XÂY DỰNG HỆ THỐNG GỢI Ý BẰNG THUẬT TOÁN NGƯỜI LÁNG GIỀNG VÀ THỬ NGHIỆM TRÊN MOVIELENS DATASET

Hà Thị Thanh Ngà, Nguyễn Đình Cường
Khoa Công nghệ thông tin, Đại học Nha Trang
E-mail:ngahtt@ntu.edu.vn, cuongnd@ntu.edu.vn

***Tóm tắt*—Hệ thống gợi ý là một kỹ thuật lọc thông tin được dùng để dự đoán sở thích của người dùng. Việc sử dụng các hệ thống gợi ý giúp người dùng ra quyết định và lựa chọn được những mục tin phù hợp (mặt hàng, nhạc, phim, ảnh, tin tức, sách,….) từ một nguồn dữ liệu sẵn có. Hiện nay các hệ thống gợi ý được dùng nhiều trong các lĩnh vực thương mại điện tử, giải trí, giáo dục,... Bài báo cáo này nhằm giới thiệu về các hệ thống gợi ý và các kỹ thuật lọc cộng tác phổ biến được dùng trong hệ thống gợi ý; đồng thời minh họa một hệ thống gợi ý cộng tác với tập dữ liệu mẫu MovieLens**

Từ khóa: Lọc cộng tác, hệ thống gợi ý, đánh giá

I. GIỚI THIỆU

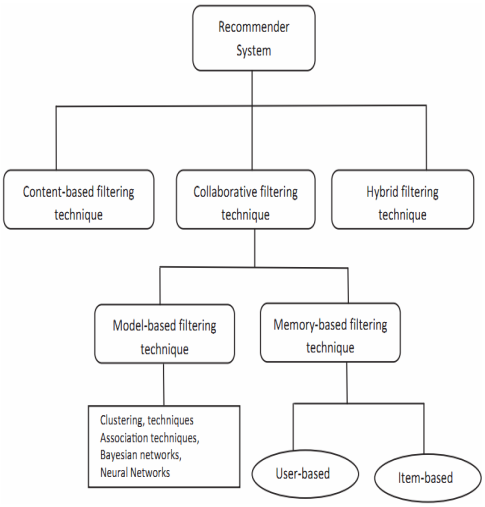
Các hệ thống gợi ý (Recommendation System - RS) là những công cụ phần mềm và kỹ thuật đưa ra đề nghị hoặc gợi ý mục tin hoặc hành động cho người dùng. Những gợi ý cá nhân hóa đưa ra danh sách các mục tin đã được xếp hạng theo sở thích và những ràng buộc của người dùng để cố gắng dự đoán việc quyết định những sản phẩm hoặc dịch vụ nào phù hợp nhất. Những quyết định liên quan những tiến trình ra quyết định khác nhau của từng người dùng cụ thể về việc mua những mặt hàng nào, nghe những bản nhạc nào, hay đọc những tin tức trực tuyến nào. Các hệ thống gợi ý xử lý vấn đề quá tải thông tin mà người dùng thường gặp phải bằng cách cung cấp cho họ các khuyến nghị về nội dung và dịch vụ được cá nhân hóa, độc quyền [1]. Mục đích chính của hệ thống gợi ý là tạo ra các đề nghị quan trọng và thông tin gợi ý, các sản phẩm hoặc các đối tượng cho xã hội người dùng mà người dùng có thể quan tâm đến. Ví dụ, gợi ý sách trên trang Amazon, Netflix đề xuất những bộ phim bằng cách sử dụng hệ thống gợi ý để xác định khuynh hướng của người dùng và sau đó, thu hút người dùng xem phim ngày càng nhiều [2].

Có nhiều phương pháp và giải thuật khác nhau có thể trợ giúp xây dựng các hệ thống gợi ý. Các cách tiếp cận có thể được phân loại cơ bản thành những hệ thống gợi ý dựa nội dung (content-based recommendations), gợi ý cộng tác (collaborative recommendations), và các cách tiếp cận lai (hybrid) kết hợp phương pháp cộng tác và dựa trên nội dung [1], [2], [3]. Các hệ thống gợi ý dựa trên nội dung sẽ gợi ý người dùng những mục tin (items) tương tự những mục tin người dùng đã từng thích trước đó. Trong những hệ thống gợi ý cộng tác người dùng sẽ được giới thiệu những mục tin mà nhiều người có cùng sở thích đã từng thích trước đó.

Những cách tiếp cận khác gồm nhóm kỹ thuật không cá nhân hóa là các hệ thống dựa trên đặc tính, dựa trên hành vi, dựa trên ngữ cảnh, dựa trên kiến thức, dựa trên luật và nhiều lớp gợi ý khác[3].

II. CÁC HỆ THỐNG GỢI Ý PHỔ BIẾN

Việc sử dụng các kỹ thuật gợi ý chính xác và hiệu quả là rất quan trọng đối với một hệ thống sẽ cung cấp khuyến nghị tốt và hữu ích cho những người dùng đơn lẻ của hệ thống. Những nhóm hệ thống gợi ý chính gồm các hệ thống gợi ý dựa nội dung (Content-based Recommendation Systems), các hệ thống lọc cộng tác (Collaborative Filtering Systems) và các hệ thống lai ghép (Hybrid Systems)



Hình 1. Hệ thống gợi ý [1]

2.1 Các hệ thống gợi ý dựa nội dung (Content-based Recommendation Systems)

Kỹ thuật dựa nội dung là một giải thuật nhấn mạnh vào việc phân tích các nội dung/thuộc tính (attributes) của các mục tin (items) để phát sinh các dự đoán. Cách tiếp cận này yêu cầu việc sắp xếp các mục tin vào từng nhóm hoặc đi tìm các đặc trưng của từng mục tin. Việc gợi ý các mục tin dựa vào hồ sơ (profiles) của người dùng bằng việc sử dụng các đặc tính được rút trích từ nội dung của các mục tin người dùng đã đánh giá trong quá khứ. Các mục tin được gợi ý đến người dùng liên quan phần lớn các mục tin đã được đánh giá tích cực bởi người dùng.

2.2 Các hệ thống lọc cộng tác (Collaborative Filtering Systems)

Lọc cộng tác là kỹ thuật dự đoán đối với nội dung không thể được mô tả dễ dàng và đầy đủ bởi siêu dữ liệu (metadata) như những bộ phim và nhạc. Kỹ thuật lọc cộng tác hoạt động bằng cách xây dựng một cơ sở dữ liệu (ma trận người dùng-mục tin) sở thích về các mục tin theo những người dùng. Sau đó kết hợp những người dùng với các sở thích và mối quan tâm thích hợp bằng cách tính toán các độ tương tự giữa những hồ sơ người dùng để tạo các gợi ý. Các kỹ thuật lọc cộng tác có thể được chia thành hai loại: dựa bộ nhớ (memory-based) và dựa mô hình (model-based):

- Kỹ thuật dựa bộ nhớ (Memory-based) (còn gọi là Phương pháp láng giềng - Neighborhood-based): có thể đạt được theo hai cách gồm các kỹ thuật dựa người dùng (user-based) và dựa mục tin (item-based), trong đó hoặc là dựa trên dữ liệu quá khứ của người dùng "tương tự - similarity" (user-based approach), hoặc là dựa trên dữ liệu quá khứ của những mục tin "tương tự" (item-based approach).
- Kỹ thuật dựa trên mô hình (Model-based): quy trình xây dựng mô hình có thể được thực hiện bằng cách dùng các kỹ thuật khai phá dữ liệu và học máy. Các kỹ thuật này liên quan đến việc

xây dựng các mô hình dự đoán dựa trên dữ liệu thu thập được trong quá khứ. Ví dụ những kỹ thuật này gồm luật kết hợp, phân cụm, mạng Bayesian, mạng nơron. Các kỹ thuật này phân tích ma trận người dùng – mục tin để nhận diện các mỗi quan hệ giữa các mục tin; những mối quan hệ này được dùng để so sánh danh sách những gợi ý top-N.

III. HỆ THỐNG GỢI Ý CỘNG TÁC LÁNG GIỀNG GẦN (NEIGHBORHOOD-BASED) VỚI BỘ DỮ LIỆU MOVIELENS

Các hệ thống gợi ý cộng tác (hay các hệ thống lọc cộng tác) cố gắng dự đoán hiệu dụng (utility) của các mục tin cho một người dùng cụ thể dựa vào những mục tin được đánh giá trước đó bởi những người dùng khác. Ý tưởng chính của các cách tiếp cận gợi ý cộng tác là sử dụng thông tin về hành vi trước đó của những người dùng đang có trong hệ thống để dự đoán mục tin nào người dùng hiện tại sẽ có thể thích nhất và vì vậy sẽ dùng đến. Các cách tiếp cận cộng tác lấy ra ma trận những đánh giá hoặc xem xét của người dùng-mục tin được đưa ra như một đầu vào và tạo ra một dự đoán là con số chỉ mức độ thích hoặc không thích một mục tin nào đó của người dùng hiện tại, hoặc một danh sách n mục tin gợi ý. Danh sách được tạo không chứa các mục tin người dùng hiện tại đã dùng.

Các hệ thống gợi cộng tác dựa trên vùng lân cận (láng giềng) hoạt động bằng cách đếm những mục tin chung hai người dùng đã xem đối với mỗi cặp người dùng trong hệ thống, hoặc số lượng những người dùng chung đã xem những cặp mục tin giống nhau. Độ tương tự giữa hai người hoặc các mục tin được tính toán. Hai người đã xem một lượng lớn các mục tin chung có những sở thích giống nhau. Cần tìm ra những cặp người dùng có sở thích giống nhau nhất hoặc những cặp mục tin có nhiều người dùng nhất đã xem cả hai mục tin. Những cặp người dùng/mục tin đó được gọi là "những láng giềng gần nhất". Hai cách tiếp cận chính của các hệ thống gợi ý dựa trên vùng lân cận là các gợi ý láng giềng gần theo người dùng và theo mục tin.

3.1 Bộ dữ liệu MovieLens

Bộ cơ sở dữ liệu MovieLens 100k bao gồm 100,000 (100k) *ratings* từ 943 *users* cho 1682 bộ phim. Trong bộ cơ sở dữ liệu này gồm nhiều tập tin nhỏ, một trong số các tập tin này gồm [9]:

- u.data: Chứa toàn bộ các đánh giá (*ratings)* của 943 *users* cho 1682 movies. Mỗi user đánh giá ít nhất 20 movies.
- u.user: Chứa thông tin về *users*
- u.item: thông tin về mỗi bộ phim
- ua.base, ua.test, ub.base, ub.test: là hai cách chia toàn bộ dữ liệu ra thành hai tập con, một cho training, một cho test.
- u.genre: Chứa tên của 19 thể loại phim. Các thể loại bao gồm: unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western,

3.2 Giải thuật User K-Nearest-neighbors (user k-NN)

User K-Nearest-neighbors (user k-NN): một ma trận xếp hạng xem xét người dùng-mục tin và ID của người dùng hiện tại như đầu vào, xác định những người dùng khác có sở thích quá khứ giống với những sở thích của người dùng hiện tại. để trả về danh sách đã xếp hạng các mục tin dựa trên những dự đoán đánh giá. Để tính toán độ tương tự giữa những người dùng, có hai phương pháp tính độ tương tự được dùng phổ biến là độ tương quan Pearsn và Cosine. Các giá trị độ tương tự nằm trong khoảng -1 và 1. Thông thường không xem xét tất cả người dùng trong dữ liệu khi tính toán độ tương tự người dùng mà chỉ xem xét k người dùng giống nhất.

Công thức tính toán độ tương tự theo hệ số tương quan Pearson [1]:

$$s(a,u) = \frac{\sum_{i=1}^{n}(r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i=1}^{n}(r_{a,i} - \overline{r_a})^2}\sqrt{\sum_{i=1}^{n}(r_{u,i} - \overline{r_u})^2}}$$

Trong đó

a,u: những người dùng,

$r_{a,i}$, $r_{u,i}$:là đánh giá của người dùng a cho item i và đánh giá của người dùng u cho item i;

i: tập các mục tin, được cả a và u đánh giá;

$\bar{r}_a$, $\bar{r}_u$: những đánh giá trung bình của người dùng a, u;

Phương pháp User_kNN để dự đoán đánh giá của người dùng u trên sản phẩm i được biểu diễn bằng ngôn ngữ giả như sau [5]:

1:procedure USERKNN-CF($\overline{ru}$, r, D^{train})

2: for u=1 to N do

3: Tính Sim_uu'

4: end for

5: Sort Sim_uu' // sắp xếp giảm dần độ tương tự

6: for k=1 to K do

7: Ku ← k // Các người dùng k gần nhất của u

8: end for

9: for i = 1 to M do

10: Tính $\hat{r}_{ui}$

11: end for

12:end procedure

3.3 Đưa ra kết quả dự đoán

Hàm dự đoán đánh giá của sản phẩm i của người dùng a được tính toán như sau[1]:

$$pred(a,i) = \overline{r_a} + \frac{\sum_{u \in N} s(a,u) * (r_{u,i} - \overline{r_u})}{\sum_{u \in N} s(a,u)}$$

Trong đó

$\bar{r}_a$, $\bar{r}_u$: những đánh giá trung bình của người dùng a, u;

s(a,u) độ tương tự giữa người dùng a và u

$r_{u,i}$: đánh giá sản phẩm i của người dùng u

3.4 Đánh giá hệ thống

Việc đánh giá độ chính xác của hệ thống có thể sử dụng căn của sai số bình phương trung bình (RMSE- Root Mean Square Error)

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

Trong đó:

$p_{u,i}$ là giá trị dự đoán đánh giá của người dùng u với mục tin i;

$r_{u,i}$ là giá trị đánh giá thực tế của người dùng u đối với mục tin i.

3.5 Minh họa hệ thống

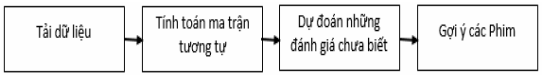
Các bước xây dựng công cụ gợi ý [10]:

Bước 1. Tải dữ liệu

Bước 2. Tính toán độ tương tự của những người dùng

Bước 3. Dự đoán những đánh giá chưa biết cho những người dùng

Bước 4. Gợi ý các mục tin cho những người dùng dựa trên các tính toán điểm người dùng tương tự



Để kiểm tra kết quả đánh giá của công cụ gợi ý cần định nghĩa một hàm đánh giá.

a) Hàm người dùng tương tự

```
def SimPearson(DataFrame,User1,User2,min_common_items=1):
    # GET MOVIES OF USER1
    movies_user1=DataFrame[DataFrame['user_id'] == User1 ]
    # GET MOVIES OF USER2
    movies_user2=DataFrame[DataFrame['user_id'] == User2 ]

    # FIND SHARED FILMS
    rep=pd.merge(movies_user1 ,movies_user2,on='movie_id')
    if len(rep)==0:
        return 0
    if(len(rep)<min_common_items):
        return 0
    res=pearsonr(rep['rating_x'],rep['rating_y'])[0]
    if(isnan(res)):
        return 0
    return res
```

b) Hàm dự đoán

```
class CollaborativeFiltering:
    """ Collaborative filtering using a custom sim(u,u'). """

    def __init__(self,DataFrame, similarity=SimPearson):
        """ Constructor """
        self.sim_method=similarity# Gets recommendations for a person by using a weighted average
        self.df=DataFrame
        self.sim = pd.DataFrame(np.sum([0]),columns=data_train.user_id.unique(), index=data_train.user_id.unique())

    def learn(self):
        """ Prepare data structures for estimation. Similarity matrix for users """
        allUsers=set(self.df['user_id'])
        self.sim = {}
        for person1 in allUsers:
            self.sim.setdefault(person1, {})
            a=data_train[data_train['user_id']==person1][['movie_id']]
            data_reduced=pd.merge(data_train,a,on='movie_id')
            for person2 in allUsers:
                if person1==person2: continue
                self.sim.setdefault(person2, {})
                if person1 in (self.sim[person2]):continue # since is a simetric matrix
                sim=self.sim_method(data_reduced,person1,person2)
                if(sim<0):
                    self.sim[person1][person2]=0
                    self.sim[person2][person1]=0
                else:
                    self.sim[person1][person2]=sim
                    self.sim[person2][person1]=sim

    def estimate(self, user_id, movie_id):
        totals={}
        movie_users=self.df[self.df['movie_id'] ==movie_id]
        rating_num=0.0
        rating_den=0.0
        allUsers=set(movie_users['user_id'])
        for other in allUsers:
            if user_id==other: continue
            rating_num += self.sim[user_id][other] * float(movie_users[movie_users['user_id']==other]['rating'])
            rating_den += self.sim[user_id][other]
        if rating_den==0:
            if self.df.rating[self.df['movie_id']==movie_id].mean()>0:
                # return the mean movie rating if there is no similar for the computation
                return self.df.rating[self.df['movie_id']==movie_id].mean()
            else:
                # else return mean user rating
                return self.df.rating[self.df['user_id']==user_id].mean()
        return rating_num/rating_den
```

c) Hàm đánh giá

```
def evaluate(estimate_f,data_train,data_test):
    """ RMSE-based predictive performance evaluation with pandas. """
    ids_to_estimate = zip(data_test.user_id, data_test.movie_id)
    estimated = np.array([estimate_f(u,i) if u in data_train.user_id else 3 for (u,i) in ids_to_estimate ])
    real = data_test.rating.values
    return compute_rmse(estimated, real)
```

d) Một số kết quả chạy chương trình

Kiểm tra độ tương tự giữa người dùng thứ 1 với người dùng thứ 8 và độ tương tự giữa người dùng thứ 1 với người dùng thứ 31

```
#Let's see how similars are user 1 with 8 and 1 with 31
print("Pearson Similarity",SimPearson(data,1,8))
print("-----")
print("Pearson Similarity",SimPearson(data,1,31))
```

```
Pearson Similarity 0.692086366077
-----
Pearson Similarity -0.0922138891954
```

Kết quả dự đoán đánh giá Movie_id=10 của người dùng User_id=1

```
Estimation the rating of user_id=1 for movie_id=10  3.81944444444444446
```

Đánh giá kết quả dự đoán của công cụ gợi ý

```
RMSE for Collaborative Recommender: 1.00468945461
```

IV. KẾT LUẬN

Có thể nói các hệ thống gợi ý áp dụng các giải thuật khai phá dữ liệu, học máy nhằm giúp thu thập thông tin cá nhân trên Internet, đồng thời giúp giảm bớt vấn đề quá tải thông tin với các hệ thống truy xuất thông tin và cho phép người dùng truy cập vào các sản phẩm và dịch vụ trên hệ thống. Bài viết đã tổng hợp lại những hệ thống gợi ý phổ biến và mô tả, thực thi một hệ thống gợi ý cộng tác láng giềng gần đơn giản với tập dữ liệu mẫu MovieLens nhỏ nhất.

TÀI LIỆU THAM KHẢO

[1] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh; "Recommendation systems: Principles, methods and evaluation"; Egyptian Informatics Journal (2015) 16, 261–273

[2] Nabizadeh Rafsanjani, Amir Hossein and Salim, Naomie and Aghdam, Atae Rezaei and Fard, Karamollah Bagheri (2013) *Recommendation systems: a review.* International Journal of Computational Engineering Research, Vol. 03, Issue.5, pp. 47-52.

[3] Naresh E, Geetha LM, Vijaya Kumar BP; "Recommendation system and its approaches- A survey"; International Journal of Scientific & Engineering Research, Volume 7, Issue 5, May-2016;

[4] Reena Pagare, Shalmali A. Patil; Study of Collaborative Filtering Recommendation Algorithm - Scalability Issue; International Journal of Computer Applications (0975 - 8887), Volume 67 - No. 25, April 2013

[5] Nguyễn Hùng Dũng, Nguyễn Thái Nghe; Hệ thống gợi ý sản phẩm trong bán hàng trực tuyến sử dụng kỹ thuật lọc cộng tác; Tạp chí Khoa học Trường Đại học Cần Thơ, số 31a (2014), trang 36-51

[6] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor; "Recommender Systems Handbook"; Springer; 2011

[7] Aristomenis S. Lampropoulos, George A. Tsihrintzis; Machine Learning Paradigms - Applications in Recommender Systems; Springer; 2015

[8] Charu C. Aggarwal; Recommender Systems – The textbook; Springer; 2016

[9] Laura Igual, Santi Seguí; Introduction to Data Science - A Python Approach to Concepts, Techniques and Applications; Springer; 2017

[10] Suresh Kumar Gorakala; Building Recommendation Engines; Packt Publishing; 2017

COLLABORATIVE FILTERING RECOMMENDER SYSTEM AND MOVIELENS DATASET USED FOR SIMULATING THE USER-BASED NEAREST NEIGHBORHOOD ALGORITHM

Abstract: Recommender System is an information filtering technique used to predict user preferences. Using recommendation systems assists users in deciding and choosing suitable items (product, music, movie, picture, news, book,…) from available data sets. Nowadays, more and more recommender systems have been used in e-commerce, entertainment, education,… This report focuses on common collaborative filtering approaches in recommender systems; and MovieLens dataset is used for simulating the User-based nearest neighborhood algorithm.

Keywords: *collaborative filtering, recommender system, evaluation*