

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



BÁO CÁO THỰC TẬP

Ngành: Khoa học máy tính

ĐỀ TÀI: Hệ thống gợi ý nội dung cho khách hàng
MyTV dựa trên lịch sử sử dụng

Cán bộ hướng dẫn: Đinh Thị Nhàn

Giảng viên đánh giá: Đặng Thanh Hải

Sinh Viên: Vũ Minh Hiếu

Mã Sinh Viên: 18020494

Lớp: K63-CACLC3

HÀ NỘI, 9/2021

Mục lục

1	Giới thiệu chung	iii
1.1	Giới thiệu công ty	iii
1.2	Giới thiệu công việc	iii
1.3	Giới thiệu bài toán	iii
2	Yêu cầu bài toán	iv
3	Lý thuyết hệ thống gợi ý	iv
3.1	Content Based Recommender System	iv
3.2	Collaborative Filtering	v
3.2.1	Neighborhood Collaborative Filtering	v
3.2.2	Matrix Factorization Collaborative Filtering	viii
4	Thực nghiệm	ix
4.1	Tập dữ liệu	ix
4.1.1	Tiền xử lý dữ liệu	ix
4.1.2	Định nghĩa 'Rating' của người dùng	xi
4.1.3	Encoding trường dữ liệu member-id và content-id	xii
4.2	Thực nghiệm các phương pháp	xii
4.2.1	Content based	xii
4.2.2	User-User Filtering	xiv
4.2.3	Item-Item Filtering	xiv
4.2.4	Matrix Factorization	xvi

Lời cảm ơn

Tôi xin chân thành cảm ơn công ty VNPT Media đã tạo điều kiện cho tôi có thể thực tập tại công ty, cán bộ hướng dẫn Đinh Thị Nhàn đã hướng dẫn và chỉ bảo tôi tận tình trong quá trình thực tập. Tôi cũng xin gửi lời cảm ơn tới giảng viên hướng dẫn Đặng Thanh Hải đã hướng dẫn và giúp đỡ tôi trong quá trình thực hiện môn học.

Vũ Minh Hiếu

1 Giới thiệu chung

1.1 Giới thiệu công ty

Tổng công ty Truyền thông (Tên viết tắt: VNPT-Media) được thành lập theo Quyết định số 89/QĐ-VNPT-HĐTV -TCCB ngày 08 tháng 05 năm 2015 của Chủ tịch Tập đoàn Bưu chính Viễn thông Việt Nam, trên cơ sở tổ chức lại Công ty VASC, Trung tâm Thông tin và Quan hệ công chúng và các bộ phận nghiên cứu, phát triển nội dung số, dịch vụ giá trị gia tăng của Công ty VDC, Công ty Vinaphone.

VNPT-Media hoạt động trong lĩnh vực nghiên cứu phát triển, kinh doanh dịch vụ Truyền hình, dịch vụ Truyền thông đa phương tiện, dịch vụ Giá trị gia tăng và Công nghệ thông tin với 4 công ty trực thuộc: Công ty Phát triển Dịch vụ Truyền hình, Công ty Phát triển Dịch vụ Giá trị gia tăng; Công ty Phát triển Phần mềm, Trung tâm Dịch vụ Tài chính số cùng các Ban chức năng, chi nhánh tại miền Trung và miền Nam.

1.2 Giới thiệu công việc

Vị trí thực tập tại cơ quan: Phòng Giải pháp phần mềm - Software Solution.

1.3 Giới thiệu bài toán

Hệ thống gợi ý nội dung cho khách hàng MyTV dựa trên lịch sử sử dụng

2 Yêu cầu bài toán

Hệ thống gợi ý (Recommender System) là hệ thống lọc thông tin tìm cách dự đoán "xếp hạng" hoặc "sở thích" của một người dùng đối với một sản phẩm.

MyTV là một ứng dụng truyền hình, trong đó có hình thức trả phí để sử dụng kho phim. Bài toán được giao: xây dựng hệ thống gợi ý dựa trên lịch sử xem của người dùng trên ứng dụng MyTV, để gợi ý các chương trình có thể phù hợp với từng người dùng.

Đầu vào (Input) của bài toán là lịch sử xem của người dùng.

Đầu ra (Output) của bài toán là "xếp hạng" độ yêu thích của người dùng đối với các chương trình họ chưa xem qua.

3 Lý thuyết hệ thống gợi ý

3.1 Content Based Recommender System

Từ thông tin mô tả của item biểu diễn item dưới dạng vector thuộc tính. Sau đó sử dụng các vector này áp dụng mô hình học máy để tìm ra được ma trận trọng số của user với mỗi item.

Thuật toán của Content Based cơ bản gồm hai bước:

Bước 1: Biểu diễn các item dưới dạng vector thuộc tính (item profile), mỗi vector này được biểu diễn dưới dạng toán học là một feature vector với n chiều (Hình 1). Trong những trường hợp đơn giản, feature vector được trực tiếp trích xuất từ item.

	A	B	C	D	E	...
Item A	1	1	0	0	1	...

Item Feature Vector

Hình 1: Ví dụ về Item Feature vector

Bước 2: Sử dụng mô hình học máy các user. Áp dụng học máy đi tìm mô hình cho mỗi user, bài toán này có thể được coi là Regression nếu đầu ra là một dải các giá trị (Ratings) hoặc có thể coi là Classification nếu kết quả đầu ra là một vài trường hợp cụ thể, ví dụ: like/dislike.

Ưu điểm:

- Có thể đưa ra dự đoán đối với item mới, khi chưa có dữ liệu rating về item đó.

Nhược điểm:

- Khi xây dựng mô hình cho mỗi user, Content Based không tận dụng được thông tin từ các User khác. Điều này khiến có những item sẽ không được xét đến.

- Không phải lúc nào các vector feature cho mỗi item cũng đầy đủ. Khiến việc xây dựng Item profile trở nên khó khăn.

- Dù Content Based có thể đưa ra dự đoán đối với item mới, tuy nhiên thuật toán này lại không hiệu quả với việc gợi ý cho người dùng mới. Vì Content Based xây dựng mô hình dự đoán cho người dùng dựa trên lịch sử các rating của người dùng đó, nên việc xây dựng mô hình dự đoán cho người dùng mới với ít hoặc không có dữ liệu rating là không hiệu quả.

3.2 Collaborative Filtering

3.2.1 Neighborhood Collaborative Filtering

Collaborative Filtering là một dạng kĩ thuật lọc các item mà người dùng (user) có thể sẽ thích dựa trên phản hồi của nhóm người dùng tương đồng hoặc một nhóm item tương đồng. Có nhiều cách để tìm người dùng tương đồng và tổng hợp sự lựa chọn của họ để đưa ra danh sách đề xuất.

Kĩ thuật này cơ bản được chia làm hai bước:

Bước 1: Tìm các user và item tương đồng.

Bước 2: Dự đoán rating của các item chưa được đánh giá bởi user.

User-user Collaborative Filtering

Xác định mức độ quan tâm của mỗi user tới một item dựa trên mức độ quan tâm của similar user (người dùng tương đồng) tới item đó.

Cách để đo similarity (độ tương đồng) giữa hai user là xây dựng feature vector cho mỗi user rồi áp dụng một mô hình có khả năng đo similarity giữa hai vector đó. Tuy nhiên việc xây dựng feature vector này khác với việc xây dựng item profiles như ở Content Based. Các feature vector này được xây dựng dựa trên rating của user với một items. Tuy nhiên, các rating của người dùng không đầy đủ điều này ảnh hưởng đến việc tính toán độ tương đồng giữa hai vector. Cách khắc phục được sử dụng là điền thêm một giá trị vào các phần còn trống sao cho việc điền không ảnh hưởng nhiều đến sự giống nhau giữa hai vector. Lưu ý, việc điền này chỉ phục vụ cho việc tính similarity và không phục vụ cho việc suy luận ra giá trị cuối cùng. Tuy nhiên trong thực tế sẽ xuất hiện các user dễ tính, rating thường cao, và user khó tính, rating thấp. Nếu lựa chọn một giá trị để điền vào phần trống, giả sử như 2.5 giá trị an toàn nằm giữa 0 và 5, vẫn gây ra ảnh hưởng đến độ similarity giữa các user.

Chuẩn hóa dữ liệu:

Lấy giá trị trung bình ratings của mỗi user. Giá trị trung bình cao tương ứng với user dễ tính và ngược lại. Sau đó trừ rating cho giá trị trung bình vừa tìm được và thay phần còn trống bằng '0'. Ý nghĩa của việc chuẩn hóa là:

- Trừ đi trung bình cộng của mỗi cột khiến giá trị sau khi thu được sẽ không bị ảnh hưởng bởi sự dễ tính hay khó tính giữa các user. Giá trị dương tương ứng với user thích item và ngược lại. Phần còn trống được thay bằng giá trị 0, vừa là số ở giữa khoảng giá trị rating và không bị ảnh hưởng bởi tính chất của user.

- Về mặt kĩ thuật, việc xây dựng utility matrix sẽ có số chiều rất lớn với số lượng lớn user và item, nên việc thay giá trị 0 vào các ô trống sẽ tốt hơn khi lưu ma trận này dưới dạng sparse matrix (ma trận thưa).

Cosine Similarity:

$$\text{cosine_similarity}(u_1, u_2) = \cos(u_1, u_2) = \frac{\vec{u}_1 \cdot \vec{u}_2}{\|u_1\| \cdot \|u_2\|} \quad (1)$$

Độ similarity của hai feature vector là một giá trị trong đoạn $[-1, 1]$. Giá trị 1 thể hiện hai vector hoàn toàn giống nhau và ngược lại. Tuy nhiên, phương pháp này có nguy cơ xảy ra hiện tượng đầy bộ nhớ nếu số lượng user là lớn.

Các ô trống (missed rating) được xác định dựa trên thông tin về k user có similarity cao nhất, chỉ quan tâm đến các user đã rated item đang xét. Công thức được sử dụng để dự đoán rating của user u với item i là:

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in N_{u,i}} \bar{y}_{i,u_j} \text{sim}(u, u_j)}{\sum_{u_j \in N_{u,i}} |\text{sim}_{u,u_j}|} \quad (2)$$

Trong đó $N_{(u,i)}$ là tập k users có similarity cao nhất của u mà đã rate i.

Sau khi tính được giá trị \hat{y} của một user, predicted rating sẽ được đưa về thang điểm trước khi chuẩn hóa bằng cách cộng với trung bình rating của user đó.

Ưu điểm:

- Chỉ cần thông tin về lịch sử đánh giá của người dùng.
- Thuật toán này có thể đề xuất những item mới lạ.

Nhược điểm:

- Trong thực tế, số lượng user rất lớn kéo theo đó là Similarity matrix là rất lớn, gây nguy cơ đầy bộ nhớ khi phải lưu ma trận này.
- Với số lượng user rất lớn so với số lượng item, ma trận thưa sau khi chuẩn hóa thường rất thưa, lý do đến từ việc user không thường xuyên rate item, do đó khi user rate thêm một item sẽ có sự thay đổi nhiều. Kéo theo đó là việc tính toán ma trận Similarity, tốn thời gian tính toán và lưu trữ do cần phải tính lại ma trận này.

Item-item Collaborative Filtering

Tính toán similarity giữa các items và recommend những item gần giống với item yêu thích của user.

Các bước thực hiện tương tự User-user Collaborative Filtering, nhưng thay user bằng item ở bước tính toán similarity matrix.

Ưu điểm:

- Vì số lượng item thường nhỏ hơn số lượng user, nên ma trận tương đồng (Similarity matrix) cũng sẽ nhỏ hơn rất nhiều, thuận lợi cho lưu trữ và tính toán.
- Các phần tử trong vector của item, bằng số user, nhiều hơn phần tử trong vector của user, nên khi có thêm các rating kết quả similarity không bị sai lệch nhiều. Kéo theo việc cập nhật ma trận tương đồng không diễn ra liên tục.

Ưu điểm:

- Không hiệu quả với các item hoàn toàn không có rating.

3.2.2 Matrix Factorization Collaborative Filtering

Ma trận rating R với kích thước $m \times n$ có thể *approximately factorized* thành 2 ma trận P và Q có kích thước $m \times k$ và $m \times k$ với $k \ll (m, n)$:

$$R \approx P^T Q \quad (1)$$

Mỗi dòng của P biểu diễn một *latent item factor*, mỗi cột của Q biểu diễn một *latent user factor*. Rating xấp xỉ r_{ij} của ma trận R được tính bởi công thức [1]:

$$r_{ij} \approx P_i^T Q_j \quad (2)$$

Để tiến hành học các latent vector p_i và q_i , phương pháp tiếp cận của *maximum likelihood approach* là tìm minimum của bình phương sai số với các rating đã biết [2]:

$$\text{Min}(P, Q) \sum_{i,j \in I} (r_{ij} - P_i^T Q_j)^2 \quad (3)$$

Với I là bộ các cặp i, j đã được rating trong ma trận R .

Tuy nhiên việc tìm minimum của bình phương sai số tại các điểm dữ liệu có thể gây ra hiện tượng overfitting. Regularization là một phương pháp toán học có thể giải quyết vấn đề này [3] bằng việc thêm $\frac{\lambda}{2}(\|P\|^2 + \|Q\|^2)$, với $\lambda > 0$ là tham số regularization, vào công thức (3). Từ đó ta được công thức:

$$\text{Min}(P, Q) \sum_{i,j \in I} (r_{ij} - P_i^T Q_j)^2 + \frac{\lambda}{2}(\|P\|^2 + \|Q\|^2) \quad (4)$$

Sau đó tôi sử dụng SGD (stochastic gradient descent) để tối ưu công thức trên.

Sử dụng quy tắc cập nhật của SGD, tôi có công thức cập nhật P_i và Q_j :

$$\begin{aligned} P_i &= P_i - \eta [Q_j (P_i^T \cdot Q_j - r_{ij}) + \lambda(P_i)] \\ Q_j &= Q_j - \eta [P_i (P_i^T \cdot Q_j - r_{ij}) + \lambda(Q_j)] \end{aligned}$$

4 Thực nghiệm

4.1 Tập dữ liệu

4.1.1 Tiền xử lý dữ liệu

Tôi được giao hai tập dữ liệu "Rs-nlp-corpus1.csv" và "Rs-cf-wacth-history.csv" để tiến hành thực nghiệm.

Tập dữ liệu "**rs-nlp-corpus1.csv**" mang thông tin về các nội dung (content) của MyTV, tập dữ liệu có 65508 bản ghi và mỗi bản ghi có 7 trường, bao gồm: content-id, content-name, content-desc, content-cate-id, content-director, content-actor và content-country.

	0	content_id	content_name	content_desc	content_cate_id	content_director	content_actor	content_country
count	65507	65507	65507	65507	65507	65507	65507	65507
unique	16582	20078	11657	13813	247	6717	11040	136
top	132171, Lang Điện Hạ, <p>Cuối thời nhà Đường, L...	131981	Mắt Tịch		7267			6
freq	16	23	35	409	6493	1885	1561	23635

Hình 2: Bộ dữ liệu về content

Trong đó các trường content-cate-id, content-actor và content-country có thể có đa giá trị (multiple value) trong một bản ghi, có nghĩa một nội dung có thể có nhiều giá trị content-cate-id, content-actor và content-country (Hình 3. 4. 5.). Dấu hiệu nhận biết vị trí nào có đa giá trị: nếu một vị trí có đa giá trị, vị trí đó sẽ được gói trong cặp kí tự '\x00'.

Tiến hành kiểm tra trùng lặp các nội dung, tôi nhận thấy chỉ có 15528 bản ghi không bị trùng lặp, nếu tất cả các trường của hai bản ghi đều giống nhau thì hai bản ghi đó được coi là trùng lặp, trong tổng số 65508 bản ghi mang thông tin về các nội dung. Tuy nhiên, trong 15528 bản ghi này vẫn tồn tại các bản ghi trùng nhau về giá trị của trường content-id (Hình 6), qua kiểm tra các bản ghi này là cùng là một content nhưng có sự khác nhau về giá trị trong các trường. Tôi quyết định cộng gộp các giá trị khác nhau ở ba trường là content-cate-id, content-director, content-actor, mục đích phục vụ cho bước xây dựng item-profile cũng như đảm bảo tính toàn vẹn của dữ liệu.

Tập dữ liệu '**rs-cf-watch-history.csv**' chứa thông tin về các lần xem nội dung

```

7267          6493
5989          5117
5965          3898
6911          3425
              3175
              ...
7433          1
6917,7265     1
5943,7463     1
6917,7255,7325 1
6917,6931,7325 1
Name: content_cate_id,

```

Hình 3: Trường dữ liệu Category

```

Animals          660
Đang cập nhật    444
Nhiều diễn viên  414
Peter Dinklage;Lena Headey 283
...
Hồng Vĩnh Thành;Thang Lạc Văn;Huỳnh Tâm Dĩnh 1
Trần Đạo Minh;Củng Lợi;Trương Tuệ Văn;Quách Đào;Lưu Bội Kỳ;Tổ Phong 1
MATTHEW MCCONAUGHEY;PENÉLOPE CRUZ;STEVE ZAHN;LAMBERT WILSON;RAINN WILSON;LENNIE JAMES;GLYNN TURMAN 1
Zhao Zuo;Yu Xin Yan;Mi Te 1
Bill Paxton;Jeffrey Dean Morgan;Olivier Martinez 1
Name: content_actor, Length: 10057, dtype: int64

```

Hình 4: Trường dữ liệu Actor

```

Love 4K Nature    660
Bùi Chí Trung     408
Neil Fearnley     343
Đang cập nhật     313
...
Sandy Johnson     1
TERRENCE MALICK   1
Rob Connolly      1
Liu Xin           1
Hào Tân Tường     1
Name: content_director,

```

Hình 5: Trường dữ liệu Director

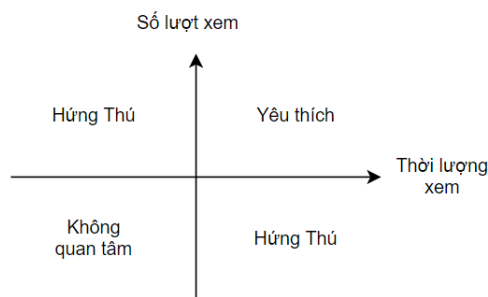
content_id	content_name	content_desc	content_cate_id	content_director	content_actor
5070	Thám tử lừng danh Conan: kẻ đánh bom cao ốc - ...	<p>Trong phần phim này;Shinichi nhận lời mời d...	7267	Kanetsugu Kodama	Minami Takayama
5070	Thám tử lừng danh Conan: kẻ đánh bom cao ốc - ...	<p>Trong phần phim này;Shinichi nhận lời mời d...		Kanetsugu Kodama	Minami Takayama
6293	Verimatrix Test 4 - mpg		7267		
6293	MillionaireDetective_01	<p>MillionaireDetective_01</p>	7317,7319		

Hình 6: Các bản ghi trùng giá trị trường content-id

của khách hàng trong 30 ngày từ 1/6/2021 đến 31/6/2021, bao gồm các trường: member-id, content-id, content-name, duration, date-time, filename-date. Sau khi load thành công tập dữ liệu, nhận thấy có 40 bản ghi không hợp lệ là con số rất nhỏ so với 5038269 nên tôi quyết định loại bỏ những bản ghi không hợp lệ. Tập dữ liệu nhận được bao gồm 5038269 bản ghi và 6 trường.

4.1.2 Định nghĩa 'Rating' của người dùng

Ban đầu, tôi tiến hành thực nghiệm mô hình với output dự đoán là thời lượng xem (duration) của người dùng (member) đối với một nội dung (content), tuy nhiên sai số của output là rất lớn do đây là trường dữ liệu có độ nhiễu cao. Vì thế, tôi đề xuất một cách chuẩn hóa dữ liệu dựa trên thời lượng xem và số lượt xem của người dùng đối với một nội dung.

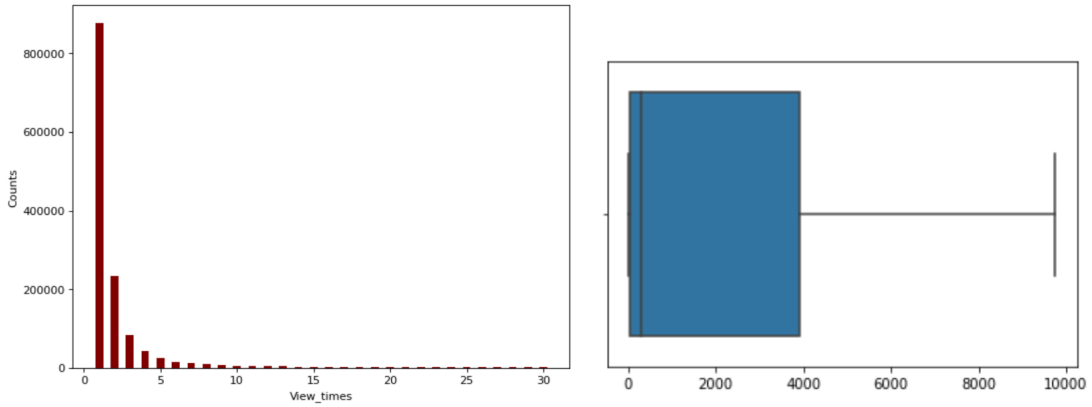


Hình 7: Phương pháp chuẩn hóa dữ liệu

Với phương pháp chuẩn hóa dữ liệu được đề xuất, độ yêu thích của một người dùng đối với một nội dung được quyết định dựa vào hai yếu tố là thời lượng xem và số

lượt xem. Phương pháp này chia độ yêu thích của người dùng thành 3 lớp: Yêu thích, hứng thú và không quan tâm

Sử dụng cách phân loại ở hình 7, tôi tiến hành phân tích hai trường dữ liệu số lượt xem và thời lượng xem để tìm mốc phân loại. Đối với thời lượng xem tôi sử dụng giá trị trung vị của trường dữ liệu thời lượng xem làm mốc phân loại. Với số lượt xem, do giá trị một chiếm 823612 trên 1396291 giá trị nên tôi quyết định chọn giá trị một làm mốc so sánh. Nếu một member xem một content nhiều hơn một lần thì số lần xem đó được tính là cao.



Hình 8: Biểu đồ thống kê lượt xem và sự phân bố thời lượng xem

4.1.3 Encoding trường dữ liệu member-id và content-id

Hai trường dữ liệu member-id và content-id được sử dụng để xây dựng utility matrix, tuy nhiên giá trị của hai trường dữ liệu này là không liên tục, điều này gây khó khăn khi xây dựng utility matrix nên tôi tiến hành mapping hai trường dữ liệu này sang hai trường dữ liệu mới có giá trị liên tục và đảm bảo mapping ở đây là mapping 1-1.

4.2 Thực nghiệm các phương pháp

4.2.1 Content based

Xây dựng Item Profile

	member_id	content_id	duration	view_time	enjoyment	encode_member_id
0	3747	128141	69	4	2	0
1	3747	131909	4177	2	3	0
2	3747	135739	51	1	1	0
3	3747	135743	3	1	1	0
4	3747	136128	41	1	1	0
...
1396286	14641535	136456	3	1	1	317361
1396287	14641673	130107	29	1	1	317362
1396288	14641673	132535	10	1	1	317362
1396289	14641673	132731	48	1	1	317362
1396290	14641703	129191	858	1	2	317363

1396291 rows × 6 columns

Hình 9: Tập dữ liệu sau tiền xử lý

Sau tiền xử lý, các trường có thể sử dụng cho thuật toán gợi ý dựa trên nội dung như content category, content actor, content director được lưu dưới dạng đa trị (multiple value) trong cùng một bản ghi. Việc đưa dạng dữ liệu này vào giải thuật là bất khả thi, nên tôi sử dụng onehot encoding cho cả ba trường. Tuy nhiên khi gộp kết quả, xảy ra hiện tượng tràn RAM, nên tôi quyết định bỏ trường content director, chỉ sử dụng hai trường content category và content actor. Kết quả thu được tập dữ liệu Item Profile gồm 12858 bản ghi và 17032 trường.

Thực hiện giải thuật gợi ý dựa trên nội dung

Tôi sử dụng mô hình Ridge Regression để thực hiện thuật toán gợi ý dựa trên nội dung với tập dữ liệu.

Input: User-id, tập dữ liệu lịch sử xem.

Output: Danh sách đề xuất các phim phù hợp với người dùng.

Top 10 bộ phim Member 3575461 có thể thích xem là: [14769. 124240. 7444. 125277. 125275. 127843. 128539. 18087. 13036. 127409.]

Hình 10: Output của mô hình

Để đánh giá mô hình, tôi lấy 100000 bản ghi cuối cùng của tập dữ liệu lịch sử xem và tiến hành thực nghiệm mô hình với tập dữ liệu con này. Kết quả được ghi lại ở bảng dưới.

Mô hình	RMSE (Sau tối ưu)
Thử nghiệm với item profile gồm gồm content-genre và content actor	0.681
Thử nghiệm với item profile gồm content-genre và content actor (không có hằng số bias)	0.684
Thử nghiệm với item profile gồm content-genre	0.684
Thử nghiệm với item profile gồm content-genre (không có hằng số bias)	0.6935

4.2.2 User-User Filtering

Việc áp dụng giải thuật User-User Filtering lên toàn tập dữ liệu gây áp lực tới bộ nhớ và mất nhiều thời gian nên tôi quyết định sử dụng một sample gồm 3000 khách hàng để áp dụng và đánh giá kết quả của giải thuật.

Kết quả sau khi tiến hành áp dụng mô hình và tính toán sai số thu được $RMSE = 0.9568$.

Sau khi đánh giá mô hình, tôi tiến hành xây dựng mô hình recommend dựa theo giải thuật User-user Filtering. Input: member-id. Output: Bảng được sắp xếp theo thứ tự giảm dần về kết quả dự đoán. (Hình 11)

4.2.3 Item-Item Filtering

Không giống User-User Filtering với số lượng user lớn gây áp lực khi tạo ma trận tương đồng (similarity matrix), Item-Item Filtering với số item là 3478 khi tạo ma trận tương đồng có kích thước tốt hơn so với User-User Filtering. Vì thế tôi tiến

User_id: 4078

	content_id	prediction
0	10361	3.0
1	122750	3.0
2	124157	3.0
3	132137	3.0
4	17589	3.0
...
1848	123130	0.0
1849	132795	0.0
1850	123144	0.0
1851	128711	0.0
1852	130353	0.0

1853 rows x 2 columns


Hình 11: Kết quả đề xuất của User-User Filtering

hành áp dụng mô hình với toàn bộ tập dữ liệu.

Kết quả sau khi tiến hành áp dụng mô hình và tính toán sai số thu được $RMSE = 0.9311$.

Sau khi đánh giá mô hình, tôi tiến hành xây dựng mô hình recommend dựa theo giải thuật User-user Filtering. Input: member-id. Output: Bảng được sắp xếp theo thứ tự giảm dần về kết quả dự đoán. (Hình 12)

User_id: 3747

100%  3471/3471

	content_id	prediction
0	130397	3.0
1	128429	3.0
2	128575	3.0
3	132307	3.0
4	136326	3.0
...
3466	128625	0.0
3467	130199	0.0
3468	130419	0.0
3469	135823	0.0
3470	6123	0.0

Hình 12: Kết quả đề xuất của Item-item Filtering

4.2.4 Matrix Factorization

Tiến hành áp dụng giải thuật Matrix Factorization với thuật toán tối ưu stochastic gradient descent tôi nhận được sai số RMSE = 0.187 đối với tập test.

Từ mô hình đã được huấn luyện, tôi xây dựng hàm đề xuất đối với từng user. Input: User ID. Output: đề xuất cho user dưới dạng bảng.

	rating	item_ids
0	2.860562	8633
1	2.491415	9819
2	2.430919	6242
3	2.365755	123022
4	2.363671	9458
...

Hình 13: Kết quả đề xuất của mô hình

Dưới đây là bảng tổng kết kết quả các mô hình.

Mô hình	RMSE
Content based (Trung bình)	0.685
User-user Collaborative Filtering	0.9568
Item-item Collaborative Filtering	0.9311
Matrix Factorization (SGD)	0.187

Tài liệu

- [1] Accelerated Singular Value Decomposition (ASVD) using momentum based Gradient Descent Optimization. Kumar Raghuwanshi, Rajesh Kumar Pateriya.
- [2] Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer (Long. Beach. Calif).
- [3] Aggarwal, C.C., 2016. Model-based collaborative filtering. In: Recommender Systems, Springer International Publishing, Cham, pp. 71–138.



Ý kiến đánh giá:

.....

.....

.....

.....

.....

.....

.....

.....

Hà Nội, ngày tháng năm 20

Người hướng dẫn



Ý kiến đánh giá:

.....

.....

.....

.....

.....

.....

.....

.....

Điểm số:..... Điểm chữ:.....

Hà Nội, ngày tháng năm 20

Giảng viên đánh giá