



RECOMMENDATION SYSTEMS WITH MACHINE LEARNING

Hieu Le Xuan, 2023

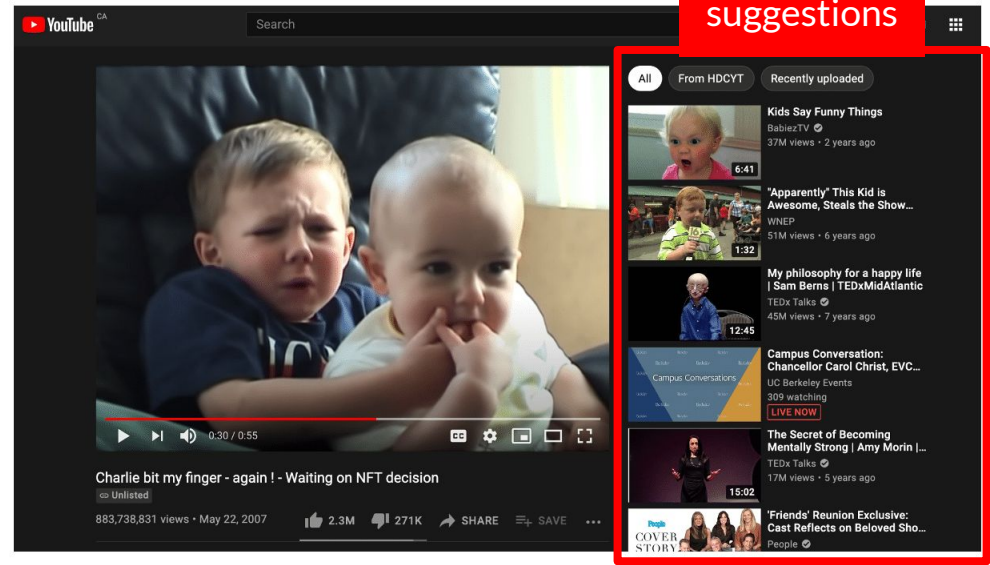


CONTENT

1. Introduction
2. Content-based approaches
3. Collaborative Approaches
4. Extensions: Using Other Approaches
5. Insights and Notes

INTRODUCTION

- Trong cuộc sống hằng ngày, chúng ta có thể đã thấy **Recommendation systems** (Hệ thống gợi ý) trong nhiều ứng dụng.
- Ví dụ: Youtube tự động đề xuất video tiếp theo



INTRODUCTION

Hoặc Amazon có danh sách “Sản phẩm được khuyến nghị” cho người dùng

amazon.com

Recommended for You

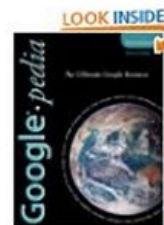
Amazon.com has new recommendations for you based on [items](#) you purchased or told us you own.



[Google Apps
Deciphered: Compute in
the Cloud to Streamline
Your Desktop](#)



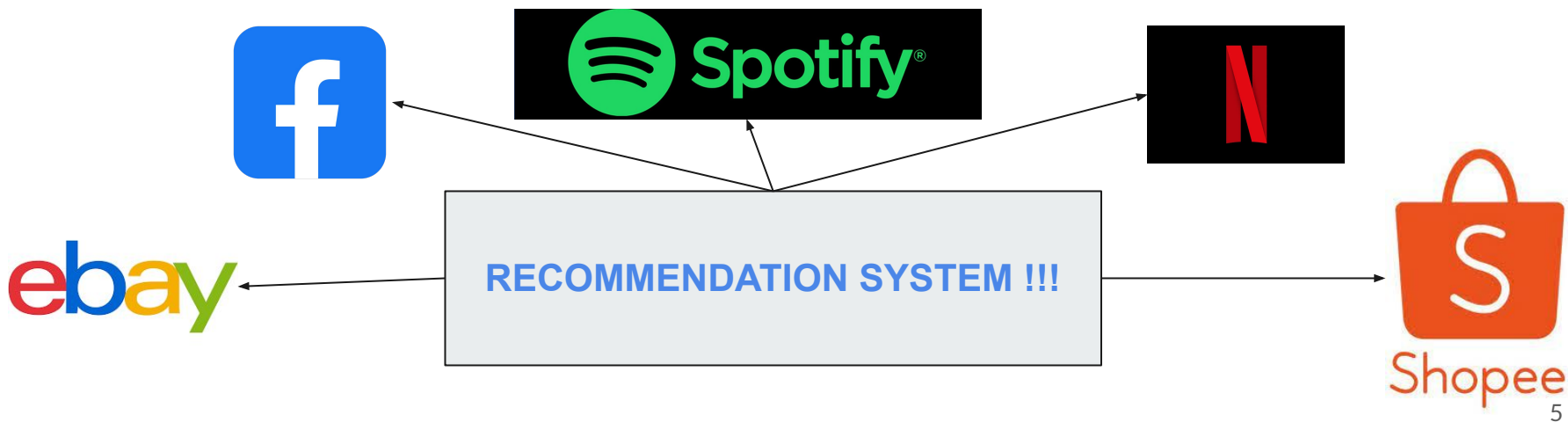
[Google Apps
Administrator Guide: A
Private-Label Web
Workspace](#)



[Googlepedia: The
Ultimate Google
Resource \(3rd Edition\)](#)

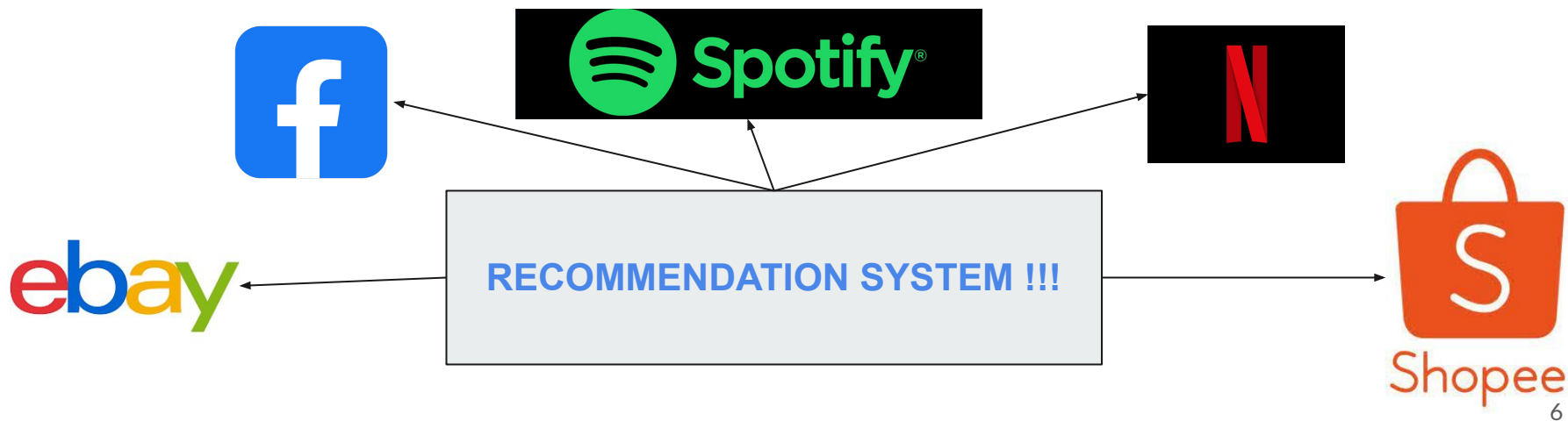
INTRODUCTION

Ngoài ra, còn rất nhiều hệ thống, ứng dụng, trang web khác sử dụng các hệ thống gợi ý có chức năng tự động gợi ý cho người dùng các sản phẩm phù hợp mà họ có thể thích/ muốn mua/ muốn xem...



INTRODUCTION

Một góc nhìn khác: hệ thống gợi ý có thể được xem như 1 mô hình hoặc 1 phương pháp dự đoán rating (đánh giá) của người dùng với sản phẩm



INTRODUCTION

Why using it (Research from UFL)

- Amazon's recommendation system drives **35% of its sales**.
- **Over 80%** of the TV shows on Netflix are discovered through its recommendation system.

AI Multiple

Increased sales/conversion

Increased user satisfaction

BENEFITS OF RECOMMENDATION SYSTEMS

Increased loyalty & share of
mind

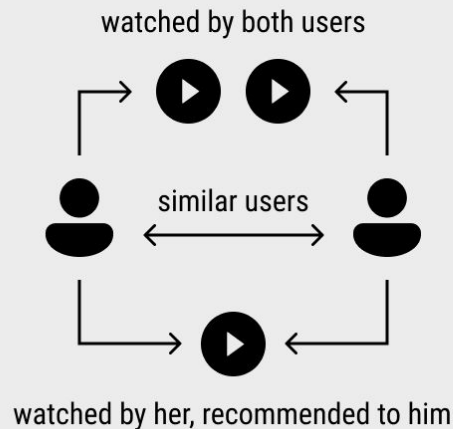
Reduced churn

INTRODUCTION

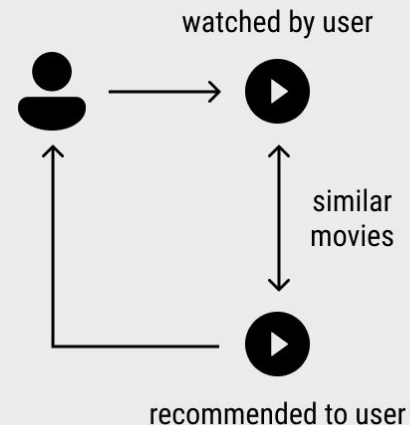
Các hệ thống gợi ý
được chia thành 2
nhóm chính:

Collaborative
Filtering vs
Content-based
Filtering

Collaborative Filtering



Content-Based Filtering





2. CONTENT-BASED APPROACH

The Utility Matrix

Gọi: X = Tập hợp các users, S = Tập hợp các items (movies, products).

Hàm ánh xạ **Utility function** $u: X * S \rightarrow R$

Trong đó R là tập hợp các giá trị đánh giá có thứ tự (ordered), ví dụ: 1 sao - 5 sao.

2. CONTENT-BASED APPROACH

The Utility Matrix

Hàm utility u dùng để trình bày dữ liệu rating ở dạng ma trận/ bảng gọi là utility matrix

Với mỗi cặp user-item thì giá trị tương ứng trong ma trận thể hiện mức độ yêu thích của user đối với item đó. Null values = not rated (chưa được đánh giá)

Trong thực tế, ma trận này thường là ma trận thưa

	 Harry Potter	 The Triplets of Belleville	 Shrek	 The Dark Knight Rises	 Memento
	✓		✓	✓	
		✓			✓
	✓	✓	✓		
				✓	✓



2. CONTENT-BASED APPROACH

The Item Profiles

Trong các hệ thống content-based đòi hỏi chúng ta phải xây dựng 1 hồ sơ (**profile**) cho mỗi item item. Profile là tập hợp các đặc điểm, đặc trưng của một sản phẩm. Ví dụ:

Movie: author, genre, actor, year...

Document: text, title

Profile của 1 item được xem là 1 vector



2. CONTENT-BASED APPROACH

Generating the Item Profiles by Text mining

What if items are text **documents**: articles, blog stories, ...?

Ideas: Biểu diễn mỗi item dưới dạng 1 vector thông qua các bước

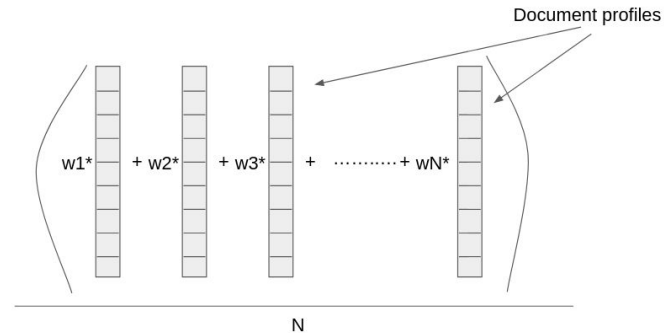
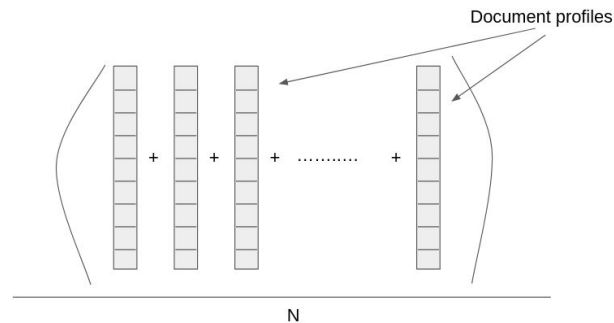
1. Loại bỏ các stop words
2. Tính TF-IDF cho các từ còn lại. Các từ có TFIDF cao có khả năng là từ khóa mấu chốt đặc tả 1 document (i.e: **subjects, main ideas**)
3. Chọn 1 số n, vd: số từ đặc tả 1 vector, để từ đó xây dựng các item profile vectors.

2. CONTENT-BASED APPROACH

Xây dựng User Profiles từ Item profiles:

Tổng hợp (lấy trung bình cộng,...) từ các **rated items**

Basic ideas: Larger weights
to Highly rated items



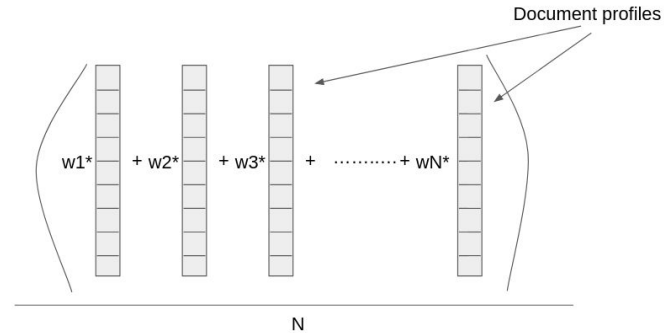
2. CONTENT-BASED APPROACH

Xây dựng User Profiles từ Item profiles:

Tổng hợp (lấy trung bình cộng,...) từ các **rated items**

Basic ideas: Larger weights
to Highly rated items

Normalization: Subtracting
average ratings and use
the subtracted ratings as
weights





2. CONTENT-BASED APPROACH

Xây dựng hàm dự đoán mức độ yêu thích của 1 user đối với 1 item

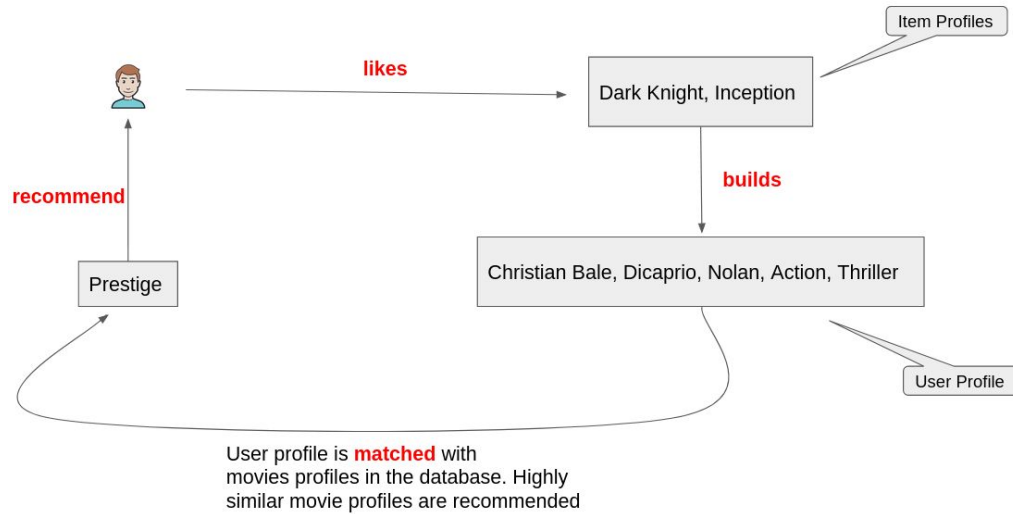
Tính cosine similarity của 2 vector user và item profiles

$$u(x, i) = \cos(x, i) = \frac{x \cdot i}{||x|| \cdot ||i||}$$

Liệu chúng ta có thể thay cosine similarity các phép tính khác ví dụ như:
Euclidean distance?

2. CONTENT-BASED APPROACH

Tóm tắt:





2. CONTENT-BASED APPROACH

An Alternative Method Utilizing Ridge Regression

Gọi $r_{ij} = 1$ khi item i được đánh giá bởi user j , y_{mn} là rating thật (ground truth) của user n với item m .

Như vậy, chúng ta có thể xây dựng 1 hàm lỗi để đo lường nhằm tối thiểu hóa sai số trong dự đoán rating:

$$\mathcal{L}_n = \frac{1}{2s_n} \sum_{m: r_{mn}=1} (\mathbf{x}_m \mathbf{w}_n + b_n - y_{mn})^2 + \frac{\lambda}{2s_n} \|\mathbf{w}_n\|_2^2$$

$$s_n = \sum_{m=1}^M r_{mn},$$

2. CONTENT-BASED APPROACH

Hàm lỗi này đã biến đổi vấn đề của chúng ta trở nên đơn giản hơn, từ 1 vấn đề phức tạp (điền rating vào 1 ma trận thưa) trở thành 1 vấn đề có thể giải quyết bằng các mô hình Regression

$$\mathcal{L}_n = \frac{1}{2s_n} \sum_{m: r_{mn}=1} (\mathbf{x}_m \mathbf{w}_n + b_n - y_{mn})^2 + \frac{\lambda}{2s_n} \|\mathbf{w}_n\|_2^2$$
$$s_n = \sum_{m=1}^M r_{mn},$$

⇒ Bài toán regression lúc này có thể giải quyết bằng nhiều phương pháp như sử dụng đại số tuyến tính hoặc gradient descent !



3. COLLABORATIVE FILTERING

- Tuy nhiên, chuyện gì sẽ xảy ra nếu chúng ta không thể xây dựng item profiles hoặc chúng ta không hề có sẵn item profiles?

Trong trường hợp này, chúng ta có thể dùng Collaborative Filtering

Có 2 loại Collaborative Filtering:

User-user collaborative Filtering vs Item - item collaborative Filtering

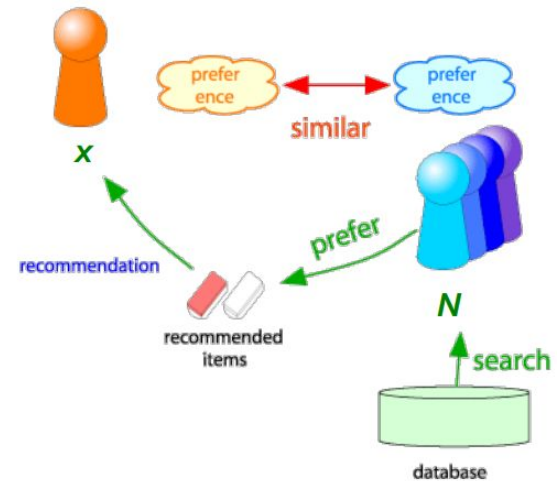
3. COLLABORATIVE FILTERING

User - User pairs

Ý tưởng: Xem user profiles như là 1 dòng trong utility matrix.

2 users được cho là tương tự (similar) nếu vector của 2 user đó có [distance nhỏ / cosine similarity lớn](#)

Trong phương pháp này, chúng ta sẽ tiến hành tìm N users tương tự nhất với user hiện tại để thực hiện recommendation.



3. COLLABORATIVE FILTERING

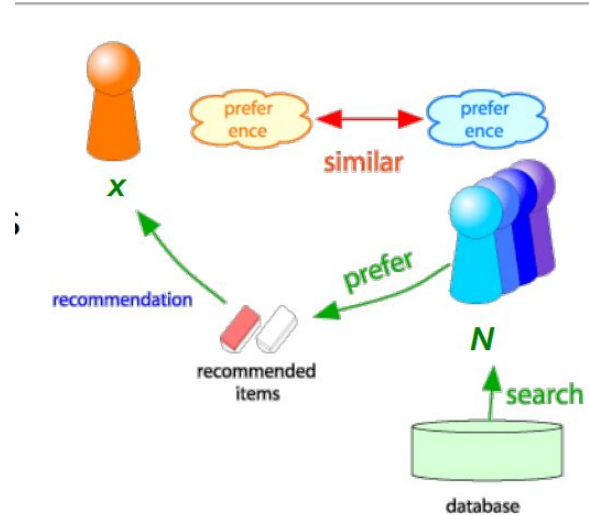
User - User pairs

“Trong phương pháp này, chúng ta sẽ tiến hành tìm N users tương tự nhất với user hiện tại để thực hiện recommendation.” - Got N users now, so What?

=> Bước tiếp theo: Dự đoán Rating của User X trên Item I qua 2 cách:

Normalized Average Rating

Weighted Average Rating



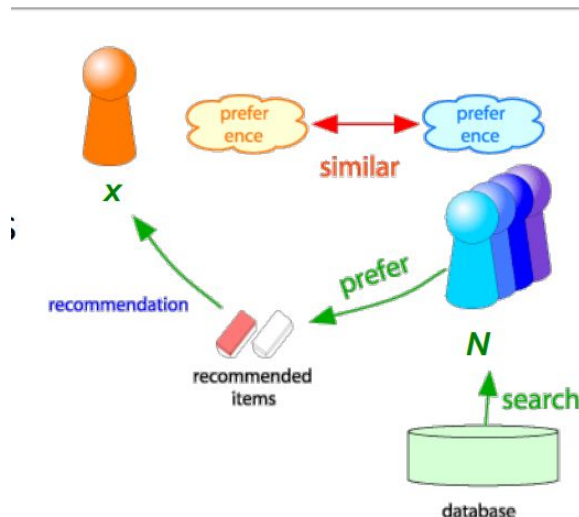
3. COLLABORATIVE FILTERING

User - User pairs

“Trong phương pháp này, chúng ta sẽ tiến hành tìm N users tương tự nhất với user hiện tại để thực hiện recommendation.” - Got N users now, so What?

Normalized Average Rating

- Sau khi tìm được N user tương tự, chúng ta tiến hành chuẩn hóa (normalize) các vector user profiles (ví dụ: có thể trừ giá trị average rating cho mỗi vector)
- Tìm mức chênh lệch (difference) giữa giá trị rating trước khi chuẩn hóa và sau khi chuẩn hóa cho mỗi user tại item I. Sau đó lấy trung bình các mức chênh lệch này (average the difference)
- Cộng giá trị tìm được này vào average rating của user U



3. COLLABORATIVE FILTERING

User - User pairs

“Trong phương pháp này, chúng ta sẽ tiến hành tìm N users tương tự nhất với user hiện tại để thực hiện recommendation.” - Got N users now, so What?

Weighted Average Rating

Note: We should normalize item rating first

Weighted Average Rating (taken from CS246 Stanford)

- Let \mathbf{r}_x be the vector of user x 's ratings
- Let N be the set of k users most similar to x who have rated item i
- **Prediction for item i of user x :**

- $r_{xi} = \frac{1}{k} \sum_{y \in N} r_{yi}$

- Or even better: $r_{xi} = \frac{\sum_{y \in N} s_{xy} \cdot r_{yi}}{\sum_{y \in N} s_{xy}}$

Shorthand:
 $s_{xy} = \text{sim}(x, y)$



3. COLLABORATIVE FILTERING

Item - Item pairs

Trong thực tế, số lượng users >> số lượng items

Ngoài ra, Đa số user khi được khảo sát đều không muốn đánh giá (rate) quá nhiều hoặc hoàn toàn không muốn đánh giá (Src: Data insights from Netflix)

>> Khi có 1 item mới xuất hiện, item này có thể có mức ảnh hưởng cao đến average rating của 1 user

Điều này dẫn tới sự phát triển của 1 phương pháp mới: **the item-item pairs cho collaborative filtering**

3. COLLABORATIVE FILTERING

The item-item pairs for collaborative filtering?

- Tương tự như phương pháp sử dụng cặp user-user, chúng ta **chỉ dự đoán** rating của Item i dựa trên những item mà user U đã đánh giá
- Lưu ý: Chúng ta cũng nên chuẩn hóa các vector item profiles

Weighted Average Rating (taken from CS246 Stanford)

- For item i , find other similar items
- Estimate rating for item i based on ratings for similar items
- Can use same similarity metrics and prediction functions as in user-user model

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

s_{ij} ... similarity of items i and j

r_{xj} ... rating of user x on item j

$N(i;x)$... set of items which were rated by x and similar to i



3. COLLABORATIVE FILTERING



Trong thực tế, việc sử dụng phương pháp **item-item** thường đem lại hiệu quả (efficiency) hơn việc sử dụng phương pháp **user-user**.

Giải thích:

- Trong thực tế, khối lượng công việc và sparsity của phương pháp user-user cao hơn: do đa phần thì số users \gg số lượng items
- “Thị hiếu” (taste) của 1 user có thể thay đổi theo thời gian hoặc, 1 user có thể có nhiều thể loại sản phẩm ưa thích khác nhau.



Pros and Cons

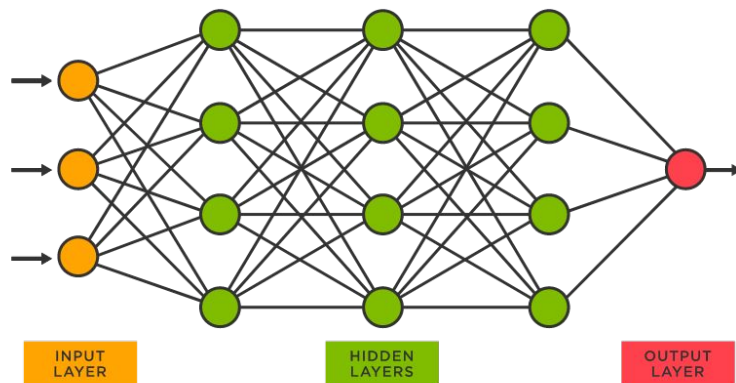


Content-based Filtering	Collaborative Filtering
<ul style="list-style-type: none">• No need data for users → No sparsity• Can recommend users with special taste• Because we have item profiles → No item cold-start• Explainable, e.g. <i>We recommend user X with movie Y because he seems to like genre Z</i>	<ul style="list-style-type: none">• No features are needed for items
<ul style="list-style-type: none">• Collecting features for Item is hard• New user: how do we build user profile• Never recommends items outside user's content profile, e.g. <i>User X may like all movies about horror but he may also like romantic movies</i>	<ul style="list-style-type: none">• User cold start• Cannot recommend items that are not yet rated• Have to deal with sparsity• May rely on other users e.g. <i>can't recommend to users with unique taste</i>

4. Extensions: Using Other Approaches

Ở 1 góc nhìn khác, liệu chúng ta có thể xây dựng các hệ thống Content-based filtering và Collaborative Filtering dựa trên

Neural Networks hay Gradient Descent?



4. Extensions: Using Other Approaches

Gradient Descent-based method for Collaborative Filtering

Trước hết, hãy diễn giải lại vấn đề theo 1 góc nhìn mới:

Cho biết 1 utility matrix trong đó các cột tượng trưng cho users và các hàng tượng trưng cho movies

w_j là vector profile của user j (mà chúng ta đã biết) và x_i là vector profile của movie i (chưa biết).

Example Utility Matrix

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

4. Extensions: Using Other Approaches

Gradient Descent-based method for Collaborative Filtering

Chúng ta có thể biểu diễn dự đoán về rating của User J trên movie I bằng 1 mô hình tuyến tính:

$$\mathbf{w}^{(J)} \cdot \mathbf{x}^{(I)} + \mathbf{b}^{(J)}$$

Rất rõ ràng, nếu chúng ta có đủ thông tin về nhiều users đã rate movie I chúng ta sẽ tìm được giá trị của vector \mathbf{x}_I , but how?

Example Utility Matrix

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	\mathbf{x}_1 (romance)	\mathbf{x}_2 (action)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

4. Extensions: Using Other Approaches

Gradient Descent-based method for Collaborative Filtering

Chúng ta có thể biểu diễn dự đoán về rating của User J trên movie I bằng 1 mô hình tuyến tính:

$$\mathbf{w}^{(j)} \cdot \mathbf{x}^{(i)} + b^{(j)}$$

Rất rõ ràng, nếu chúng ta có đủ thông tin về nhiều users đã rate movie I chúng ta sẽ tìm được giá trị của vector \mathbf{X}_i , but how?

PP: Tìm vector \mathbf{X}_i minimize sai số giữa ratings mà mô hình dự đoán trên toàn bộ users đã đánh giá item I này và giá trị dự đoán thực tế

$$\frac{1}{2} \sum_{j:r(i,j)=1} (\mathbf{w}^{(j)} \cdot \mathbf{x}^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{k=1}^n (x_k^{(i)})^2$$

4. Extensions: Using Other Approaches

Gradient Descent-based method for Collaborative Filtering

Chúng ta có thể biểu diễn dự đoán về rating của User J trên movie I bằng 1 mô hình tuyến tính:

$$\mathbf{w}^{(J)} \cdot \mathbf{x}^{(I)} + b^{(J)}$$

Rất rõ ràng, nếu chúng ta có đủ thông tin về nhiều users đã rate movie I chúng ta sẽ tìm được giá trị của vector \mathbf{x}_i , **but** how?

Với M items: Tìm M vector \mathbf{x} sao cho mỗi vector **minimize** sai số giữa **ratings** mà mô hình dự đoán trên toàn bộ users đã đánh giá item I này và giá trị dự đoán thực tế

Cost function to learn $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n_m)}$:

$$\min_{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} (\mathbf{w}^{(j)} \cdot \mathbf{x}^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2$$

4. Extensions: Using Other Approaches

Gradient Descent-based method for Collaborative Filtering

Chúng ta có thể biểu diễn dự đoán về rating của User J trên movie I bằng 1 mô hình tuyến tính:

$$\mathbf{w}^{(J)} \cdot \mathbf{x}^{(I)} + b^{(J)}$$

Tương tự, nếu chúng ta có đủ thông tin về nhiều items đã được rate chúng ta sẽ tìm được giá trị của vector \mathbf{W}_j , but how?

W sao cho mỗi vector
minimize sai số giữa ratings
mà mô hình dự đoán trên
tất cả item mà 1 user đã
đánh giá và giá trị dự đoán
thực tế

$$\min_{\mathbf{w}^{(1)}, b^{(1)}, \dots, \mathbf{w}^{(n_u)}, b^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} (\mathbf{w}^{(j)} \cdot \mathbf{x}^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (w_k^{(j)})^2$$

4. Extensions: Using Other Approaches

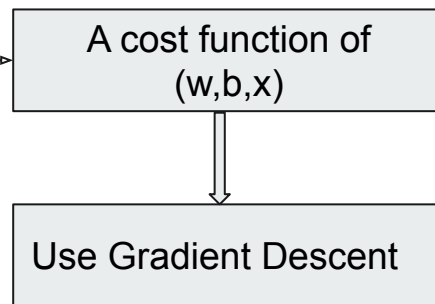
Gradient Descent-based method for Collaborative Filtering

Our model:

$$\mathbf{w}^{(j)} \cdot \mathbf{x}^{(i)} + b^{(j)}$$

Duality: Gộp 2 hàm lỗi bên trên lại ta có

$$\frac{1}{2} \sum_{(i,j):r(i,j)=1} (\mathbf{w}^{(j)} \cdot \mathbf{x}^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (w_k^{(j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2$$



4. Extensions: Using Other Approaches

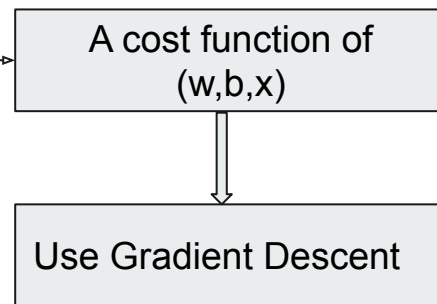
Gradient Descent-based method for Collaborative Filtering

Our model:

$$\mathbf{w}^{(j)} \cdot \mathbf{x}^{(i)} + b^{(j)}$$

Duality: Gộp 2 hàm lỗi bên trên lại ta có

$$\frac{1}{2} \sum_{(i,j): r(i,j)=1} (\mathbf{w}^{(j)} \cdot \mathbf{x}^{(i)} + b^{(j)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (w_k^{(j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^n (x_k^{(i)})^2$$



$$w_i^{(j)} = w_i^{(j)} - \alpha \frac{\partial}{\partial w_i^{(j)}} J(\mathbf{w}, \mathbf{b}, \mathbf{x})$$

$$b^{(j)} = b^{(j)} - \alpha \frac{\partial}{\partial b^{(j)}} J(\mathbf{w}, \mathbf{b}, \mathbf{x})$$

$$x_k^{(i)} = x_k^{(i)} - \alpha \frac{\partial}{\partial x_k^{(i)}} J(\mathbf{w}, \mathbf{b}, \mathbf{x})$$



4. Extensions: Using Other Approaches

Neural-based method for Content-based Recommendation

Nhắc lại: Trong các hệ thống content-based approach, chúng ta đã có hoặc xác định được các vector item profile và sẽ tiến hành nội suy các vector user profile

Và, các vector user profiles thường chứa đặc tính của users (age, gender, geographic...) trong khi item profiles chứa đặc tính của item



4. Extensions: Using Other Approaches

Neural-based method for Content-based Recommendation

Nhắc lại: Và, các vector user profiles thường chứa đặc tính của users (age, gender, geographic...) trong khi item profiles chứa đặc tính của item

“What if we could **transform user profiles** into **some degree of preference vector** V_u ?”

i.e. first element denotes “how much this user like horror movies”, second element denotes “how much this user like romantic movies”



4. Extensions: Using Other Approaches

Neural-based method for Content-based Recommendation

Nhắc lại: Và, các vector user profiles thường chứa đặc tính của users (age, gender, geographic...) trong khi item profiles chứa đặc tính của item

What if we could **transform item profiles** into **some descriptive characteristic vector V_m** ?

i.e. first element denotes “how much this item seems to be horror movies”, second element denotes “how much this item seems to be a romantic movies”



4. Extensions: Using Other Approaches

Neural-based method for Content-based Recommendation

Nhắc lại: Và, các vector user profiles thường chứa đặc tính của users (age, gender, geographic...) trong khi item profiles chứa đặc tính của item

V_u và V_m phải có cùng số lượng phần tử (same dimension) !

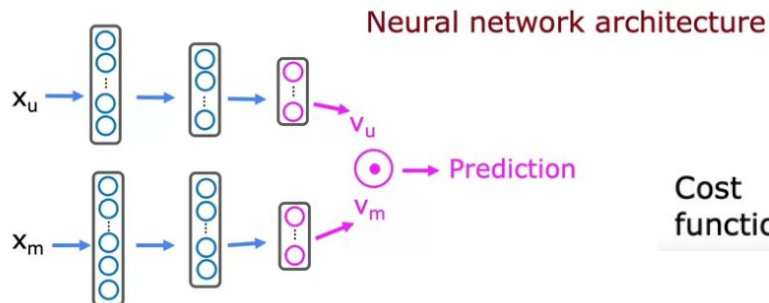
$V_u \cdot V_m \approx$ how much this user like this item = predicted rating of user u for the item m

4. Extensions: Using Other Approaches

Neural-based method for Content-based Recommendation

Nhắc lại: $V_u \cdot V_m \approx$ predicted rating of user u for the item m

Use Neural Network to solve this !



Cost function

$$J = \sum_{(i,j):r(i,j)=1} (v_u^{(j)} \cdot v_m^{(i)} - y^{(i,j)})^2 + \text{NN regularization term}$$



5. Insights and Notes

Retrieve and Ranking:

Giả sử rằng, chúng ta có 1 collection gồm rất nhiều items (vd: 1M+ movies), và

Chúng ta muốn giới thiệu cho user 1 vài items trong collection, trong khi chi phí và thời gian tính toán bị giới hạn

Làm thế nào để chúng ta làm điều này mà không cần phải dự đoán rating trên toàn bộ collection?



5. Insights and Notes

Retrieve and Ranking: **Làm thế nào để chúng ta làm điều này mà không cần phải dự đoán rating trên toàn bộ collection?**

Rõ ràng trong các trường hợp này, Brute Force không phải là giải pháp hiệu quả, thay vào đó chúng ta có thể sử dụng phương pháp gồm 2 bước: Retrieve (truy hồi) and Ranking (đánh giá)

Retrieve: Tạo 1 danh sách các items candidates

Vd: Top 10 movies cho mỗi thể loại, trong 3 thể loại được xem nhiều nhất HOẶC top 20 movies được xem nhiều ở VN.



5. Insights and Notes

Retrieve and Ranking: **Làm thế nào để chúng ta làm điều này mà không cần phải dự đoán rating trên toàn bộ collection?**

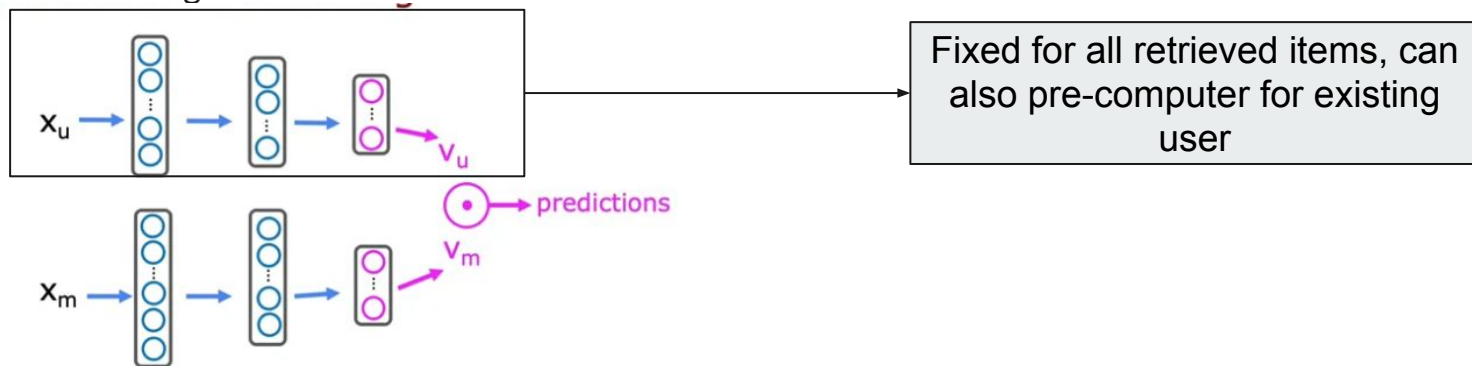
Retrieve: Tạo 1 danh sách các items candidates

Lưu ý: trong danh sách các item candidates được retrieved, chúng ta NÊN loại bỏ các item mà user đã xem qua

5. Insights and Notes

Retrieve and Ranking: **Làm thế nào để chúng ta làm điều này mà không cần phải dự đoán rating trên toàn bộ collection?**

Ranking: Lấy danh sách items candidate và dự đoán giá trị rating cho từng item. Khuyến nghị cho user các item có rating cao nhất.

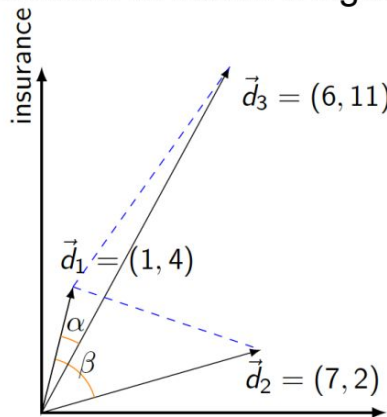


5. Insights and Notes

Cosine Similarity: Why not Euclidean distance?

- Euclidean distance is sensitive to the vector magnitude (related to how many terms are in a document)
- Cosine similarity is not sensitive to vector magnitude
- Euclidean distance l of *normalised* vectors vs. cosine similarity $\cos(\alpha)$:

$$l^2 = 2(1 - \cos(\alpha))$$





THANK YOU FOR YOUR ATTENTION

HIEU LE XUAN, 2023

What's more...

We can view the recommendation as a classification task => Use regular metric: Precision , Recall

$$P = \frac{\# \text{ of our recommendations that are relevant}}{\# \text{ of items we recommended}}$$

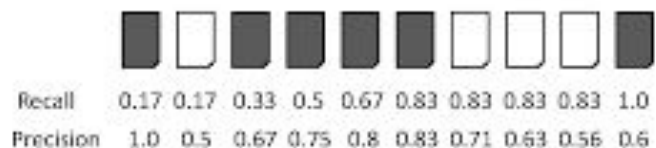
Precision

$$r = \frac{\# \text{ of our recommendations that are relevant}}{\# \text{ of all the possible relevant items}}$$

Recall



Ranking #1



Ranking #2



What's more...

Word Embedding

