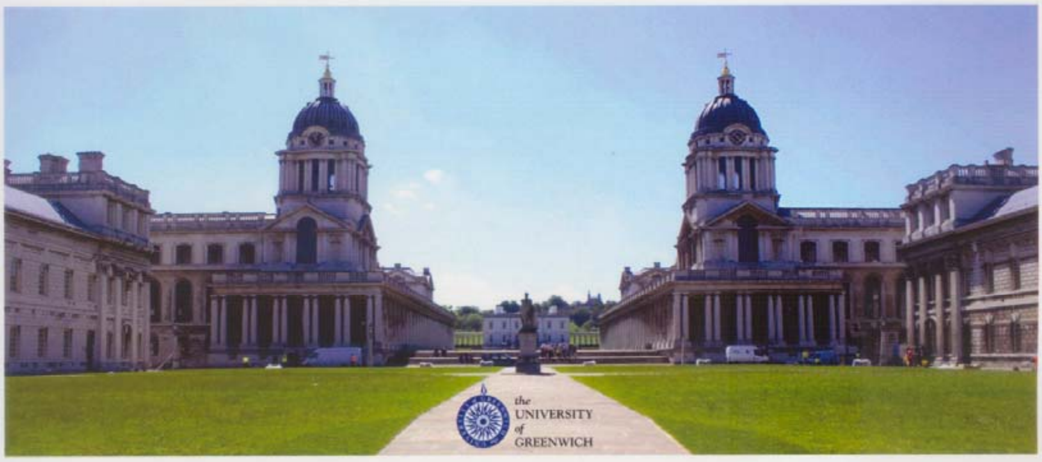
the UNIVERSITY of GREENWICH

Proceedings

11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD2010)



9-11 June 2010
The University of Greenwich, London, United Kingdom

Editors:
Dr. Jixin Ma, The University of Greenwich, United Kingdom
Dr. Liz Bacon, The University of Greenwich, United Kingdom
Dr. Wencai DU, Hainan University, China
Dr. Miltos Petridis, University of Greenwich, United Kingdom

Sponsored by the International Association for Computer & Information Science (ACIS)

2010 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing

SNPD 2010

Table of Contents

Artificial Intelligence

User and Item Pattern Matching in Multi-Criteria Recommender Systems

Pitaya Poompuang
Graduate School of IT in Business
Siam University
Bangkok, Thailand
p_pitaya@yahoo.com

Wichian Premchaiswadi
Graduate School of IT in Business
Siam University
Bangkok, Thailand
wichian@siam.edu

Abstract— **Information on the ratings of several features of items can be deployed to improve the quality of recommendations in recommender systems by incorporating them into similarity calculation between any two users or two items. However, the incremental information of these features has important impacts on recommender systems. For example, the complexity of similarity calculation is increased and more resources are consumed during the process for generating recommendations. In this paper, we propose several techniques by using this information to provide relevant recommendations and to reduce the complexity in similarity computation by directly matching between preferences of user and the strength of item features.**

Keywords-recommender system; recommendation; multi-criteria; user profile; item profile; transformation; reduction; pattern matching

I. INTRODUCTION

One of the several core aspects of Web 2.0 is to invite people to interact and participate in web applications, for example, providing their contents, assigning ratings on product items, giving comments to online articles, and etc. This occurrence causes information provided by people are more and more increased and led to the information overload problem [6],[12],[13]. Recommender systems [4],[17],[19],[20] have become the key tool for helping users to filter out information and alternatives based on their preferences. Traditional recommender systems utilize only overall rating information of item as single feature or single criteria to analyze and generate recommendations for users. Currently, the modern recommender system, Yahoo! Movies is trying to leverage several rating information on other features of item, i.e., story, genre, director, and author as multi-criteria to improve the quality of movies recommender systems.

Adomavicius et al [1] gave a conclusion of their findings that changing only the similarity function in the traditional collaborative filtering technique to reflect multi-criteria rating information should results in a more accurate identification of similar users and, consequently, in better recommendation quality. Adomavicius et al [1] also suggested two different approaches to incorporate this information in similarity calculation of recommender system. The first approach is the aggregating traditional similarities from individual criteria. In this approach, the similarity between two entities, such as between two users or between two items, is evaluated by using some standard similarity metric, such as cosine-based and correlation-based [1]. Then, all evaluated similarity of individual features between two entities can be aggregated to generate an overall similarity in several ways, such as averaging all individual similarities and worst-case similarity. The second approach is the using multidimensional distance metrics in similarity calculation. In this approach, each corresponding individual feature rating of the two entities is represented by a point in k +1 dimension space. Then, the standard multi-dimensional distance metrics such as Manhattan distance and Euclidean distance [1] can be applied to calculate distance between two entities. Finally, the distance metric is translated to similarity metric.

Even though the methods suggested by [1] can provide good personalized recommendations, but for gaining the effective recommender system, many resources such as storages and memories are required in the process of calculating the similarity measure between two entities based on their several features. The complexity of similarity computation is also linearly increased by the number of features. In this paper, we propose the methods to reduce the complexity and resources consuming in multi-criteria recommender systems. These techniques can reduce the number of dimensions of a user profile and reduce the number of individual user profiles into a smaller set of a user profiles. The results of these techniques are a new user profiles, which has the same structure as the traditional item profile. The new user profile contains the preferences information of each user over several item features, which is called the global features of a user. The global features of a user generally indicate the user's preference for the items, i.e. preference on watching movies. It is not the preference on any particular movie. We also propose the new technique to quantify the strength degree of item features based on users' opinions to generate new item profile. Finally, we propose the simple overall rating prediction techniques by summarizing multi-criteria rating information on several features to provide more accurate recommendations for a particular user. This technique relies on the similarity score between two entities from different classes, the user class and the item class, rather than relies on the similarity score between two entities from the same class.

This paper is organized as follows: Section 2 presents related work. Section 3 addresses some more detail about the background of recommender system. Section 4 presents the

978-0-7695-4088-7/10 $26.00 © 2010 IEEE
DOI 10.1109/SNPD.2010.13

proposed transformation and reduction techniques to reduce the calculation complexity and resources consuming in multi-criteria recommender system. Section 5 presents a conclusion and discusses some perspectives and ideas for future work.

II. RELATED WORKS

The traditional problem of multi-criteria rating is to find the items that are optimal in general (i.e., optimal for all users), not intended for environment of personalized recommendation. Currently, this problem is considered as one of the important issues for the next generation of recommender system [4].

Ricci et al[11] develop a personalizing travel recommender system by using case-base reasoning technique, ranking and aggregating elementary items (locations, activities, services) based on the user's preferences and a repository of past travels. This system, however, does not consider each elementary item as a multi-criteria, just performs optimization over a multidimensional solution space. There are several projects that make a comparison of the items based on each attribute's weights ranking [8],[9],[10]. The weight of each attribute is directly obtained from the individual user. However, the value of ranking for each attribute is the same for all users.

Schafer [5] implements a meta-recommendation system that allows users to specify their preference for each content attribute (e.g., movie genre, MPAA rating, or film length) and also allows them to set an important condition for recommender by rating the importance of these attributes to filter the recommendations toward what the users really want. This system, however, does not represent a multi-criteria rating environment because the users only specify general filtering requirements for all movies, such as the preferred value and weights for movie genre attribute.

In multi-criteria rating environment, users can specify their subjective ratings for various feature components of individual item. This information can be used for prediction and personalization purposes. Adomavicius et al [1] incorporate and leverage multi-criteria ratings in calculation of the similarity between two different users or two different items in two different ways, i.e., using aggregating traditional similarities and multidimensional distance metrics, then predict the rating using the weighted sum or adjusted weighted sum in the same way as with a standard collaborative filtering algorithm. Using an aggregating similarity method, the similarity between users or items is calculated based on each individual criterion by using some standard similarity metric such as cosine-based and correlation-based. The overall similarity can be computed by aggregating the individual similarities in several ways such as average sum and adjusted weighted sum of individual similarities approach. Using a multidimensional distance metrics method, each rating can be represented as a point in the k+1 dimension space and distance between ratings or points is calculated by using standard multidimensional distance metrics, such as Manhattan distance, Euclidean distance, and so on. Finally, the distance metrics then is further transformed to similarity metrics.

In this paper, we focus on the aggregating similarity method proposed by [1] as mentioned above. Let's consider the case of using single criteria in recommendation systems, the characteristics of users are described by the list of overall feature rating information for items that users have rated. This information can be represented as a User-Item metric. When applying the aggregating similarity method [1] in multi-criteria rating environment, several individual feature ratings will be represented with several individual User-Item metrics (one metric per one feature). For example, if there are four criteria to be considered, four individual User-Item metrics are required as shown in the Table I.

Applying collaborative filtering approach with multi-criteria ratings information, the number of comparing a current user with other users will increase by the number of User-Item metrics. For example, assuming that, there are 100 users and 1,000 items available to recommend in the recommender system domain. To generate recommendations for a particular user with collaborative-filtering approach, the system has to calculate the similarity measure between a current user and other users based on individual feature or each single-criteria rating. The number of comparing user with other users is 99 * 1,000 * 1= 99,000 times. Owing to the four criteria shown in Table I, the number of comparing will increase up to 99 * 1,000 * 4 = 396,000 times. Therefore, the conclusion is that the more features yield the more complexity in comparison.

TABLE I. REPRESENTATION OF FOUR CRITERIA WITH FOUR INDIVIDUAL USER-ITEM METRIC

(a)

Users	m1	m2	m3	m4	m5
u1	3	0	2	3	0
u2	2	0	4	0	2
u3	1	3	5	0	4

Users	m1	m2	m3	m4	m5
u1	2	0	3	3	0
u2	2	0	2	0	3
u3	1	2	2	0	3

(c)

Users	m1	m2	m3	m4	m5
u1	2	0	3	3	0
u2	2	3	2	0	4
u3	2	1	2	0	3

Users	m1	m2	m3	m4	m5
u1	3	0	3	3	0
u2	2	0	3	0	2
u3	1	3	3	0	4

III. BACKGROUND

A. Basic Approaches to Recommender Systems

In general, recommender systems can be classified into two categories based on how the recommendations are generated. The first one is a content-based approach [2],[7]. In this approach, the method to generate recommendation relies on the description information of an item. The characteristics of an item are compared with other items which user has rated. The items with high rating and high similarity to items that the user has rated will be recommended. The second one is a collaborative-filtering approach [2],[3],[7],[10],[13],[14],[19]. In this approach, the method to generate recommendation relies on the correlations between individual rating information provided by users. Items that users, who have high similarity rate to the current user, have rated will be considered. The items with high rating will be recommended.

In summary, both basic recommendation approaches rely on similarity measure. The content-based approach relies on the similarity of items, whereas the collaborative filtering approach relies on the similarity of users. Since most of the traditional recommender systems are developed by using either one of two basic approaches. However, there were several good research papers in the literature that proposed the techniques to combine both basic approaches together in order to improve the performance of recommender system. They are known as a "Hybrid approach" [2],[7]. All the approaches mentioned in this section, the recommendations are indirectly generated by matching the item-item or the user-user in the same class. Conversely, we suggest that the reliability and the quality of recommendations can be improved by directly matching between the characteristic of a current user and the characteristic of items in the system domain. This approach will recommend items that its' characteristics are more satisfied to the characteristic of users.

B. Single Criteria and Multi-Criteria Recommender Systems

Since recommendation process in recommender system is usually concerned with two entities, item and user. Items can be the kinds of things that the users interest in such as, books, photos, movies, blogs, web sites, and etc. General characteristic of user can be explained by user demographic information such as user name, age, gender, education, occupation, and son on. In addition, the characteristics of user can also be explained by the activity information provided by the user when he or she interacts with an automated recommender system. For example, users expressed their preference on product items by giving a rating to the product items they have consumed. Most recommender systems apply the collaborative-filtering technique to generate recommendations by analyzing user activity information which are known as an "overall feature rating" of item. This information represents overall preference of a user or the single criteria in selecting relevant items for a particular user. In general, the user activity information is represented as User-Item metric. See Table II. A value in each row presents a profile of any user, which is called a "user profile". A value contained in each cell reflects the preferences of user on a particular item as single criteria rating. That is why the traditional recommender system is viewed as single-criteria rating recommendation system.

TABLE II. USER PROFILES AS A USER-ITEM METRIC

Users	m1	m2	m3	m4	m5
u1	3	0	2	3	0
u2	2	0	4	0	2
u3	1	3	5	0	4

Similarly, items are explained by their attached features. These features may be gained by several ways. For example, the characteristic of a product might be assigned by its provider and web content might be assigned by user tagging. The information of items can be represented as the Item-Feature metric as shown in Table III. Each row represents an individual item profile. Each cell reflects the meaning of existing features or attributes, "1" represents the presence of feature and "0" represents the absence of feature.

TABLE III. ITEM PROFILES AS A ITEM-FEATURE METRIC

Items	Action	drama	comedy
m1	1	1	1
m2	1	1	0
m3	0	1	1
m4	1	0	1
m5	0	1	1

IV. METHODOLOGIES

The case study for a movie recommendation was selected for describing our proposed methods. Let's consider genre of movie, different movies generally have different kinds of the number and the type of genre. For example, a movie m1 contains three types of genre; action, drama, and comedy, whereas a movie m2 contains two types of genre; action and drama. In a real application, some movies may have the same type of genre but it is possible that each type of genre may be different in its strengths or level of importance. The conclusion is that even we know what the existing types of genre in a movie, but we still cannot know the strength degree of each type.

Similarly, users may have different in their taste for watching movies based on genre of movies they like. For example, let's consider Table IV(b), a user u1 may equally prefer all three types of genre, action, drama and comedy. A user u2 prefers drama rather than comedy and prefers comedy rather than action. It can be inferred that a user u2 favors drama the most. This information can be analyzed and used to distinguish the users based on their taste.

In fact, for the large single-criteria recommender system, the data in a user profile is very sparse, the proportion of total of movie items that user have watched and the total of movie items in domain are very small. Similarly, in the context of multi-criteria recommender system, the sparse of data in user profiles is increased by multiplying with the number of features of items. Therefore, we proposed the method to transform several traditional user profiles, User-Item metrics, for each feature of item into new single user profiles, User-Feature metric. In addition, an efficient collaborative filtering method was developed to quantify the strength for each feature to generate the new item profiles, Item-Feature Metric. Then, the preferences of user on features and the strength of items features can be matched to generate the recommendations.

The overall rating of a movie has the hierarchy relationship with its features. It means that ratings on these features are ingredient of overall rating of a movie. Therefore, it is possible to predict ratings of these features from the overall feature rating based on the preference of user rather than directly asking user to provide rating on each feature.

A) Quantify the Preference Pattern of User

Our transformation and reduction techniques for finding the pattern of a particular user can be done in three steps by analyzing the frequency of existing features in each item that the users have rated.

1) Join the traditional user and item profile.

Let's consider the structure of the traditional user profiles in Table II and the structure of the traditional movie profiles in Table III. The user "u1" has rated the three movies; m1, m3, and m4. Each movie has different features as shown in Table III. Notice that both tables contain movie id. Therefore, the tables can be joined together with a pair of corresponding movie id as shown in Table IV(a). The Table IV(b) presents the relationship between user and item features.

TABLE IV. JOINING USER - ITEM AND ITEM - FEATURE METRICS.

Action	1	1	0	1	0
Drama	1	1	1	0	1
Comedy	1	0	0	1	1
	m1	m2	m3	m4	m5
u1	3	0	2	3	0
u2	2	0	4	0	2
u3	1	3	5	0	4

(a)

Users	Ratings	Movies	Action	Drama	Comedy
u1	3	m1	1	1	1
u1	2	m3	0	1	0
u1	3	m4	1	0	1
u2	2	m1	1	1	1
u2	4	m3	0	1	0
u2	2	m5	0	1	1
u3	1	m1	1	1	1
u3	3	m2	1	1	0
u3	5	m3	0	1	0
u3	4	m5	0	1	1

(b)

2) Count the frequency of existing features associated with the items.

Focusing on each row or each user profile, the frequency number of features that user has concerned are derived by (1) and the results are shown in Table V.

$$F_u(a_k) = \sum_{i=1}^{n} a_{i,k} \tag{1}$$

$F_u(a_k)$ is a function that returns the frequency number of the k^{th} feature of items that user u has rated. a_{ik} is the k^{th} order feature of an item i.

TABLE V. THE FREQUENCY OF FEATURES RELATED WITH A USER, U3

Items	Ratings	Action	Drama	Comedy
m1	1	1	1	1
m2	3	1	1	0
m3	5	0	1	0
m5	4	0	1	1
	$F(a_k)$	2	4	2

3) Normalize the frequency for each individual feature.

Once the frequency number of features concerned with a particular user is counted, the weight of each feature can be calculated by dividing $F_u(a_k)$ with the sum of all features concerned with a particular user as in (2).

$$w_{u,k} = \frac{F_u(a_k)}{\sum_{j=1}^{m} F_u(a_j)} \tag{2}$$

$w_{u,k}$ is the weight of the k^{th} feature of the items that concern with a particular user u.

Each weight of feature can be combined as a vector of weight as (3) to indicate a pattern of a particular user.

$$\overrightarrow{W_u} = (w_{u,1}, w_{u,2}, \dots, w_{u,m}) \tag{3}$$

Therefore, we can describe users with the degree of items' features, which is called "global features", they have concerned with. The global features of users reflect the general preference or general style of each user in watching movies, not specific for any particular movie in domain.

TABLE VI. THE WEIGHT OF FEATURES RELATED WITH A USER U3

Items	Ratings	Action	Drama	Comedy
m1	1	1	1	1
m2	3	1	1	0
m3	5	0	1	0
m5	4	0	1	1
	$F(a_k)$	2	4	2
	$Norm(F(a_k))$	0.25	0.50	0.25

For example, the pattern of user u3 is assigned with the weight of global features represented as a preference weight vector, (0.25, 0.50, 0.25). From this information, it is concluded that the most feature user u3 has preferred is drama because most of movies, he or she selected to watch, contain this feature as shown in Table VI.

When the last two steps are iteratively applied with overall users in the system, the patterns of all users will be identified. These patterns are used to explain the characteristic of users in the sense of their taste in watching movie as shown in the Table VII. It is called the collection of the patterns of users as the new user profiles. Note that, these new user profiles have the same structure as the item profile as shown in Table III.

TABLE VII. NEW USER PROFILES

Users	Action	Drama	Comedy
u1	0.33	0.33	0.33
u2	0.17	0.50	0.33
u3	0.25	0.50	0.25

B) Quantify the Strength of Item's Feature

Our technique to investigate the pattern of a particular item involves a two-step process.

1) Extract the features weights of a particular item from preference weight vector of users.

Traditional movie item profile as shown in Table III contains the information about the features of movies in binary values format. These values tell us about the existing of movie's features which can be represented by a binary vector as (4).

$$\overrightarrow{B_i} = (b_{i,1}, b_{i,2}, \dots, b_{i,m}) \tag{4}$$

$\overrightarrow{B_i}$ is the binary value vector that describes the item i in which $b_{i,k}$ is the binary value of the k^{th} feature.

The important features of a particular movie for a particular user can be extracted from the preference weight vector by multiplying each pair of the corresponding elements of the preference weight vector with the binary vector of a movie as (5).

$$w_{u,k} \cdot b_{i,k} \tag{5}$$

$w_{u,k}$ is the weight of the k^{th} feature in user u point of view. $b_{i,k}$ is the binary value of the k^{th} feature of an item i.

For example, the user u3 is described by preference weight vector as (0.25, 0.50, 0.25), To extract the features weight vector for movie m2 in the user u3's point of view, each value of the corresponding position between two vectors is multiplied each other, ($w_{u3,m2} * w_{m2,1}$, $w_{u3,m2} * w_{m2,2}$, $w_{u3,m2} * w_{m2,3}$). The final result is (0.25, 0.50, 0.00).

To limit the value of each extracted feature derived from the previous example in the range of 0 to 1, we normalize each value in the resulting vector with the sum of all element values. It is derived by (6).

$$w_{u,i,k} = \frac{w_{u,k} \cdot b_{i,k}}{\sum_{j=1}^{m}(w_{u,j} * b_{i,j})} \tag{6}$$

$w_{u,i,k}$ is the weight of the k^{th} feature in the user u's point of view on an item i.

For example, let's consider a weight value of vector for m2 in the user u3 point of view, (0.25, 0.50, 0.00), the value of each element in the resulting vector can be normalized by the sum of each feature weight of movie m2 as below.

$$\sum_{j=1}^{3}(w_{u3,j} * b_{m2,j}) = 0.25 + 0.50 + 0.00 = 0.75$$

$$w_{u3,m2,action} = 0.25/0.75 = 0.33$$

$$w_{u3,m2,drama} = 0.50/0.75 = 0.67$$

Similarly, the item i is described by the importance of features in the user point of view in term of vector, which is derived by (7).

$$\overrightarrow{W}_{u,i} = (w_{u,i,1}, w_{u,i,2}, \dots, w_{u,i,m}) \tag{7}$$

$\overrightarrow{W}_{u,i}$ is the vector of weight in the user u point of view on item i. From the above example, the opinions of user u3 on the movie m2 can be represented as a vector $\overrightarrow{W}_{u3,m2} = (0.33, 0.67, 0.00)$

2) Aggregate each weight of item features that is derived from all users to generate the pattern for items.

After applying Step1 to the overall users, who have concerned with the particular item, the pattern of items can be generated by aggregating each weighting vector from each user point of view on that item as in (8). Each pattern of item is represented as a local weight vector as (9). The patterns of all items are shown in Table VIII.

$$w_{i,k} = avg_{u \in U_i} \frac{w_{u,k} \cdot b_{i,k}}{\sum_{j=1}^{m}(w_{u,j} * b_{i,j})} \tag{8}$$

$$\overrightarrow{W}_i = (w_{i,1}, w_{i,2}, \dots, w_{i,m}) \tag{9}$$

U_i is a set of users who have watched an item i.

Additionally, the average of overall rating of each movie also includes the new movie profile to indicate its general movie rating information as shown in Table VIII, column 2.

TABLE VIII. NEW ITEM PROFILES

Items	Ratings	action	drama	comedy
m1	2.00	0.25	0.44	0.31
m2	3.00	0.33	0.67	0.00
m3	3.67	0.00	1.00	0.00
m4	3.00	0.50	0.00	0.50
m5	3.50	0.00	0.63	0.37

C) Generate Recommendations

Obviously, the structure of the new user profiles and the structure of new item profiles are the same. Therefore, we can directly make recommendations by mean of matching between the user and the item pattern based on their feature values. Finally, the rating of each feature of an item can be estimated using (10).

$$r_{u,i,k} = w_{i,k} \cdot sim(\overrightarrow{W}_{u,i}, \overrightarrow{W}_i) \cdot r_i \tag{10}$$

$r_{u,i,k}$ is a predicted rating of the k^{th} feature on a particular item i for a particular user u. $w_{i,k}$ is the weight of the k^{th} feature of an item i. The $sim(\overrightarrow{W}_{u,i}, \overrightarrow{W}_i)$ is any similarity function that compare the vector of weights for a user u on item i, $\overrightarrow{W}_{u,i}$, with the vector of features' weights (degrees of features' strengths) for a particular item i , $\overrightarrow{W}_i$. r_i is the overall rating score of an item i provided by user u.

Although all features rating on a particular item can be calculated, but most recommender system favors to recommend items based on their predicted overall rating. It can be computed by summing up all predicted rating of every feature of an item as in (11) or (12).

$$r_{u,i} = \sum_{j=1}^{m} r_{u,i,j} \tag{11}$$

$$r_{u,i} = sim(\overrightarrow{W}_{u,i}, \overrightarrow{W}_i) \cdot r_i \tag{12}$$

$r_{u,i}$ is a predicted overall rating of an item i.

For example, assume that recommender would like to recommend user u2 the movie m4. Since the pattern of movie m4 is represented as a local weight vector, $\overrightarrow{W}_{m4} = (0.50, 0.00, 0.50)$, and the binary vector of movie m4 is $b_{m4} = (1, 0, 1)$.The pattern of user u2 can be represented as a preference weight vector, $\overrightarrow{W}_{u2,m4} = (0.34, 0.00, 0.66)$.

Assume that the similarity between user u2 and movie m4 is 0.951 and average rating for m4 is 3. The prediction rating of feature action, and comedy can be calculated by using (10).

$$r_{u2,m4,1} = 0.50 * 0.951 * 3 = 1.427$$

$$r_{u2,m4,3} = 0.50 * 0.951 * 3 = 1.427$$

Applying (11), the overall rating prediction of m4 for a user u2 can be calculated.

$$r_{u2,m4} = r_{u2,m4,1} + r_{u2,m4,3} = 2.853$$

V. CONCLUSION AND FTURE WORKS

In the context of multi-criteria recommender system, the quality of recommendations can be improved by incorporating several features ratings information into similarity calculation process. However, the calculation complexity and a number of resources usages, such as memory and storage, are immensely increased. In this paper, the transformation and reduction techniques were proposed to reduce the calculation complexity by condensing the dimensions of user profiles and generating a new user profiles. The technique to quantify the strength degree of each item features based on users' preferences was also proposed. It was used to generate a new item profiles. The new user profiles and the new item profiles generated by our approaches have similar pattern in their structure. Finally, the technique to generate recommendations was presented by directly matching the preference of user with the strength degree of item features. Obviously, the two challenges of this work are quantifying preference of users based on their behavior and quantifying the strength of item based on user preferences. Our future work based on this study, we will explore the use of probabilistic models in quantifying important characteristics of users and items for multi-criteria rating recommender system.

REFERENCES

[1] Adomavicius G. and Kwon Y., "New Recommendation Techniques for Multi-Criteria Rating Systems," IEEE Intelligent Systems, Special Issue on Recommender Systems, 20(3), 2007.

[2] Burke R., "Hybrid web recommender systems," In The Adaptive Web, Springer Berlin Heidelberg, pp. 377–408 , 2007

[3] Schafer J.B., Frankowski D., Herlocker J., and Sen S., "Collaborative Filtering Recommender Systems," In Brusilovsky P., Kobsa A., Nejdl W. (eds.) , "The Adaptive Web: Methods and Strategies of Web personalization," Lecture Notes in Computer Science, Vol. 4321., Springer-Verlag, Berlin Heidelberg New York ,2007.

[4] 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, pp. 734-749, Jun. 2005

[5] Schafer J. B., "DynamicLens: A Dynamic User-Interface for a Meta-Recommendation systems," Beyond personalization 2005: A workshop on the next stage of recommender systems research at the *ACM Intelligent User Interfaces Conf.*, Jan. 2005.

[6] O'Donovan J., and Smyth B., "Trust in recommender systems," In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI'05)*, pp. 167–174,2005.

[7] Burke R., "Hybrid recommender systems: Survey and experiments," User Modeling and User-Adapted Interaction, 12(4): pp.331–370., 2002.

[8] W. Lee, C. Liu, and C. Lu, "Intelligent agent-based systems for personalized recommendations in Internet commerce," *Expert Systems with Applications*, vol. 22, no. 4, pp. 275-184, May 2002.

[9] W. Lee, C. Liu, and C. Lu, "Collaborative Filtering with Multi-component Rating for Recommender Systems," *Expert Systems with Applications*, vol. 22, no. 4, pp. 275-184, May 2002.

[10] W. Lee, C. Liu, and C. Lu, "Incorporating Multi-criteria Ratings in Recommendation System," *Expert Systems with Applications*, vol. 22, no. 4, pp. 275-184, May 2002.

[11] Ricci F., Werthner H., "Case-Based Querying for Travel Planning Recommendation," *Information Technology and Tourism*, vol. 4, nos. 3–4, pp. 215–226, 2002.

[12] Schafer J.B., Konstan J. A., and Riedl J., "E-commerce recommendation applications," Data Mining and Knowledge Discovery, 5(1/2): pp.115–153,2001.

[13] Schafer J. B., Konstan J. and Riedl J., "Recommender Systems in E-Commerce," In: *EC '99: Proceedings of the First ACM Conference on Electronic Commerce*, Denver, CO, pp. 158-166,1999.

[14] Herlocker J., Konstan J., Borchers A., and Riedl J., "An Algorithmic Framework for Performing Collaborative Filtering," In Proceedings of ACM SIGIR'99. ACM press.1999.

[15] J. S. Breese, D. Heckerman, and C. Kadie., "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proc.14th Conf. on Uncertainty in Artificial Intelligence*, July 1998.

[16] M. Balabanovic and Y. Shoham., "Fab: Content-Based, Collaborative Recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66-72, 1997.

[17] Resnick P., and Varian H. R. "Recommender systems," Communications of the ACM, 40(3): pp.56–58.,1997.

[18] J. Gray A., Bosworth A., Layman, and Pirahesh H. "Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total," In *ICDE*, Feb. 1996.

[19] Shardanand U. and Maes P., "Social information filtering: algorithms for automating word of mouth," In Proceedings of the Conference on Human Factors in Computing Systems (CHI'95), pp. 210–217,1995.

[20] Resnick P., Iacovou N., Suchak M., Bergstrom P., and Riedl J., "Grouplens: An open architecture for collaborative filtering of netnews," In Proceedings ACM Conference on Computer-Supported Cooperative Work, pp.175–186,1994.