận đánh giá (rating matrix, user-item rating matrix) [Đã hiểu]

- m người dùng U = { u_1, u_2, ..., u_m}
- n sản phẩm I = {i_1, i_2, ..., i_n}
- Ma trận đánh giá R = $\{r_{u,i}\}_{mxn}$ với $r_{u,i}$ thuộc R

	I_1	I_2	I_3	I_4
U_1	4	?	5	5
U_2	4	2	1	
U_3	3		2	4
U_4	4	4		
U_5	2	1	3	5

	Shrek	Snow White	Spider-man	Super-man
Alice	Like	Like		Dislike
Bob		Like	Dislike	Like
Chris		Dislike	Like	
Tony	Like		Dislike	?

Ma trận tường minh (explicit rating matrix). Người dùng đánh giá trực tiếp đối với các sản phẩm, dịch vụ, nội dung. Thang điểm thường là:

- Nhị phân: Like, Dislike. (0,1)
- Liên tục trong đoạn [0,1]
- Năm mức rời rạc: 1, 2, 3, 4, 5 (với 5 là mức đánh giá tốt nhất)

Ma trận đánh giá suy diễn (implicit rating matrix). Ma trận được suy diễn từ thông tin thu thập được về hành vi người dùng như:

- Tìm kiếm (browsing)
- Đọc (reading)
- Xem (watching)
- Chia sẻ (sharing)
- Mua (buying)

Sau đó ánh xạ hành vi người dùng vào các mức điểm.

1.2. Các tính chất của lọc cộng tác

1.2.1. Dữ liệu thưa (data sparsity) [Đã hiểu - ở tuần báo cáo trước]

- Ma trận đánh giá có thể rất thưa.
- Dữ liệu thưa ảnh hưởng rất nhiều đến hiệu quả hệ tư vấn bởi rất khó tính toán sự tương tự giữa các người dùng (users) hoặc giữa các sản phẩm (items)
  - Hai sản phẩm có thể rất giống nhau nhưng có ít người cùng đánh giá đồng thời hai sản phẩm.
  - Hai người dùng có thể giống nhau về sở thích nhưng chưa đánh giá cùng sản phẩm.
- Giải pháp: Áp dụng các kỹ thuật giảm số chiều (dimensionality reduction)

1.2.2. Xuất phát nguội (cold start) [Đã hiểu]

- Vấn đề người dùng mới (new user problem)
  - Chưa đánh giá sản phẩm nào
  - Chưa có các dữ liệu về các hành vi
- Vấn đề sản phẩm mới (new item problem)
  - Chưa được người dùng nào đánh giá
  - Chưa được ai xem, mua, tìm kiếm, ...
- Giải pháp:
  - Tư vấn các sản phẩm phổ biến, ngẫu nhiên cho người dùng mới; các sản phẩm mới được xuất hiện ở đầu trang
  - Content-boosted CF: tích hợp thêm hồ sơ (profile) người dùng mới hoặc sử dụng thêm các đặc tính của sản phẩm.

1.2.3. Khả năng mở rộng (scalability) [Chưa hiểu giải pháp khắc phục]

- Khi ma trận đánh giá lớn, tức số người dùng lẫn sản phẩm lớn thì thời gian tính toán sẽ tăng cao, khó đáp ứng tư vấn thời gian thực hoặc gần thời gian thực.
- Giải pháp:
  - Áp dụng các kỹ thuật giảm số chiều như SVD, PCA.
  - Item-based CF có khả năng mở rộng cao hơn so với user-based CF.

1.2.4. Vấn đề từ đồng nghĩa (synonymy) [Chưa hiểu giải pháp khắc phục]

- Các từ đồng nghĩa có thể gây cản trở cho việc tính toán độ tương tự.
- Ví dụ: children movie và children film có thể gây ra keyword-mismath, làm ảnh hưởng đến việc tính toán độ tương tự.
- Giải pháp: Áp dụng các kỹ thuật phân tích ngữ nghĩa như LSI (Latent Semantic Indexing), mô hình chủ đề (Topic Models) hoặc Deep Learning để giải quyết vấn đề này.

1.2.5. Gray sheep và Black sheep

- Gray sheep:
  - Những người có sở thích không giống ai.
  - CF không có hiệu quả trong trường hợp này.

- Có thể kết hợp CF và content-based.

- Black sheep:
  - Những người có đánh giá kì quặc (ví dụ như thích nhưng lại dùng những từ ngữ đánh giá như không thích) nên không thể recommend chính xác cho những người này.

1.2.6. Shilling attacks

- Xảy ra khi cạnh tranh không lành mạnh:
  - Đánh giá sản phẩm của mình cao, đánh giá sản phẩm của đối thủ thấp.
- Item-based CF ít bị ảnh hưởng bởi shilling attacks hơn so với user-based CF.
- Có thể phát hiện hiện tượng này ở bước tiền xử lý bằng phân tích phát hiện ngoại lệ.

1.3. Memory-based CF (lọc cộng tác dựa trên ghi nhớ)

- Sử dụng ma trận đánh giá để thực hiện dự đoán và tư vấn.
- Giả sử mỗi người dùng thuộc ít nhất một nhóm những người có chung sở thích, mối quan tâm.
- Người cần được tư vấn được gọi là active user.
- Những người dùng có sở thích tương tự với active user được gọi là neighbors.
- Cách tiếp cận:
  - User-based: dựa trên người dùng để dự đoán.
  - Items-based: dựa trên sản phẩm để dự đoán.

1.3.1. Bước 1: Tính toán mức độ tương tự (similarity computation) *[Hiểu được quy trình nhưng chưa hiểu cách tính toán, sẽ tìm hiểu kỹ hơn ở tuần tới]*

➢ Đối với item-based CF: Tính toán độ tương tự $w_{i,j}$ giữa hai item i và j dựa trên những người dùng cùng đánh giá hai item này.

➢ Đối với user-based CF: tính toán độ tương tự $w_{u,v}$ giữa hai người dùng u, v dựa trên những đánh giá của hai người dùng này trên cùng các items.

a) Khoảng cách Consine

	I_1	I_2	I_3	I_4
U_1	4	?	5	5
U_2	4	2	1	
U_3	3		2	4
U_4	4	4		
U_5	2	1	3	5

$$w_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}|| \times ||\vec{j}||}$$

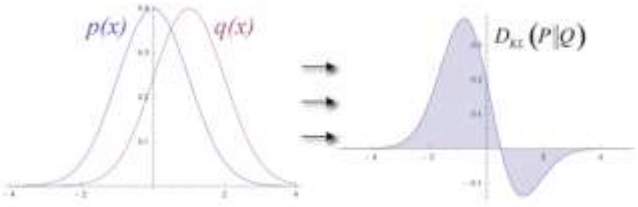
- i và j là hai vecto trong ma trận rating của hai sản phẩm i và j.

- Ở ma trận trên: i1 = (4,4,3,4,2) và i2 = (?,2,?,4,1)
- Khi tính toán wi,j: i1 = (4,4,2) và i2 = (2,4,1)

b) Khoảng cách Kullback-Leiber

Ký hiệu p(x) và q(x) là hai phân bố xác suất:

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$



$$\text{KL–similarity} = \frac{D(p||q) + D(q||p)}{2}$$

- Với user-based: hai phân bố là hai hàng trong ma trận đánh giá R
- Với item-based: hai phân bố là hai cột trong ma trận đánh giá R
- Các phân bố cần được chuẩn hóa trước khi tính D(p||q) và D(q||p).

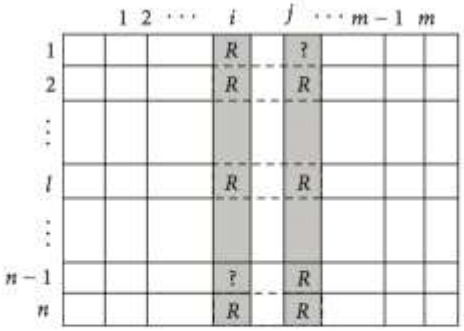
c) Tương quan Pearson - user-based CF

	I_1	I_2	I_3	I_4
U_1	4	?	5	5
U_2	4	2	1	
U_3	3		2	4
U_4	4	4		
U_5	2	1	3	5

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}}$$

- I là tập các items cả hai người dùng u và v cùng đánh giá.
- ru và rv là rating trung bình của u và v trên các sản phẩm trong I.

d) Tương quan Pearson - item-based CF



$$w_{i,j} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_j)^2}}$$

- U là tập các users cùng đánh giá hai sản phẩm i và j.
- ru,i: rating u đối với i, tương tự cho ru,j.
- ri, rj: trung bình rating của các người dùng trong U đối với i và j.

1.3.2. Bước 2: Dự đoán và tư vấn - Weight Sum of Other's Rating *[Hiểu được quy trình nhưng chưa hiểu cách tính toán, sẽ tìm hiểu kỹ hơn ở tuần tới]*

Dự đoán mức độ rating của **active user a đối với một sản phẩm i** nào đó, ký hiệu là $P_{a,i}$:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u \in U} |w_{a,u}|}$$

- ra và ru là rating trung bình của a và u trên các sản phẩm.
- wa,u là mức độ tương tương tự giữa hai người dùng a và u.
- U là tập tất cả người dùng (trừ a) đã đánh giá sản phẩm i.

	I_1	I_2	I_3	I_4
U_1	4	?	5	5
U_2	4	2	1	
U_3	3		2	4
U_4	4	4		
U_5	2	1	3	5

$$
\begin{aligned}
P_{1,2} &= \bar{r}_1 + \frac{\sum_u (r_{u,2} - \bar{r}_u) \times w_{1,u}}{\sum_u |w_{1,u}|} \\
&= \bar{r}_1 + \frac{(r_{2,2} - \bar{r}_2)w_{1,2} + (r_{4,2} - \bar{r}_4)w_{1,4} + (r_{5,2} - \bar{r}_5)w_{1,5}}{|w_{1,2}| + |w_{1,4}| + |w_{1,5}|} \\
&= 4.67 + \frac{(2 - 2.5)(-1) + (4 - 4)0 + (1 - 3.33)0.756}{1 + 0 + 0.756} \\
&= 3.95
\end{aligned}
$$

a) Dự đoán và tư vấn - Simple Weighted Average

Với tư vấn dựa trện sản phẩm (item-based), giá trị dự đoán rating của một người dùng u trên sản phẩm i, ký hiệu là $P_{u,i}$, được tính như sau:

$$P_{u,i} = \frac{\sum_{j \in J} r_{u,j} w_{i,j}}{\sum_{j \in J} |w_{i,j}|}$$

Trong đó:

- J là tập tất cả các sản phẩm (trừ i) mà người dùng u đã đánh giá.
- wi,j là mức độ tương tự giữa hai sản phẩm i và j.
- ru,j là rating của người dùng đối với sản phẩm j.

1.3.3. Top-N recommendations

a) Gợi ý top-N sản phẩm theo người dùng (user-based) [Đã hiểu]

- Gọi a là active user
- Tìm Ua là tập k người dùng tương tự nhất với a.
- Dùng độ đo tương quan Pearson hoặc Cosine
- Gọi C là tập tất cả các sản phẩm mà các người dùng trong Ua đã mua hoặc đánh giá mà a chưa mua hay đánh giá.
- Xếp hạng các sản phẩm trong C giảm dần theo số người dùng (trong Ua) mua hoặc đánh giá.
- Lấy top-N sản phẩm từ C theo thứ tự xếp hạng trên để tư vấn hay gợi ý cho a.

b) Gợi ý top N sản phẩm theo sản phẩm (items-based) [Đã hiểu]

- Gọi a là active user, R là ma trận đánh giá.

- Gọi Ia là tập sản phẩm mà a đã mua hoặc đánh giá.
- Với mỗi sản phẩm i trong Ia, xác định k sản phẩm tương tự nhất với i.
- C là tập tất cả các sản phẩm tương tự các sản phẩm trong Ia.
- Loại bỏ các sản phẩm Ia trong C.
- Tính độ tương tự giữa các sản phẩm trong C vơi tập sản phẩm Ia.
- Xếp hạng C giảm dần theo mức độ tương tự nói trên.
- Lấy top N sản phẩm từ C theo thứ tự giảm dần của độ tương tự, sau đó tư vấn cho người dùng a.

1.4. Lọc cộng tác dựa trên mô hình (model-based CF) [Chưa tìm được nhiều tài liệu]

- Thực hiện tư vấn dựa trên các mô hình học máy.
- Các mô hình được xây dựng dựa trên dữ liệu huấn luyện.
- Các phương pháp để xây dựng mô hình lọc cộng tác thường dùng:
  - Bayesian models
  - Clustering model

2. Content-based filtering – Phương pháp gợi ý theo nội dung

2.1. Ý tưởng

Từ thông tin mô tả của item, biểu diễn item dưới dạng vec-tơ thuộc tính. Sau đó dùng các vec-tơ này để học mô hình của mỗi user, là ma trận trọng số của user với mỗi item.

Thuật toán content-based gồm 2 bước:

- Bước 1: Biểu diễn items dưới dạng vec-tơ thuộc tính - item profile
- Bước 2: Học mô hình của mỗi user

2.2. Xây dựng Items Profile

Trong các hệ thống content-based, chúng ta cần xây dựng một bộ hồ sơ (profile) cho mỗi item. Profile này được biểu diễn dưới dạng toán học là một "feature vector" n chiều.

Một số phương pháp thường được sử dụng để xây dựng feature vector là:

- Sử dụng TF-IDF
- Sử dụng biểu diễn nhị phân

2.2.1. TF-IDF [Đã hiểu]

Là 1 kĩ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản

a) TF là gì?

Là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn.

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

Trong đó:

- tf(t, d): tần suất xuất hiện của từ t trong văn bản d
- f(t, d): Số lần xuất hiện của từ t trong văn bản d
- max({f(w, d) : w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d

b) IDF là gì?

Giúp đánh giá tầm quan trọng của một từ . Khi tính toán TF , tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng một số từ như "is", "of" và "that" thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế chúng ta cần giảm độ quan trọng của những từ này xuống.

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Trong đó:

- idf(t, D): giá trị idf của từ t trong tập văn bản
- |D|: Tổng số văn bản trong tập D
- |{d ∈ D : t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.

c) Công thức tính TF-IDF

Chúng ta có **công thức tính tf-idf** hoàn chỉnh như sau:

TF-IDF (t, d, D) = TF(t, d) x IDF(t, D)

Khi đó:

Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

2.2.2. Sử dụng biểu thức nhị phân [Chưa hiểu rõ cách sử dụng]

Movie	Adventure	Action	Science-Fiction	Drama	Crime	Thiller		User 1	User 2
Star Wars IV	1	1	1	0	0	0		1	-1
Saving Private Ryan	0	0	0	1	0	0			
American Beauty	0	0	0	1	0	0			
City of Gold	0	0	0	1	1	0		-1	1
Interstellar	0	0	1	1	0	0		1	
The Matrix	1	1	1	0	0	1			1

Trên đây là danh sách 6 bộ phim. Mỗi giá trị 0/1 thể hiện bộ phim đó không/có thuộc thể loại ở cột tương ứng. Bên cạnh đó, một hồ sơ người dùng cũng được tạo ra, với 1 là quan tâm, -1 là không, và null là chưa đánh giá. Như trong ví dụ trên, User 1 có quan tâm bộ phim Star Wars IV, còn User 2 thì không.

2.3. Học mô hình biểu diễn của user [Đã hiểu cách áp dụng công thức để tìm hàm mất mát]

Trong đó, x(m) là vector đặc trưng của item m, λ là một tham số dương
Mục tiêu của chúng ta sẽ là học ra mô hình của user, tức là tìm ra w(n) và b(n).
Biểu thức hàm mất mát của mô hình cho user thứ n

$$\mathcal{L}_n = \frac{1}{2s_n}||\hat{\mathbf{X}}_n\mathbf{w}_n + b_n\mathbf{e}_n - \hat{\mathbf{y}}_n||_2^2 + \frac{\lambda}{2s_n}||\mathbf{w}_n||_2^2$$

Đây chính là bài toán Ridge Regression, đã có sẵn trong thư viện sklearn.linear_model. Ridge của klearn. Chúng ta sẽ sử dụng thư viện này để tìm w(n) và b(n) cho mỗi user.
Ví dụ:
Xét bài toán: Ta có 5 items, vector đặc trưng của mỗi item được biểu diễn bởi một hàng:

$$\mathbf{X} = \begin{bmatrix} 0.99 & 0.02 \\ 0.91 & 0.11 \\ 0.95 & 0.05 \\ 0.01 & 0.99 \\ 0.03 & 0.98 \end{bmatrix}$$

Đồng thời, chúng ta có thông tin về user 5, đã đánh giá các item 1 và 4:
$\mathbf{y}_5 = [1, ?, ?, 4, ?]^T \Rightarrow \mathbf{r}_5 = [1, 0, 0, 1, 0]^T$
Đầu tiên, tiền xử lý để thu được sub vector:

$$\hat{\mathbf{X}}_5 = \begin{bmatrix} 0.99 & 0.02 \\ 0.01 & 0.99 \end{bmatrix}, \hat{\mathbf{y}}_5 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \mathbf{e}_5 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Sau đó áp dụng công thức **(*)**, ta sẽ được hàm mất mát:

$$\mathcal{L}_5 = \frac{1}{4}|| \begin{bmatrix} 0.99 & 0.02 \\ 0.01 & 0.99 \end{bmatrix} \mathbf{w}_5 + b_5 \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 4 \end{bmatrix} ||_2^2 + \frac{\lambda}{4}||\mathbf{w}_5||_2^2$$

Cuối cùng, chúng ta có thể sử dụng Stochastic Gradient Descent (SGD), hoặc Mini-batch GD để tìm ra w(5) và b(5).

3. *Công việc dự kiến tuần sau:*

- Tìm hiểu chi tiết về các thuật toán trong bước 1 của lọc cộng tác dựa trên ghi nhớ.
- Tìm hiểu cách sử dụng thư viện sklearn.linear_model.Ridge của klearn để tìm w(n) và b(n).

- Tìm hiểu cách trình bày ví dụ cụ thể về cách xây dựng một mô hình gợi ý sử dụng content-based.