A UNIFY METHOD BETWEEN COLLABORATIVE FILTERING AND CONTENT-BASED FILTERING BASED ON GRAPH MODEL

Manh Son Nguyen, Duy Phuong Nguyen
Posts and Telecommunications Institute of Technology of Vietnam

Abstract: Recommender systems are the capable systems of providing appropriate information and removing unappropriate information for Internet users. The recommender systems are built based on two main information filtering techniques: Collaborative filtering and content-based filtering. Each method exploits particular aspects related to content features or product usage habit of users in the past to predict a brief list of the most suitable products with each user. Content-based filtering perform effectively on documents representing as text but have problems selecting information features on multimedia data. Collaborative filtering perform well on all information formatsbut have problems with sparse data and new users. In this paper, we propose a new unify method between collaborative filtering and content-based filtering based on graph model. The model allows us to shift general hybrid filtering recommender problem to collaborative filtering recommender problem, then build new similar measures based on graph to determine similarities between two users or two items, these similar measures are used to predict suitable products for users in the system. The experimental results on real data sets about films show that the proposed methods utilize advantages effectively and are disadvantages significant limitations of basedline methods.

Keywords: Collaborative Filtering Recommendation, Content-based FilteringRecommendation, Hybrid Filtering Recommendation System, Item-Based Recommendation, User-Based Recommendation;

I. INTRODUCTION

Nowadays, users use online Internet services are always in information overload. To approach useful information, the users must handle and except almost unnecessary information. Recommender systems resolve this problem by giving prediction and providing a brief list of products (website, news, movie, video…) that are appropriate for each user. In fact, the recommender systems are not only toward offload information issues for each user but also decided to success of e-commerce systems [4]. Baseline recommender problem can be stated as below.

Supposedly, we have a finite set U = {u1, u2,…, uN} is the set of N users, P = {p1, p2,.., pM} is the set of M items. Each item px ∈P can be paper, news, merchandise, movie, service or any informational types that the users need. Relationship between the users set U and the items set P are represented by evaluative matric R={ rix: i = 1, 2, ..N; x = 1, 2,..M }. Each value rix represents evaluation of the user ui∈Uwith the item px∈P. Normally, rixhaving a value in the domain F = { 1, 2,.., g}. The value rixcan be collected directly by inquiring user's opinion or indirectly by user's feedback. The value rix = ϕ can understand that the user uihas never given evaluation or known the item pxyet. Actually, the evaluative matrixs of recommender systems are often very sparse. Density of rating values rix≠0 is less than 1%, almost remain rating values are ϕ [4]. The matrix R is the input matrix of collaborative filtering recommender systems.In short px∈P as x∈P; ui∈U as i∈U. The letters i, j always used to refer to the user set in next section of the paper.

Each item x∈P is presented by |C| content features, C = {c1, c2,.., c|C|}. The content feature cs∈C can receive from feature selection methods in the field of information retrieval. For example x∈P is the movie then content features may represent the movie are C={genre, producer, studio, actor, director...}. Conventionally, wx = {wx1, wx2,.., wx|C| } is the weighted vector for content feature values of the item x∈P . Meanwhile, the weighted matrix W = {wxs: x =1, 2, .., M; s =1, 2, .., |C|} is the input of content-based recommender systems based on information of items[2,3,17]. In short, cs∈C as s∈C. The letters is always used to refer to content feature set of items in next section of the paper.

Each user x∈P is presented by |T| content features,T = {t1, t2,.., t|T|}.The content feature tq∈T is usually individual information of each user (Demographic Information).For example,content features of the user i∈U can be T={gender,age, occupation, degree,…}. Conventionally,vi = {vi1, vi2,.., vi|T|} is the weighted vector for content feature values of the user i∈U. Meanwhile, the weighted

Contact author: Manh Son Nguyen
Email: sonnm@ptit.edu.vn
Manuscript received: 7/2022, revised 8/2022, accepted: 8/2022.

matrixV ={viq: i = 1, 2, .., N;q = 1, 2, .., |T| }is the input of content-based recommender systems based on information of users [3,13]. For convenience in representation, I write short tq∈T as q∈T.The letter q is always used to refer to content feature set of users in next section of the paper.

Next, we sign Pi⊆Pis the item set x∈P that is evaluated by the user i∈U and Ux⊆Uis the user set i∈U that gave evaluation about the item x∈P. With each user that need recommendation i∈U (known asthe current user, the user need to be recommended or the active user), tasking recommendatory methods is suggesting K items x∈(P\Pi) that appropriate with the user i.

There are many different proposed to resolve recommender problem. However, we can divide approachesinto three main trends: collaborative filtering recommendation, content-based filtering recommendation, hybrid filtering recommendation. Content-based filtering recommender systems give recommender methods based on the weighted matrix of item content features W={wxs}or the weighted matrix of user content featuresV ={viq} [3,13,17]. In the other hand, collaborative filtering recommender systems give recommender methods based on the evaluative matrix R={rix} [1,2,4]. Hybrid filtering recommender system give recommender methods based on 3 matrixs R, W and V[3,9].

The effectiveness of the hybrid filtering method was confirmed in many researches [2,8]. The most common approach is linear combination method between collaborative filtering and content-based filtering. In this approach, the authors conducted collaborative filtering method and content-based filtering method separately, then combined linearly predictive results of two methods or selected the best candidate from one of two methods[17]. Second approach resolve hybrid filtering recommender problem by combinating features of content-based filtering into collaborative filtering. The second approach is executed by building a data combinative procedure to create input data, the input data included rating values of collaborative filtering and content features. Pazzani [13] proposed the method to present a item profile by a weighted vector of user content features. Using this presentation, the predictive method is gived by Pazzani that is executed by pure collaborative filtering technique. Third approach consider hybrid filtering recommender problem by adding features of collaborative fileting into content-based filtering. Under this method, item content features become central and rating values of users in collaborative filtering as assumed feature values in predictive process[17,18].

The last approach is interested by research community is unified method between collaborative filtering and content-based filtering based on machine learning techniques. Basu[19] proposed way to build a set of features representing for collaborative filtering and content-based filtering. The predictive method is performed by building a set of deductive rules on specific features. Popescul[20] proposed a model to analyse hidden semantic meaning to unify between collaborative filtering and content-based filtering. Balisico and Hofman[21] used multiple funtion to combine similar values from one user to other user, one item to other item, then apply support vector machine to generate predictions. Crammer and Singer[22] consider hybrid filtering recommender problem as raking items by addting item content features.

Relating to graphical models, many different proposals have been given to solve recommender problem. Aggarwall[23] was represented relationships between pairsof users by a directed graph, where each edge is set to reflect degree of similarity between two users. The predictive method is performed by calculating weight of shortest paths between two users. Lien[7] proposed a method to calcule similar measuresbetween pairs of users or pairs of items by a weighted bipart graph model. Similarity degrees of users is done by estimating total weights of all paths from one user vertices to other user vertices, similarity degrees of items is done by estimating total weights of all paths from one item vertices to other item vertices. Phuong[6] proposed a method to combine between collaborative filtering and content-based filtering by building relationships between users and item content features. The predictive method was performed by linear combining all weights of paths from a user vertices to a item vertices. The item have total weights of path are max that become destination of predictive process.

In this paper, we proposed a unify model between collaborative filtering and content-based filtering based on graph representation. The model is built by taking centered collaborative filtering, build user profiles based on evaluative matrix to establish a direct relationship between the user set and the set of item content features. Then, we proceed to build item profiles also based on evaluative matrix to establish a direct relationship between the item set and the set of user content features. Based on the relationship between the user set with the set of item content features and the relationship between the item set with the set of user content features, we determine latent relationship between the item content features with the user content features.In this way, we reduced the general hybrid recommender model to the standard collaborative filtering recommender model.

In principle, after obtained the standard collaborative filtering recommender model, we can deploy any collaborative filtering methods have been proposed before. However, to exploit the strength of graph, we give similarity measures based on graph by evaluating similarity degrees of users based on summary weights of paths from one user vertices to other user vertices, similarity degrees of items based on summary weights of paths from one item vertices to other item vertices. By this way, we can maximize efficiency of search algorithms that has been implemented on the graph. To focus on the proposed methods, in the section 2, we present method to shift hybrid filtering recommender problem to collaborative filtering recommender problem. In the section 3, we present hybrid recommender method based on graph. In the section 4, we present experimental method and compare with baseline methods. The last section is some conclusions.

II. SHIFTING HYBRID FILTERING RECOMMENDER TO PROBLEM COLLABORATIVE FILTERING PROBLEM

As mention above, hybrid filtering recommender problem executes generating prediction using the rating set of users with each item, the item content features and the user content features. In this section, we propose a method to shift hybrid filtering recommender problem to pure collaborative filtering problem by building user profiles and item profiles based on the native rating set of users with items. Based on the user profiles and item profiles had been developed, we determined latent relationship between the set of user content features and the set of item content features to obtain similar model with the model of collaborative filtering recommender problem. To demonstrate the correctness of the proposed method we used graph model to resolve hybrid filtering recommender problem.

2.1. Graphical representative method for hybrid filtering

No limiting generality of the problem stated in section 1, we assume evaluative value of the user $i \in U$ with the item $x \in P$ be determined by the formula (1). Each item $x \in P$ is presented by |C | content features, C = {c1, c2,..,c|C|} is determined by the formula (2). Each user $i \in U$ is presented by |T| content features = {t1, t2,.., t|T|} is determined by the formula (3).

$$r_{ix} = \begin{cases} v & If\ the\ user\ i\ evaluate\ the\ item\ x\ with\ v\ level (v \in F) \\ \phi & If\ the\ user\ i\ hasn't known\ the\ item\ x\ yet \end{cases} \quad (1)$$

$$c_{xs} = \begin{cases} 1 & If\ the\ item\ x\ has\ the\ content\ feature\ s \\ 0 & If\ the\ item\ x\ hasn't\ the\ content\ feature\ s \end{cases} \quad (2)$$

$$t_{iq} = \begin{cases} 1 & If\ the\ user\ i\ has\ the\ content\ feature\ q \\ 0 & If\ the\ user\ i\ hasn't the\ content\ feature\ q \end{cases} \quad (3)$$

The recommender system with the rating matrix R = {r_{ix}: i=1, 2,..,N; x=1, 2,..,M}, the item content feature matrix C={c_{xs}:x=1, 2, .., M; s =1, 2, ..,|C|}, the user content feature matrix T = {t_{iq} : i=1, 2, .., N; q =1, 2, ..,|T|} can be represented as a weighted graph G =(Ω, E), which Ω is the vertices set and E is the edge set. The vertices setΩof the graph is determined by the formula (4) is union of the user set U, the item set P, the set of item content features C and the user content features T. The edge set E of the graph include 3 edge types: the edge (i, x) connect from user vertices with item vertices, the edge (x,s) connect from item vertices with item content feature, the edge (i, q) connect from user vertices with user content feature.

$$\Omega = U \cup P \cup C \cup T \quad (4)$$

$$E = \begin{cases} e = (i,x) & If\ r_{ix} \neq 0 : i \in U, x \in P. \\ e = (x,s) & If\ c_{xs} \neq 0 : x \in P, s \in C. \\ e = (i,q) & If\ t_{iq} \neq 0 : i \in U,\ q \in T. \end{cases} \quad (5)$$

For example, the recommender system include 3 users U = {u1, u2, u3}, 4 items P = {p1, p2, p3, p4}. In there, the rating matrix R is given by the Table 1; the matrix of item content features Cis given by the Table 2; the matrix of user content features T is given by the Table 3. Therefore, represented graph for general recommender problem is presented by Figure 1. The graph is represented by 3 child bipartite graph. The middle child bipartite graph represent option of users with items through the rating matrix R=(r_{ix}). The edge connect from the user vertices $i \in U$ to the item vertices $x \in P$ is weighted by r_{ix}. The top child bipartite graph represent relationship between items with the set of item content features through the matrix C=(c_{xs}). The edge connect from the item vertices $x \in P$ to the item content feature vertices $s \in C$ is weighted by 1. The bottom child bipartite graph represent relationship between users with the set of user content features through the matrix T=(t_{iq}). The edge connect from the user vertices $i \in U$ to the user content feature vertices $q \in T$ is also weighted by 1.

Table 1. *The rating matrix R*

	p_1	p_2	p_3	p_4
u_1	5	ϕ	4	ϕ
u_2	ϕ	4	ϕ	3
u_3	ϕ	5	4	ϕ

Table 2.*The matrix of item content features C*

	c_1	c_2	c_3
p_1	1	0	1
p_2	1	1	0
p_3	1	0	1
p_4	0	1	1

Table 3. *The matrix of user content features T*

	t_1	t_2	t_3	t_4
u_1	1	0	0	1
u_2	1	0	1	0
u_3	0	1	0	1

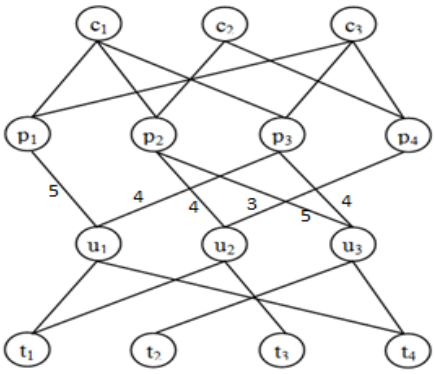

Figure 1. *The graphical representation for recommender system*

Based on the graphical representation above, collaborative filtering recommender method is executed based on edges connecting the user vertices $i \in U$ and the item vertices $x \in P$ with the weight r_{ix} [5]. The item-content-based filtering recommender method is executed based on edges connecting the item vertices $x \in P$ and the item content feature vertices $s \in C$[7]. The user-content-based filtering recommender method is executed based on edges connecting the user vertices $i \in U$ and the user content

feature vertices t∈T[17].The hybrid filtering recommender method is executed based on 3 edge types (i , x), (x, s), (i, q) [9,10].

2.2. Building user profiles based on evaluative matrix

Content recommender methods generate prediction items having informative content or description of goods similar to those items that the user had ever used or accessed in the past. Quality of the methods dependent on methods of feature extraction to represent vector of item content features and vector of item using profiles of the user. The biggest drawback of the feature extraction methods is many content features don't contribute to determine similarity between vector of user profiles and vector of item profiles are still participating in calculation [3,5]. To reduce this issues, we propose method to build item using profiles of the user through rating values of recommender system, then we establish direct relationship between users and each item feature to enhance recommender efficiency. The method is performed below.

To build item using profiles ò the user, we need performing 2 tasks: determining the set of items that the user had ever accessed or used in the past and estimating weight for each item content feature in user profiles. Symbol Pi⊆P is determined by the formula (6) is the set of items that the user i∈Ue valuated the item x∈P. Meanwhile, Pi is the set of items that the user had ever accessed in the past, the set of items is used by content-based recommendation while building user profiles. Remaining problem is how to estimate weight of each item content feature s∈C with each user profile i∈U.

$$P_i = \{x \in P \mid r_{ix} \neq 0 \;\; (i \in U, x \in P)\} \quad (6)$$

Symbol ListItem(i, s) is the set of items x∈Pi containing item content features∈C be determined by the formula (7). Therefore, |ListItem(i , s)|is the number of times the user i∈U using the items x∈P that contain item content feature s∈C in the past.

$$ListItem(i,s) = \{x \in P_i \mid c_{xs} \neq 0 \;\; (i \in U, x \in P, s \in C\} \quad (7)$$

Based on Pi and ListItem(i,s), content-based recommender methods estimate weight w is reflecting importance of the item content features to the user i. The most popular method is often used in building user profiles is the technique TF-IDF. The value w is float number spread around [0,1]. However, while observing collaborative filtering recommender problem, we found itself that have already exist a native assessment of user to item through rating value r_{ix}. The value r_{ix} reflect user's prefer after using items and giving prefer level with items. For example with the movie recommender system, the value r_{ix} = 1, 2, 3, 4, 5 is known by opinion levels "very bad", "bad", "normal", "good", "very good". Because of that, we wanted to get a weigh estimative method of item content features with each user having same native evaluative level of the value r_{ix}.

To perform the above idea, we implement observation ListItem(i, s). If the value |ListItem(i, s)| exceeds a certain threshold θthen weigh of the item content feature s∈C with the user i∈U that be calculated by average of all rating values. In the other hand, if |ListItem(i, s)| is less than θ, the value wis is calculated by sum of all rating value then divide for θ. In experiment, we calculated average number of all users i∈U rated the items x∈P, then we chose θ equivalent with 2/3 the average number of ratings that the user i∈U rated the item x∈P containing the feature s∈C. In this way, we can limit some item content features the user isn't interest but still be evaluated with high weights.

$$w_{is} = \begin{cases} \frac{1}{|ListItem(i,s)|}\sum_{x \in ListItem(i,s)} r_{ix} & If \;\; |ListItem(i,x)| \geq \theta \\ \frac{1}{\theta}\sum_{x \in ListItem(i,s)} r_{ix} & If \;\; |ListItem(i,x)| < \theta \end{cases} \quad (8)$$

The value wis is estimated by the formula (8) reflecting opinion of the user i∈U with item content features s∈C, this is also the profile of user i∈U used the item content feature s∈C in the past. Easily find wis∈F, while F = {1, 2, .., g}. So, we can treat each item content feature acts as assistant item complementing to the set of items. Based on this observation, we extend the bipartite graph of primitive collaborative filtering recommender problem (the middle child graph) by staying at the set of user vertices U, the set of item vertices is extended by P∪C. Link between the user vertices i∈U and the item vertices x∈P will be established if $r_{ix} \neq 0$. Link between the user vertices i∈U and the item feature vertices s∈C will be established if wis ≠ 0. The extended rating matrix will be determined by the formula (9).

$$r_{ix} = \begin{cases} r_{ix} & If \; x \in P \; and \; r_{ix} \neq 0 \\ w_{is} & If \; s \in C \; ad \; w_{is} \neq 0 \; (x = s) \end{cases} \quad (9)$$

For example, the representative graph for hybrid filtering recommender system is shown by the Figure1, chosen θ = 2 we'll calculate the extend rating matrix in Table 4and extended collaborative filtering recommender graph is shown by the Figure 2. The red edges are new edges be complemented to bipartite graph of collaborative filtering.

***Table 4**. The extended rating matrix R*

	p_1	p_2	p_3	p_4	c_1	c_2	c_3
u_1	5	0	4	0	4	0	4
u_2	0	4	0	3	2	3	1
u_3	0	5	4	0	4	2	2

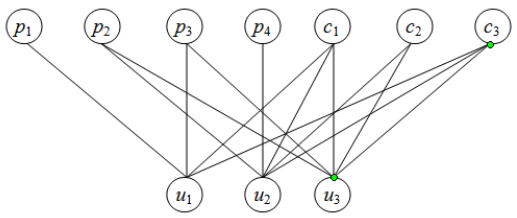

***Figure 2**. The graph expands following item side.*

2.3. Building item profiles based on evaluative matrix

Similar to user profiles, item profiles record trace of user content features using item. To build item profiles, we need performing 2 tasks: determining the set of users that had ever used the item in the past and then estimating weight of each user content feature in item Meanwhile, Ux is the

set of users that need recording user content features in item profiles. Remaining problem is how to estimate weight of each user content feature q∈T with each item profile x∈P.

$$U_x = \{i \in U \mid r_{ix} \neq 0 \ \ (i \in U, x \in P)\} \qquad (10)$$

Symbol ListUser (x, q) is the set of users i∈Ux containing user content feature q∈T be determined by the formula (11). Therefore, |ListUser(x , q)| is the number of times the item x∈P be used by the users i∈U having user content feature q∈T in the past.

$$ListUser(x,q) = \{i \in U_x \mid t_{iq} \neq 0 \ \ (i \in U, x \in P, q \in T\} \qquad (11)$$

Based on Ux and ListUser(x, q), content-based recommender methods estimate weight t_{xq} reflecting importance of the user content feature q to the item x. Same as user, item itself have already exist a native assessment of users set with the item through rating value r_{ix}. Because of that, we propose a weigh estimative method of user content features with each item having same native evaluative level of the value r_{ix}. To perform the above idea, we implement observation ListUser(x, q) .If the value |ListItem(i, s)| exceeds a certain threshold θ then weigh of the user content feature q∈T with the item x∈P is v_{xq} that be calculated by average of all rating values. In the other hand, if |ListUser(x, q)| is less than θ, the value v_{xq} is calculated by sum of all rating value then divide for θ. In experiment, we calculated average number of all items x∈P are rated by the user i∈U, then we chose θ equivalent with 2/3 number of users i∈U containing the feature q∈T using the item x∈P. In this way, we can limit some user content features are less interest to the item but still be evaluated with high weights.

$$v_{xq} = \begin{cases} \frac{1}{|ListIUser(x,q)|}\sum_{i \in ListUser(x,q)} r_{ix} & If \ \ |ListUser(x,q)| \geq \theta \\ \frac{1}{\theta}\sum_{i \in ListUser(x,q)} r_{ix} & If \ \ |ListUser(x,q)| < \theta \end{cases} \qquad (12)$$

The value v_{xq} is estimated by the formula (12) representing the item profile x∈P are used by the user i∈U containing the feature q∈T. Easily find v_{xq}∈F, while F = {1, 2, .., g}. So, we can treat each user content feature acts as assistant user complementing to the set of users. Based on this observation, we extend the bipartite graph of collaborative filtering recommender problem in the section 2.2 by staying at the set of item vertices P∪Cand extending the set of user vertices to U∪T. Link between the item vertices x∈P and the user vertices i∈U will be established if $r_{ix} \neq 0$. Link between the item vertices x∈P and the user feature vertices q∈Twill be established if $v_{xq} \neq 0$. The extended rating matrix recorded weight of edges (x, i) and (x, q) will be determined by the formula (13).

$$r_{ix} = \begin{cases} r_{ix} & If \ i \in U, x \in P \ and \ r_{ix} \neq 0 \\ w_{is} & If \ i \in U, s \in C \ and \ w_{is} \neq 0 \ (x = s) \\ v_{xq} & If \ x \in P, q \in T \ and \ v_{xq} \neq 0 \ (x = q) \end{cases} \qquad (13)$$

For example, the representative graph for hybrid filtering recommender system is shown by the Figure 1, chosen θ = 2 we'll calculate the extended rating matrix in Table 5 and extended collaborative filtering recommender graph is shown by the Figure 3. The blue edges are new edges be complemented to bipartite graph of collaborative filtering.

***Table 5**. The extended rating matrix R*

	p_1	p_2	p_3	p_4	c_1	c_2	c_3
u_1	5	0	4	0	4	0	4
u_2	0	4	0	3	2	3	1
u_3	0	5	4	0	4	2	2
t_1	2	2	2	1			
t_2	0	0	2	0			
t_3	0	2	0	1			
t_4	2	2	4	0			

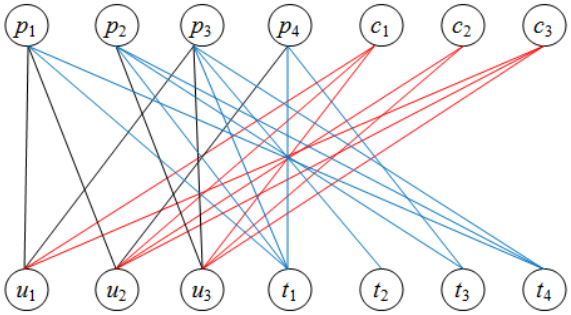

***Figure 3**. The graph expands following user side.*

2.4. Building relationship between user features and item features

The user profiles are determined according to the formula (8), the item profiles are determined according to the formula (12). They was based on native rating of users with items and usage habit for items of users. Clearly, the set itself of user content features and the set itself of item content features are also exist a native relationship between user profiles and item profiles. For example, why children like watching cartoons, teen girls like watching romantic films, teen boys like watching active films…? We believe that exploiting the above latent relationship will significantly improve predictive quality items that appropriate with each user.

To determine latent relationship between the user content feature q∈T and the item content feature s∈C, we build two different kinds of observation. The first observation will perform from user profiles to item content features. The second observation will perform from item profiles to user content features. Since both kinds of observation only purpose determining latent relationship between the pair of features q∈T and s∈C so we combine results between two kinds of observation to obtain final result. The detail method will perform below.

Observing from user profiles to item content features:

Symbol Uq is the set of users i∈U containing user content feature q∈T be determined by the formula (14). Symbol UserAttr(i,s) is the set of users i∈U containing user content feature q∈T rated the items x∈P containing the item

content feature s∈C be determined by the formula (15).Therefore, relationship between the feature q∈T and the feature s∈C is estimated by the formula (16). With, wis is the user profile i∈U are determined according to the formula (8),

$$U_q = \{i \in U \mid t_{iq} \neq 0\} \quad (14)$$

$$UserAttr(q,s) = \{i \in U_q \mid w_{is} \neq 0\} \quad (15)$$

$$a_{qs} = \begin{cases} \frac{1}{|UserAttr(q,s)|}\sum_{i \in UserAttr(q,s)} w_{is} & If\ |UserAttr(q,s)| \geq \theta \\ \frac{1}{\theta}\sum_{i \in UserAttr(q,s)} w_{is} & If\ |UserAttr(q,s) < \theta \end{cases} \quad (16)$$

The value a_{qs} is estimated by (16) reflecting effect level of the feature s∈C to the set of users containing the feature q∈T. If the number of users i∈U containing the feature q∈T rated the items x∈P containing the feature s∈C exceeds a certain threshold θ then a_{qs} be calculated by averaging weights of the features s in user profiles. In the other hand, the value a_{qs} is calculated by sum of weights of the features sin user profiles then divide for θ.In this way, we can limit some user content features or some item content features are less used by users but still be evaluated with high weights.

Observing from item profiles to user content features:

Symbol Ps is the set of items x∈P containing item content feature s∈C be determined by the formula (17). Symbol ItemAttr(q, s) is the set of items containing the item content feature s∈C be rated the set of users x∈Pi∈U containing the user content feature q∈T that is determined by the formula (18).Therefore, appropriate levels of the set of items containing the feature s with the set of users i∈U containing the feature q are determined according to the formula (19). With v_{xq} is item profile x∈P is determined by (12).

$$P_s = \{x \in P \mid c_{xs} \neq 0\} \quad (17)$$

$$ItemAttr(q,s) = \{x \in P_s \mid v_{xq} \neq 0\} \quad (18)$$

$$b_{qs} = \begin{cases} \frac{1}{|ItemAttr(q,s)|}\sum_{x \in ItemAttr(q,s)} v_{xq} & If\ |ItemAttr(q,s)| \geq \theta \\ \frac{1}{\theta}\sum_{x \in ItemAttr(q,s)} v_{xq} & If\ |ItemAttr(q,s) < \theta \end{cases} \quad (19)$$

The value b_{qs} is estimated by (19) reflecting effect level of the feature q∈T to the set of items containing the feature s∈C. If the number of items x∈P containing s∈Care rated by users i∈U containing the feature q∈T exceeds a certain threshold θ then b_{qs} be calculated by averaging weights of the features qin item profiles. In the other hand, the value b_{qs} is calculated by sum of weights of the features q in user profiles then divide for θ. In this way, we can limit some user content features or some item content features are less used by users but still be evaluated with high weights.

Combining two kinds of observation above:

As mention above, the value a_{qs} is determined by (16) and b_{qs} is determined by (19) both reflect usage habit of users containing the feature q with the set of items containing the feature s. The only difference between a_{qs} and b_{qs} is the kind of observation based on user profiles or item profiles. To reconcile both kinds of observation, we choose averaging value of a_{qs} and b_{qs} following the formula (20). With, the value d_{qs} is established if and only if the items containing the feature s are really interested by many users and vice versa, many users containing the feature q are really interested in items containing the feature s. This is entirely consistent with general sentiment of the peoples using items.

$$d_{qs} = \begin{cases} \frac{1}{2}(a_{qs} + b_{qs}) & If\ a_{qs} \neq 0\ và\ b_{qs} \neq 0 \\ 0 & Otherwise \end{cases} \quad (20)$$

After determining relationship between user content features and item content features, we extend the bipartite graph of collaborative filtering recommender problem in the section 2.3 by supplementing links between each feature s∈C and the feature q∈T. The final graph we receive having the set of user vertices U, the set of item vertices P, the set of user content features T and the set of item content features P. The vertices of graph are separated into 2 sides, one side is U∪T and another side is P∪C. The edges set of the graph contain 4 kind of edges: the edge (i , x) link user vertices and item vertices weighted by r_{ix}, the edge (i , s) link user vertices and item content feature vertices weighted by wis, the edge (q , x) link user content feature vertices and item content feature vertices weighted by v_{qx}, the edge (q , s) link user content feature vertices and item content features weighted by d_{qs}.

$$r_{ix} = \begin{cases} r_{ix} & If\ r_{ix} \neq 0\ (i \in U\ and\ x \in P) \\ w_{is} & If\ w_{is} \neq 0\ (i \in U\ and\ x = s \in C) \\ v_{qx} & If\ v_{qx} \neq 0\ (i = q \in T\ and\ x \in P) \\ d_{qs} & If\ d_{qs} \neq 0\ (i = q \in T\ and\ x = s \in C) \end{cases} \quad (21)$$

***Table 6**. The extended rating matrix R*

	p_1	p_2	p_3	p_4	c_1	c_2	c_3
u_1	5	0	4	0	4	0	4
u_2	0	4	0	3	2	3	1
u_3	0	5	4	0	4	2	2
t_1	2	2	2	1	2	1	1
t_2	0	0	2	0	1	1	1
t_3	0	2	0	1	1	1	0
t_4	2	2	4	0	4	1	3

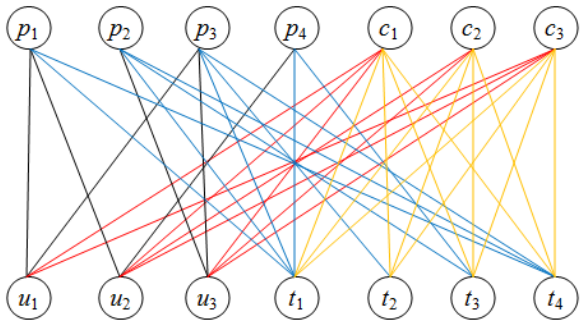

***Figure 4.**The graph represent hybrid recommender filtering problem*

For example, the representative graph for hybrid filtering recommender system is shown by the Figure 1, chosen θ = 2 we'll calculate the extended rating matrix in Table 6 and extended collaborative filtering recommender graph is shown by the Figure 4. The yellow edges are new edges be complemented to bipartite graph of collaborative filtering.

The extended rating matrix is proposed by (21) fully integrated ratings of collaborative filtering, user profiles, item profiles, relationships between user profiles and item profiles of content-based filtering. Weights of content features in user profiles, item profiles and relationship between content features having same metric with rating value. Therefore, the methods of collaborative filtering based on memory [15,16] or the methods of content-based filtering based on model [6,11,12] can be deployed on the extended rating matrix. This is the main contribution of the paper in building a unify model between collaborative filtering recommendation and content-based filtering recommendation.

III. PREDICTIVE METHODS BASED ON THE HYBRID GRAPH

After shifting hybrid recommender problem to standard collaborative filtering recommender problem, in principle, we can deploy any collaborative filtering recommender method based on the extended rating matrix. Within the paper, we propose to extend methods of collaborative filtering recommender based on memory by expanding correlative measures based on extended rating matrix. Then, we build a similarity measure based on searching engine on graph. The experimental results on real data sets show that the proposed methods achieve superior performance compared to baseline methods.

3.1. Similarity measure between pairs of users based on graph

One of the biggest challenges of recommender systems is sparse data problem [1,3]. The problem occur when known rating values ($r_{ix}\neq0$) very little, less than with unknown rating values ($r_{ix}=0$). The current similarity measures calculated similar degree between the user $i\in U$ and the user $j\in U$ based on the set of intersection items $Pi\cap Pj$. When the number of intersection items $|Pi\cap Pj|$ is small, this will make calculating similarity between the user i and the user inaccurate. In the case $|Pi\cap Pj|=0$, similarity between the user I and the user j will not be identified. This affects directly to predictive quality of recommender methods based on user.

The method to determine similarities between pairs of users can be done easily on graph model by considering all paths that length equals 2 from one user vertices to other user vertices. There are two types of path having length 2 from the user vertices i to the user vertices j on hybrid graph. The first type comes from the user vertices i to the item vertices x through rating edges (i,x). For example, the path u1-p3-u3belongs the first type that is used to determine similarity between the user u1 andu3. The second type comes from the user vertices i to the item feature vertices s through the edges of item feature (i, s). For example, the path u1-c1-u3, u1-c3-u3belong second type that is used to determine similarity between the user u1 and u3. Weight of each path having length 2 is calculated by multiple weights of each edge. Similarity between two users is calculated by sum weights of all paths having length 2 between them. The pair of users i, j that total weights of paths having length 2 is greater then similarity between them is higher. Collaborative filtering method based on users predict appropriate items for each user based on total weights of paths that belong first type. Content filtering method predict appropriate items for each user based on total weights of paths that belong second type. Hybrid filtering method predict appropriate items for each user based on total weights of both types.

In case of sparse data when number of ratings differ 0 lowly, this will lead to number of the edges (i, x) determined by (9) lowly and number of the edges (i, s) determined by (13) also lowly. This makes predictive results of the above methods achieving not high. To reduce this problem, we execute extending path lengths from user vertices to other user vertices to leverage indirect relationship between pairs of users and pairs of different content features. Paths can be the rating edges (i, x), edges (i, s), edges (q, x) or edges (q, s).

For example, to determine similarity between u2andu3on bipartite graph representing hybrid filtering recommender problem in the Figure 4, we use some paths u2-p1-u1-p3-u3, u2-p4-t3-p2-u3, u2-c1-t4-p3-u3. This is quite reasonable because u2likesp1, p1is liked by u1, u1 likes p3, p3 is liked by u3 so indirectly, u2 is similar with u3 at a certain degree. Or in another case, u2 likes p4, p4 is liked by the user containing content feature t3, the user containing content feature t3 likes p2, u3 likes t2 so indirectly, u2 is similar with u3 at a certain degree. Or u2 likes c1, c1 is appropriate with the set of users containing the content feature, t4 is appropriate with the item p3, u3 likes p3 so indirectly, u2 is similar with u3 at a certain degree.

Because hybrid filtering recommender graph is a bipartite graph so paths from user vertices to other user vertices are always even natural number (2, 4, 6, 8) [7]. Weight of each path is calculated by multiple weights of each edge so path pass through the edges having high weights are still be appreciated, path pass through the edges having lower weights are still underestimated. To give priority to the shortest path (length equals 2), we use the parameter α ($0<\alpha<1$) to underestimate weights of high length paths. Specifically, the method for estimating total weights of the path having length L from user vertices to other user vertices is determined by the formula (22) [7].

$$R^L = \begin{cases} R.R^T & If\ L = 2 \\ \alpha.R.R^T.R^{L-2} & If\ L = 4, 6, 8, \ldots \end{cases} \quad (22)$$

In there, L is path length, R is extended rating matrix be determined by (21), RT is the transpose matrix of R. The even value L is determined when every $r_{ij}^L\neq0$. Total weights of path have length L from the vertices $i\in U$ to other vertices $j\in U$ be similarity between the user i and the user j. K users $j\in U$ have the highest value $r_{ij}^L$to become neighbors set of the user $i\in U$. Based on this observation, we adjust step 1 of Hybrid-User Based algorithm in the

section 3.1 to Hybrid-User Based-Graph graph in the Figure 5.

Hybrid-UserBased-Graph algorithm:

Input:

- *The extended rating matrix R= (r_{ix}) represents hybrid graph be determined by (21).*
- *$i \in U$ is the active user.*
- *K is the number of users in neighbors set.*

Output:

- *Prediction x: $r_{ix} \mid x \in P \setminus P_i$ (rating of the user i with new items $x \in P$).*

Steps:

Step 1. *Calculating similarities between pairs of users on the hybrid graph:*

L ⇐2;//Set the initial path length

Repeat

$$R^L = \begin{cases} R.R^T & nếu\ L = 2 \\ \alpha.R.R^T.R^{L-2} & nếu\ L = 4,6,8,\ldots \end{cases}$$

L⇐L + 2; //Increasing length of path.

Until ($r_{ij}^L \neq 0$ with every$j \in (U \setminus i)$);

Step 2. *Determining neighbors set for the user $i \in U$.*

- Arranging $r_{ij}^L \neq 0$ in descending order ($i \neq j$).
- Selecting *K*first users $j \in U$ to become neighbors set of the user i (Symbol: the neighbors set of the user $i \in U$ is K_i).

Step 3. *Predicting rating of the user ifor the items $x \in P \setminus P_i$.*

$$r_{ix} = \frac{1}{|K_i|}\sum_{j \in K_i} r_{ix}$$

Step 4. *Seleting K items that have the highest value r_{ix} to*

Figure 5. *Hybrid-User Based-Graph algorithm.*

3.2. Similarity measure between pairs of items based on graph

Method determine similarities between pairs of items can be done easily on graph by considering all paths having length 2 from item vertices to other item vertices on graph. For example, to determine similarity measure between the item p1 and p3 on graph in the Figure 4, we based on some paths: p1-u1-p3, p1-t1-p3, p1-t2-p3. Weight of each path can calculated by multiple weights of corresponding edges. Total weights of all paths itself from the vertices x∈P to the vertices y∈P is similarity between the two users. K items have the highest total weights of paths from the vertices x∈P to the vertices y∈P become neighbors set of the item x. Then using the neighbors set to generate prediction about the most appropriate items for the user I [7].

To reduce effect of sparse data problem, we execute extending path lengths from item vertices to other item vertices to leverage indirect relationship between pairs of items and pairs of different content features. Paths can be the rating edges (i, x), edges (i, s), edges (q, x) or edges (q, s). For example, to determine similarity between p1 and p2on bipartite graph representing hybrid filtering recommender problem in the Figure 4, we use some paths p1-u1-p3-u2-p2, p1-u2-p4-t1-p2, p1-t2-c3-u3-p2. The rationality of this deduction is also explained similarly with the case of calculating similarities between pairs of users.

Because hybrid filtering recommender graph is a bipartite graph so paths from item vertices to other item vertices are always even natural number (2, 4, 6, 8). Weight of each path is calculated by multiple weights of each edge so path pass through the edges having high weights are still be appreciated, path pass through the edges having lower weights are still underestimated. To give priority to the shortest path (length equals 2), we use the parameter α ($0<\alpha<1$) to underestimate weights of high length paths. Specifically, the method for estimating total weights of the path having length L from item vertices to other item vertices is determined by the formula (23) [7].

$$R^L = \begin{cases} R^T.R & If\ L = 2 \\ \alpha.R^T.R\ .R^{L-2} & If\ L = 4,6,8,\ldots \end{cases} \quad (23)$$

In there, L is path length, R is extended rating matrix be determined by (21), RT is the transpose matrix of R. The even value L is determined when every $r_{xy}^L \neq 0$. Total weights of path have length L from the vertices x∈P to other vertices y∈P be similarity between the two items. K item y∈P have the highest value r_{xy}^L to become neighbors set of the itemx∈P. Based on this observation, we adjust step 1 of Hybrid-Item Based algorithm in the section 3.2 to Hybrid-Item Based-Graph graph in the Figure 6.

Hybrid-ItemBased-Graph algorithm:

Input:

- *The extended rating matrix R= (r_{ix}) represents hybrid graph be determined by (21).*
- *$i \in U$ is the active user.*
- *K is the number of items in neighbors set.*

Output:

- *Prediction x: $r_{ix} \mid x \in P \setminus P_i$(rating of the user i with new items $x \in P$).*

Steps:

Step 1.*Calculating similarities between pairs of items on the hybrid graph:*

L ⇐2;//Set the initial path length

Repeat

$$R^L = \begin{cases} R^T.R & nếu\ L = 2 \\ \alpha.R^T.R.R^{L-2} & nếu\ L = 4,6,8,\ldots \end{cases}$$

L⇐L + 2; //Increasing length of path.

Until ($r_{xy}^L \neq 0$ with every$y \in (P \setminus x)$);

Step 2. *Determining neighbors set for the itemx $\in P$.*

- Arranging $r_{xy}^L \neq 0$ in descending order ($x \neq y$).
- Selecting *K* first items $y \in P$ to become neighbors set of the item x (Symbol: the neighbors set of the item $x \in P$ is K_x).

Step 3. *Predicting rating of the user ifor the items $x \in P \setminus P_i$.*

$$r_{ix} = \frac{1}{|K_x|}\sum_{x \in K_x} r_{ix};$$

Step 4. *Seleting K items that have the highest value r_{ix} to recommend for the user i.*

Figure 6. *Hybrid-Item Based-Graph algorithm.*

IV. EXPERIMENT AND EVALUATION

To evaluate effectiveness of proposed methods for hybrid filtering recommendation, we experiment on real data set of movies[24]. The above representative methods are evaluated and compared to baseline methods below.

4.1. Data set

The hybrid filtering recommender method is experimented by the data set MovieLens of the research group GroupLens belong to Minnesota university[24]. MovieLens subsets have three options with different sizes respectively: MovieLens 100k, MovieLens 1M and MovieLens 10M. We selected MovieLens 1M because this subset provides full movie content features as well as user content features. The subset MovieLens 1M includes 1MB ratings of 6040 users for 3952 movies. Rating levels set from 1 to 5. Sparse level of rating data is 99.1%.

Detailed datas provide in files:

- u.data: store full 1MB ratings of 6040 users for 3952 movies. Each user rate 20 movies at least. Each row have same struct: user id | item id | rating | timestamp.
- u.info: store number of users, number of items, number of ratings of data set.
- u.item: store information of movies.
- u.genre: store list of 19 types of movies diffently. This is item content features that are used to experiment proposed method.
- u.user: store information of users. Each row have same struct: user id | age | gender | occupation | zip code. User id is used by the file u.data.
- u.occupation: store list of occopations. Thí is user content features that are used to experiment proposed method.

4.2. Experimental method

At first, all experimental data set is divided into 2 parts, one part Utr be used as training data, the rest data Ute is testing data.The U_{tr} contains 75% ratingsandU_{te} contains 25% ratings. The training data is used to build model following above representative algorithm. Each user *i*belongs to the testing data, exited ratingsof the active user is divided into 2 parts O_{i} and P_{i}. O_{i} is known, whereas P_{i} is ratings that need prediction from the training data and O_{i}.

Forecasting error MAE_{u} for eah user *u*belongs to testing data is calculated by averaging absolute errors between predicted value and actual value with all items of P_{u}.

$$MAE_u = \frac{1}{|P_u|}\sum_{y\in P_u}|\hat{r}_{uy} - r_{uy}| \quad (38)$$

Forecasting error over the testing data is calculated by averaging predicted errors of each users belongs to U_{te}.If the value MAE is small, the predictive method will give high accuracy.

$$MAE = \frac{\sum_{u\in U_{te}} MAE_u}{|U_{te}|} \quad (39)$$

4.3. Comparison and evaluation

The hybrid filtering recommender method based on users*Hybrid-UserBased-Graph*are proposed by 4.1 be compared with baseline methods below:
The method CF-User Based use the correlative measure Pearson. This is the standard collaborative filtering recommender method based on users. In there, similarities between pairs of users are calculated based on a set of intersection items between two users[15].

The hybrid filtering method based on users (symbol as Hybrid-User Based) use the correlative measure Pearson. This is hybrid recommender method based on the correlative measure Pearson[15]. In there, similarities between pairs of users are calculated on extended rating matrix toward to items side following (9).

The hybrid filtering recommender method based on items*Hybrid-ItemBased-Graph*are proposed by 4.2 be compared with baseline methods below:
The method CF-Item Based use the correlative measure Pearson. This is the standard collaborative filtering recommender method based on items. In there, similarities between pairs of items are calculated based on a set of users that rated items [15].

The hybrid filtering method based on items (symbol as Hybrid-Item Based) use the correlative measure Pearson. This is hybrid recommender method based on the correlative measure Pearson[15]. In there, similarities between pairs of items are calculated on extended rating matrix toward to users side following (13).

Choosing θ = 15 follows the above representative methods to deterimined w_{is}, v_{qx}, d_{qs} in order of the formulas (8), (12), (20). Choosing α=0.8 to determine weights of paths following the formulas (22), (23). The experimental method choose randomly *1000, 2000, 4000*users in the set MovieLens to make training data. Choosing randomly 300, 600, 1000users in remain set to become testing data.The value*MAE*in the Table7 and Table8 are estimated by average of 10 times of random experiment.

The results on Table7 show that the filtering method based on pure usersCF-UserBased give the highest MAE with remain methods. This may explain limitations of collaborative filtering methods in training process that only based on the small set of value $r_{ix}\neq 0$. When size of training data set large then predictable results of the methods are improved gradually. Specifically, the values MAE on the data set consisting1000, 200, 400 users be respectively(0.865, 0.859, 0.855), (0.846, 0.841, 0.836), (0.824, 0.817, 0. 813)in order. The large neighbors set perform not proportional to the results expected.This result is entirely consitent with the previous researchs.

The Hybrid-UserBased method give the value MAE much lower than the CF-UserBased method. Specifically, the size of neighbors set K=10 and the training data set contains1000, 2000, 4000 users then MAE values are in order 0.793, 0.798, 0.782 in comparison with0.865, 0.846,

0. 824 of the CF-UserBased method; When K=20 MAE values are in order 0.792, 0.788, 0.738 in comparison with 0.859, 0.841, 0.817 of the CF-UserBased method; When K=30 MAE values are in order 0.791, 0.782, 0.715 in comparison with 0.855, 0.836, 0.813 of the CF-UserBased method. The number of users in neighbors set are large making predictive results more stable. This may explain the Hybrid-UserBased method calculating similarity between pairs of users more accuracy because the method be executed on total rating data set and user profiles. So, the Hybrid-UserBased method determine neighbors set of the active user to give predictive results better.

Table 7. *MAE of recommender methods based on users*

Size of training data set	Method	Size of neighbors set		
		10	20	30
1000 users	CF-USERBASED	0.865	0.859	0.855
	HYBRID-USERBASED	**0.793**	**0.792**	**0.791**
	HYBRID-USERBASED-GRAPH	**0.672**	**0.629**	**0.687**
2000 users	CF-USERBASED	0.846	0.841	0.836
	HYBRID-USERBASED	**0.798**	**0.788**	**0.782**
	HYBRID-USERBASED-GRAPH	**0.632**	**0.629**	**0.598**
4000 users	CF-USERBASED	0.824	0.817	0.813
	HYBRID-USERBASED	**0.782**	**0.738**	**0.715**
	HYBRID-USERBASED-GRAPH	**0.694**	**0.629**	**0.696**

MAE values in the Table 8 of some filtering methods based on items are similar with filtering methods based on users. MAE values of the hybrid filtering method Hybrid-ItemBased is much smaller than the CF-ItemBased method. Reason of this happening can only explain the methods to calculate similarities between pair of items be performed on ratings set and item profiles are more accuracy than the methods based on only ratings set. MAE values of the Hybrid-ItemBased-Graph method are significant lower than the Hybrid-ItemBased method. This can only explain similarities between items based on graph have combined all indirect relationships between users, items, user profiles and item profiles.

Table 8. *MAE of recommender methods based on items*

Size of training data set	Method	Size of neighbors set		
		5	10	20
1000 users	CF-ITEMBASED	0.894	0.883	0.875
	HYBRID-ITEMBASED	**0.781**	**0.788**	**0.794**
	HYBRID-ITEMBASED - GRAPH	**0.668**	**0.674**	**0.633**
2000 users	CF-ITEMBASED	0.838	0.831	0.827
	HYBRID-ITEMBASED	**0.751**	**0.737**	**0.713**
	HIBRID-ITEMBASED - GRAPH	**0.696**	**0.639**	**0.617**
4000 users	CF-ITEMBASED	0.811	0.806	0.801
	HYBRID-ITEMBASED	**0.788**	**0.711**	**0.714**
	HYBRID-ITEMBASED - GRAPH	**0.648**	**0.619**	**0.611**

V. CONCLUSIONS

The paper proposed a unify model between collaborative filtering recommender methods and content-based fitlering recommender methods. The model is built by shifting hybrid filtering recommender problem to standard collaborative filtering recommender problem to leverage advantages of the method. The shifting method is performed by building user profiles of content-based filtering based on natural rating of users with items. Then, establishing direct relationships between users and each item content features. In this way, we extend the rating matrix of collaborative filtering toward items side. Next, the process of building item profiles is also done based on natural usage habit of users with items. Based on item profiles, we established direct relationships between items and each user content features. In this way, we extend the rating matrix of collaborative filtering toward user side. Finally, we sought determining latent relationships between each item content feature and item content features based on user profiles and item profiles. The last model is expansion of the baseline collaborative filtering model.

After collapsing to collaborative filtering problem, the extended rating matrix proposed be integrated fully all rating values of collaborative filtering, user profiles, item profiles, relationships between user profiles and item profiles. Weights of content features in the user profiles, item profiles and relationships between content features having same matric with rating values. So, collaborative filtering recommender methods based on memory or collaborative filtering recommender methods based on model can be deployed on the extended rating matrix. To take advanges of graph model, we proposed bulding similarity measures to explore indirect relationships between users, items, user content features, item content features to improve predicted results. The experimental results on real data sets show that the proposed hybrid filtering recommender methods achieve superior performance compared to baseline methods. We believe that the model will give good results with recommender

methods based on model. These results will be presented by next researches of the paper.

REFERENCES

1. Su X., Khoshgoftaar T. M., "*A Survey of Collaborative Filtering Techniques.*,". Advances in Artificial Intelligence ,2009, pp.1-20.
2. Adomavicius G., Tuzhilin A., "*Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*", IEEE Transactions On Knowledge And Data Engineering, vol. 17, No. 6, 2005.
3. Robin D. Burke, "*Hybrid Recommender Systems: Survey and Experiments*". User Model. User-Adapt. Interact. 12(4): 331-370 (2002).
4. M. D. Ekstrand, J. T. Riedl and J. A. Konstan, "*Collaborative Filtering Recommender System*". Foundations and Trends in Human–Computer Interaction, Vol 4, No2, 2010, pp 81:173.
5. Nguyen Duy Phuong, Le Quang Thang, Tu Minh Phuong, *"A Graph-Based Method for Combining Collaborative and Content-Based Filtering.* PRICAI 2008: 859-869.
6. Nguyen Duy Phuong, Tu Minh Phuong, "Collaborative Filtering by Multi-task Learning", RIVF 2008, pp: 227-232.
7. Do Thi Lien, Nguyen Duy Phuong, "*Collaborative Filtering with a Graph-based Similarity Measure".* ComManTel, 2014, pp. 251-256.
8. Asela Gunawardana, Guy Shani, "*A Survey of Accuracy Evaluation Metrics of Recommendation Tasks*. Journal of Machine Learning Research 10: 2935-2962 (2009).
9. Asela Gunawardana, Christopher Meek, " *A unified approach to building hybrid recommender systems*". RecSys 2009: 117-124.
10. Robin D. Burke, Fatemeh Vahedian, Bamshad Mobasher, "*Hybrid Recommendation in Heterogeneous Networks*". UMAP 2014: 49-60.
11. J. Wang, A. P. de Vries, and M. J. T. Reinders., "*Unifying user-based and item-based collaborative filtering approaches by similarity fusion*.,". In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '06). ACM, New York, NY, USA, 501-508.
12. Raghavan, S., Gunasekar, S., Ghosh, J. "*Review quality aware collaborative filtering*". In Proceedings of the sixth ACM conference on Recommender systems, pp. 123–130. ACM(2012).
13. Pazzani, M.J. " *A framework for collaborative, content-based and demographic filtering*", Artificial Intelligence Review 13(5-6), 393–408 (1999).
14. Herlocker J.L., Konstan J.A., Terveen L.G., and Riedl J.T., "*Evaluating Collaborative Filtering Recommender Systems*", ACM Trans. Information Systems, vol. 22, No. 1 (2004), pp. 5-53.
15. Breese J. S., Heckerman D., and Kadie C., "*Empirical analysis of Predictive Algorithms for Collaborative Filtering*", In Proc. of 14th Conf. on Uncertainty in Artificial (1998).
16. Sarwar B., Karypis G., Konstan J., and Riedl J., "*Item-Based Collaborative Filtering Recommendation Algorithms*", Proc. 10th Int'l WWW Conf (2001).
17. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M. "*Combining content-based and collaborative filters in an online newspaper*". In: Proceedings of ACM SIGIR workshop on recommender systems, vol. 60. Citeseer (1999).
18. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. : Combining contentbased and collaborative fillters in an online newspaper. Proceedings of ACM SIGIR Workshop on Recommender Systems.(1999).
19. Basu, C., Hirsh, H., And Cohen, W.: Recommendation as classification: Using social and content-based information in recommendation. In Proceedings of the 15th National Conference on Artificial Intelligence, 714–720. (1998).
20. Popescul A., Ungar L.H., Pennock D.M., and Lawrence S.: Probabilistic Models for Unified Collaborative and Content-Based Eecommendation in Sparse-Data Environments, Proc. 17th Conf. Uncertainty in Artificial Intelligence, (2001).
21. Balisico J., Hofman T.: Unifying collaborative and content-based filtering. In Proceedings. of Int. Conf. on Machine learning (ICML-04) (2004).
22. Crammer, K., and Singer, Y: Pranking with ranking. Advances in Neural Information Processing Systems 14 pp. 641-647. (2002).
23. Aggarwal C.C., Wolf J.L., Wu K.L., and Yu P.S.: Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering, Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, Aug. (1999).
24. http://www.grouplens.org/
25. Poonam B. Thorat, R. M. Goudar, Sunita Barve: Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System, International Journal of Computer Applications, Volume 110 – No. 4, (2015)

MỘT PHƯƠNG PHÁP HỢP NHẤT LỌC CỘNG TÁC VÀ LỌC THEO NỘI DUNG DỰA TRÊN MÔ HÌNH ĐỒ THỊ

Tóm tắt: Hệ thống tư vấn là hệ thống có khả năng cung cấp thông tin thích hợp và loại bỏ thông tin không phù hợp cho người dùng Internet. Hệ thống tư vấn được xây dựng dựa trên hai kỹ thuật lọc thông tin chính: Lọc cộng tác và lọc dựa trên nội dung. Mỗi phương pháp khai thác các khía cạnh cụ thể liên quan đến đặc tính nội dung hoặc thói quen sử dụng sản phẩm của người dùng trong quá khứ để dự đoán danh sách ngắn gọn các sản phẩm phù hợp nhất với từng người dùng. Lọc dựa trên nội dung hoạt động hiệu quả trên các tài liệu biểu diễn dưới dạng văn bản nhưng gặp vấn

đề khi lựa chọn các đặc tính thông tin trên dữ liệu đa phương tiện. Lọc cộng tác hoạt động tốt trên tất cả các định dạng thông tin nhưng có vấn đề với dữ liệu thưa thớt và người dùng mới. Trong bài báo này, chúng tôi đề xuất một phương pháp hợp nhất giữa lọc cộng tác và lọc dựa trên nội dung dựa trên mô hình đồ thị. Mô hình đề xuất cho phép chúng ta chuyển bài toán tư vấn lọc kết hợp chung sang bài toán tư vấn lọc cộng tác, sau đó xây dựng các độ đo tương tự mới dựa trên đồ thị để xác định sự tương đồng giữa hai người dùng hoặc hai sản phẩm. Các độ đo tương tự này được sử dụng để dự đoán sản phẩm phù hợp cho người dùng trong hệ thống. Kết quả thực nghiệm trên tập dữ liệu thực về phim cho thấy các phương pháp đề xuất phát huy được hiệu quả và hạn chế đáng kể các nhược điểm của phương pháp trước đó.

Từ khóa: Tư vấn lọc cộng tác, tư vấn dựa trên lọc nội dung, hệ thống tư vấn lọc kết hợp, tư vấn dựa trên sản phẩm, tư vấn dựa trên người dùng.

BIOGRAPHY

Duy Phuong Nguyen was born in Hanoi, Vietnam, in 1965. He received the Ph.D. degrees from VNU University of Engineering and Technology (VNU-UET) in 2010. He is head of Information Technology Faculty, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam.

His research interests include machine learning, recommender systems, graph applications, automated testing techniques, optimization techniques for online programming systems.

Manh Son Nguyen was born in Hanoi, Vietnam, in 1981. He graduated from the Institute of Posts and Telecommunications Technology (PTIT) in 2004. He received M.E degree from VNU University of Engineering and Technology (VNU-UET) in 2010. He is currently a Lecturer in Information Technology Faculty, PTIT.

His main research interests include data mining, collaborative filtering, machine learning applications in online programming systems.