Content-Based-Recommendation-System

Linkgit:

https://github.com/Thaibao247/Algorithms-for-recommendation-system?fbclid=IwAR39-K20qoUQHbH4wokuWWtCwJCvAySFaHBpBjrKPGIgeZQGT8BEbn4l28I

Nhóm:

Đinh Quốc Hùng 19133025

Trần Nguyên Thái Bảo 19133010

1. Mục tiêu

- Dựa vào những sản phẩm mà người dùng chưa mua, chưa đọc, chưa xem .. tương tự với những sản phẩm mà người dùng đã từng mua, từng đọc, từng xem trong quá khứ để đề xuất cho người dùng.

2. Utility matrix

2.1. Tổng quan về Utility matrix

- Chúng ta có 2 thực thể chính cần quan tâm trong Recommendation System là user và item. Mối quan hệ này sẽ là sự quan tâm của user đến từng item khác nhau, có thể nói là đánh giá của người dùng cho sản phẩm đó. Tất cả các đánh giá của người dùng cho sản phẩm đó và các đánh giá cho các sản phẩm chưa biết mà chúng ta cần phải dự đoán, tạo nên một ma trận là Utility matrix tương tự như ví dụ dưới đây.

	A	B	C	D	E
Sản phẩm 1	**5**	**?**	**1**	**5**	**2**
Sản phẩm 2	**2**	**4**	**?**	**?**	**?**
Sản phẩm 3	**1**	**4**	**2**	**?**	**?**
Sản phẩm 4	**?**	**?**	**0**	**1**	**5**

- Trong ví dụ này thì chúng ta thấy 0 đến 5 là mức đánh giá của người dùng đối với từng sản phẩm. Những dấu "?" là thông tin chúng ta chưa có hay không có trong dữ liệu mà chúng ta sẽ phải dự đoán. Nhiệm vụ của hệ thống gợi ý là sẽ dự đoán những điểm đó.
- Trong thực tế thì có rất nhiều user và item trong một hệ thống, mà đánh giá của user cho item chỉ có 1 phần nhỏ, có khi còn có user không đánh giá bất kì một item nào(đối với user này ta sẽ gợi ý dựa vào các item phổ biến nhất). Vì vậy các ô dấu '?' trong Utility Matrix là rất nhiều và những ô được điền thì rất ít.

- Muốn được điền nhiều ô trong Utility Matrix thì chúng ta cần đòi hỏi sự quan tâm của user đó đến từng item và đánh giá của user đó cho item đó, dựa vào đó chúng ta có thế đưa những item chất lượng tốt cho người dùng và giúp hệ thống hiểu về người dùng và có thể gợi ý sản phẩm phù hợp nhất với user.

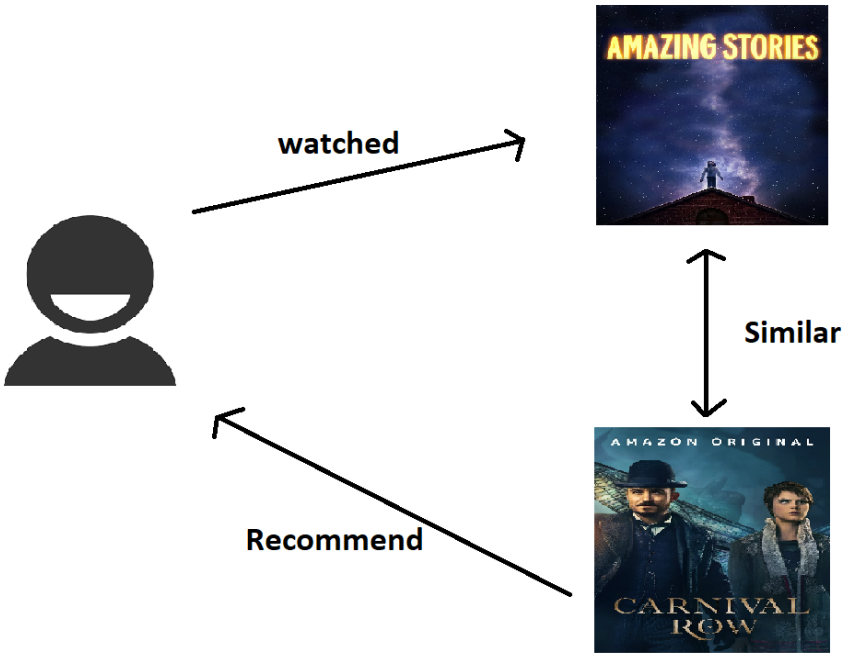
2.2. Xây dựng Utility matrix

- Không có Utility matrix thì chúng ta không thể gợi ý sản phẩm, nên việc xây dựng Utility matrix là khá quan trọng. Thường có 2 cách tiếp cận để xây dựng Utility matrix:
    + Nhờ vào rate của sản phẩm. Có hạn chế là rất ít user đánh giá sản phẩm, và nếu có thì đó là những đánh giá không mấy khách quan.
    + Nhờ vào hành vi mua hàng của người dùng, Ví dụ như người dùng like hay từng mua, xem qua một sản phẩm nào đó thì hệ thống sẽ gợi ý cho họ những sản phẩm tương tự như vậy. Trường hợp này ma trận được xây dựng với 2 con số là 0 và 1, 1 thể hiện người dùng thích sản phẩm còn 0 thì thể hiện chưa thông tin.

3. Tìm hiểu Content-Based Recommendations

3.1. Tổng quan

- Dựa vào những sản phẩm mà người dùng chưa mua, chưa đọc, chưa xem ,... tương tự với những sản phẩm mà người dùng đã từng mua, từng đọc, từng xem trong quá khứ để đề xuất những cái tương tự cho người dùng.



3.2. Hoạt động

- Dựa vào các sản phẩm đã được đánh giá bởi người dùng. Thông tin của người dùng đã dùng để đánh giá nguồn thông tin chưa biết, để xây dựng truy vấn cho search engine.
- Hồ sơ khách hàng được xây dựng bằng cách phân tích phản hồi đến các câu hỏi của họ, lịch sử đường dẫn. Nó gợi ý sản phẩm với các đặc điểm tương tự với những sản phẩm mà người dùng đã chọn trong quá khứ. Cụ thể hơn là nó dựa vào nội dung của thuộc tính. Hàm tương tự để xác định ra mối quan hệ của các sản phẩm, theo giả thiết các sản phẩm mà tương tự với nhau về nội dung sẽ được đánh giá tương tự nhau.
- Đối với phương pháp này, hệ thống gợi ý trích ra 1 bộ các thuộc tính của sản phẩm được đánh giá bởi người dùng, sau đó hệ thống phân tích điểm tương tự giữa các sản phẩm đã xử lý và tất cả các sản phẩm còn lại.
- Mục tiêu của hệ thống này tập trung vào việc tìm sự tương quan giữa nội dung của các sản phẩm như là đối ngược nhau để tìm ra sự tương quan giữa những người dùng. Gốc rễ của phương pháp content-based có thể được bắt nguồn từ việc truy xuất thông tin.
- Phương pháp này áp dụng cho hầu hết các tài liệu văn bản khó để cho máy tính diễn giải. Nó không được xem như là thái độ của khách hàng với sản phẩm mà là giới hạn độ chính xác của hệ thống. Cũng vì vậy chúng ta không thể lọc sản phẩm dựa vào một vài đánh giá về chất lượng, phong cách, điểm hoặc là xem.

3.3. Ưu nhược điểm

- Lợi ích chính của **Content-based filtering** là hiệu suất của chúng không dựa trên sự tồn tại của dữ liệu ưu tiên, và do đó, chúng phù hợp trong trường hợp của vấn đề cold-start.
- Nhược điểm:
    + Người dùng chỉ có thể nhận được các khuyến nghị tương tự như trải nghiệm trước đó của họ
    + Một số mục, chẳng hạn như đa phương tiện, âm nhạc và hình ảnh, rất khó phân tích
    + Vấn đề đối với người dùng mới

3.4. Tổng kết

- Content-based filtering kiểm tra mô tả mặt hàng để nhận ra mặt hàng được người dùng quan tâm đặc biệt. Các hệ thống này sử dụng các đặc điểm của mặt hàng và xếp hạng mà người dùng đã đưa ra để đưa ra các đề xuất. Nó đề xuất các sản phẩm tương tự như những sản phẩm mà người dùng đã yêu thích trước đây. Họ cũng mô tả các mục mục tiêu theo các thuộc tính của chúng như màu sắc, hình dạng và chất liệu và áp dụng phân tích nội dung cho chúng. Tuy nhiên, một hệ thống dựa trên nội dung thuần túy cũng có những khuyết điểm như trên.

4. Demo Content-Based Recommendations sử dụng thuật toán TF-IDF

4.1. Thuật toán TF-IDF

- **TF-IDF** là một kĩ thuật khá nổi tiếng trong các bài toán xử lý ngôn ngữ tự nhiên và khai phá dữ liệu dạng văn bản, nhằm mục đích tính độ quan trọng của word trong văn bản, văn bản đó nằm trong nhiều tệp khác nhau.
    + IF(Term frequence): tần suất xuất hiện của w trong văn bản d. Một từ xuất hiện càng nhiều thì độ quan trọng của nó càng cao. Công thức của IF:

$$Tf(w,d) = \frac{c(w,d)}{len(d)} .$$

    + IDF(Inverse document frequence): nghịch đảo của số văn bản chừa từ w / tổng số văn bản. Một từ xuất hiện trong nhiều văn bản thì độ quan trọng của nó càng giảm. Công thức tính IDF:

$$\text{IDF(w, D)} = \log \frac{M}{f(w,D)}$$

Vậy TF-IDF của từ w trong văn bản d trên tập D là:

$$\text{TF-IDF(w, d, D)} = \frac{c(w,d)}{len(d)} * \frac{M}{f(w,D)}$$

4.2. Dataset Movielen 100k

- Bộ dữ liệu gồm 100.000 ratings của 1000 người dùng cho khoảng 1700 bộ phim
- Sử dụng các file chủ yếu để thực hiện demo bao gồm:
  - ***u.data***: chứa các đánh giá của người dùng

	user_id	movie_id	rating	unix_timestamp
0	1	1	5	874965758
1	1	2	3	876893171
2	1	3	4	878542960
3	1	4	3	876893119
4	1	5	3	889751712
5	1	6	5	887431973
6	1	7	4	875071561
7	1	8	1	875072484
8	1	9	5	878543541
9	1	10	3	875693118

  - ***ua.base, ua.test***: chia dữ liệu thành tập train và test

    - Tập train:

	user_id	movie_id	rating	unix_timestamp
0	1	1	5	874965758
1	1	2	3	876893171
2	1	3	4	878542960
3	1	4	3	876893119
4	1	5	3	889751712
5	1	6	5	887431973
6	1	7	4	875071561
7	1	8	1	875072484
8	1	9	5	878543541
9	1	10	3	875693118

    - Tập test:

	user_id	movie_id	rating	unix_timestamp
0	1	20	4	887431883
1	1	33	4	878542699
2	1	61	4	878542420
3	1	117	3	874965739
4	1	155	2	878542201
5	1	160	4	875072547
6	1	171	5	889751711
7	1	189	3	888732928
8	1	202	5	875072442
9	1	265	4	878542441

- ***u.user***: chứa thông tin người dùng

	user_id	age	sex	occupation	zip_code
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213
5	6	42	M	executive	98101
6	7	57	M	administrator	91344
7	8	36	M	administrator	05201
8	9	29	M	student	01002
9	10	53	M	lawyer	90703

- ***u.genre***: chứa tên loại phim

	genre_name	genre_id
0	unknown	0
1	Action	1
2	Adventure	2
3	Animation	3
4	Children's	4
5	Comedy	5
6	Crime	6
7	Documentary	7
8	Drama	8
9	Fantasy	9

- ***u.item***: chứa thông tin mỗi bộ phim

	movie_id	movie_name	date		link_imdb	g0	g1	g2	g3	g4	...	g9	g10	g11	g12	g13	g14	g15	g16	g17	g18
0	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	0	1	1	...	0	0	0	0	0	0	0	0	0	0
1	2	GoldenEye (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?GoldenEye%20(...	0	1	1	0	0	...	0	0	0	0	0	0	0	1	0	0
2	3	Four Rooms (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Four%20Rooms%...	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	0
3	4	Get Shorty (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Get%20Shorty%...	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0

4.3. Xây dựng demo

4.3.1. Biểu diễn các items dưới dạng vector thuộc tính (items profiles) bằng TF-IDF:

- Biểu diễn dựa trên thể loại phim (19 loại)
- Kết quả của 5 vectors thuộc tính của 5 bộ phim đầu sau khi sử dụng TF-IDF có trong thư viện sklearn:

```
feature vector for movie  0
[0.         0.         0.         0.74066017 0.57387209 0.34941857
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.        ]
feature vector for movie  1
[0.         0.53676706 0.65097024 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.53676706 0.
 0.        ]
feature vector for movie  2
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
feature vector for movie  3
[0.         0.71065158 0.         0.         0.         0.5397592
 0.         0.         0.45125862 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.        ]
feature vector for movie  4
[0.         0.         0.         0.         0.         0.
 0.735504   0.         0.36318585 0.         0.         0.
 0.         0.         0.         0.         0.57195272 0.
 0.        ]
```

4.3.2. Tạo hàm lấy ra đánh giá của mỗi user cho từng bộ phim

```python
# Create function get items that rated by per user
import numpy as np
def get_items_rated_by_user(rate_matrix, user_id):
    y = rate_matrix[:,0]
    ids = np.where(y == user_id +1)[0]
    item_ids = rate_matrix[ids, 1] - 1
    scores = rate_matrix[ids, 2]
    return (item_ids, scores)
```

✓ 0.7s

- Ví dụ cho user 1 đánh giá 10 bộ phim trên thang điểm 5 (sao)

	Movie_id	rating by user 1
0	0	4
1	9	2
2	13	4
3	18	3
4	24	4
5	99	5
6	110	4
7	126	5
8	236	4
9	241	5
10	254	4

4.3.3. Học mô hình cho từng user

- Vấn đề: tìm mô hình sao cho không phụ thuộc vào các user khác

4.3.4. Đánh giá mô hình

5. Tham khảo

https://www.analyticssteps.com/blogs/what-content-based-recommendation-system-machine-learning

https://viblo.asia/p/tim-hieu-ve-content-based-filtering-phuong-phap-goi-y-dua-theo-noi-dung-phan-1-V3m5WGBg5O7

https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/

https://www.youtube.com/watch?v=2uxXPzm-7FY

https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/

Kế hoạch tuần tới: Đánh giá thuật toán.