International Journal of Innovative Computing, Information and Control
Volume **13**, Number **6**, December **2017**

ICIC International ©2017 ISSN 1349-4198
pp. **2025–2038**

ENHANCING PERSONAL RECOMMENDATION SYSTEM USING FAMILIARITY FACTOR ON SOCIAL NETWORK

Rung-Ching Chen, Hendry and Lijuan Liu

Department of Information Management
Chaoyang University of Technology
No. 168, Jifeng E. Rd., Wufeng Dist., Taichung 41349, Taiwan
{ crching; s10314905; ljliu }@cyut.edu.tw

Received March 2017; revised July 2017

Abstract. *Social media has become one of the most popular web platforms, with an enormous user base. Many application systems use social media information to generate search results. A recommendation system is one of technologies that may have applications on social media. Numerous studies investigate recommendation systems by combining semantic web reasoning mechanisms. They exploit technologies such as ontologies to achieve intelligent recommendations. Users may then more easily use mobile devices to obtain digital content. We propose a novel method to personalize a recommendation system through the user's affecting index to exploit user similarity and familiarity between users.*

Keywords: Social network, Recommendation system, Ontology, Personal user index

1. **Introduction.** Social media has become one of the most popular web platforms, driven by the development of web 2.0. Web 2.0 encourages users to participate in content management, where users can insert, update, and delete their information through the Internet [1]. Data that users upload to social media are usually in the form of pictures [2], videos [3,4], likes, and comments. At present, users upload a basic profile, preferences, and much personal data to social media platforms. Social media platforms such as Facebook, Twitter, and Yelp may also be used for online shopping. A common problem which users face is information overload. Users often cannot find the information or items they desire. Though search engines such as Google can help users to find desired items or information, according to Chitika Research [5], users only use 33% of the top result from search engines, and only 18% from the second page. This shows that many resources are wasted because the search engine does not know the preferences of the user. Recommendation systems can address this problem. By applying user preferences and their Internet surfing habits, a recommendation system can enhance the accuracy of search engines results for users.

Recommendation systems can help users find items on the Internet based on their preferences. Many techniques for calculating the recommendation items for the recommendation system have been proposed. One approach is collaborative filtering, which determines the recommendation items based on similarity of ratings on the website. Il and Alexander proposed a recommendation system based on a combination of content based filtering, collaborative filtering, and across different domains to produce recommendations [6]. Parvatikar and Joshi proposed a recommendation system based on collaborative filtering and association rule mining [7]. Their research focuses on user interest in the items and similarity of interests, in a book recommendation system. This system only considers how users rate items on the Internet without considering how well users know each other. In general, this system treats all users as having similar background, culture, and preferences, a weakness of these systems.

Many studies have explored approaches for enhancing recommendation systems. Personal recommendation systems have been proposed to customize recommendation systems to user needs. Jiang et al. [8] proposed a tag method to recommend services on the Internet. They proposed a hybrid method using tag, time, and user social relationship information for service recommendations. María et al. [9] proposed a personal recommendation system based on web mining technology.

The main problem of traditional recommendation systems is the cold start user problem, in which a user has no user history on the platform. Traditional recommendation systems face serious problems recommending items for this type of user. Without a history of user ratings of items on the platform, collaborative filtering methods lack the information needed to build a comparison matrix to calculate the similarity between users. Collaborative filtering also faces issues with lack information datasets. Since social media data is usually unstructured or semi-structured data, constructing the comparison matrix requires large investments of computing resources.

In this paper, we propose an enhancement of our previous algorithm [10]. In our previous work, we proposed a novel method to recommend items to users based on the Acceptance Rate (AR) to calculate their similarity. Using AR we can classify users based on their preferences for items. We enhanced the system considering not only the user preferences, but also the relations between users. We assume that users' relations will affect their decisions. In previous research [10], it did not consider users' relations of familiarity factor. In this paper, we use the main concept of user familiarity to calculate social activity between users [11]. In addition, we justified through experiments that considering social relation factor we get better accuracy of recommendation items. Although the main formula is the same, we modify the calculation of social activity between users. We call this personal model in our recommendation system the Enhance Familiarity for Personal User Index (EFPUI). Our system is built based on ontology reasoning and uses an inference engine to extract the relation between users. This enables better accuracy for the recommendation system, especially for users with the cold start problem.

The remainders of the paper are organized as follows. Section 2 is related literature review. Section 3 is methodology description and Section 4 is experiment. Finally, we give conclusions and future works in Section 5.

2. **Literature Review.**

2.1. **Social network data.**

Social media data are commonly generated by the users themselves. Social media data usually take the form of posts, comments, and profiles [12]. These data are usually highly unstructured or semi structured [13]. Researchers have explored how to manage the unstructured data on social media by using ontology [14]. Social media data types are usually subjective, with many comments or posts based on the writer's perspective. Sometimes social media data have hidden meanings, such as likes on the posts of others, check-ins to a location, or watching a movie. In this case, no meaning may be identified without performing a mining process. For some companies this data does not have value, and they need to process it to extract information from the data.

2.2. **Crawler.**

One of the ways to collect data from a social media platform is to build a crawler engine to read data from social media users. A crawler is an engine that works with a list of URLs to visit, called the seeds. When the crawler reads and visits the seed URLs, it will collect another URL link from the page being read. The link collected is called the crawler frontier. URLs from the frontier will be visited recursively based on the policies of the crawler. In this research we use crawler data from Facebook users and collect information about the users and copy and save the information "on the fly". The collected data are saved as an archive that can be re-read and modified as the user modifies it, but are preserved as 'snapshots' [15].

Every change in the user data implies that the crawler needs to read the data again and modify the archive with the most recent data. This may occur because the system contains a scheduler engine that always detects Facebook user data, and is notified if there are changes made by the user. Because the crawler can only download using limited bandwidth and time, the changes must become a priority for the crawler. Users who have high similarity value with many other users will have the highest re-read rate from our crawler. Given that bandwidth is limited and reading bytes is not free, it is essential to manage the reading process of this crawler efficiently to maintain quality and freshness of data [16].

2.3. **Personal recommendation system.** Personal recommendation systems have been extensively researched in the recent decades. Recommendation systems have wide application in book [6,7], video [17], music [18], and other recommendation areas. Collaborative Filtering (CF) is a well known method for developing recommendation systems. CF works based on user ratings of items in websites. From the user rating, CF predicts user interest based on a comparison matrix with other users [19]. CF calculates a rating of the similarity between user interests to determine whether a given item will be recommended to other users. Researchers have offered many similarity algorithms, such as XOR Similarity [20], cosine similarity [21], improving missing value estimation on rough set theory [22], and singularity-based user similarity [23]. The problem of this method is that CF does not consider the relation between users. Users from different backgrounds may have different interests. CF also does not consider the age of the user, which in some domain recommendation systems, such as movies, is important because of age restrictions. CF also has difficulties with social media data which are unstructured or semi-structured. To construct a comparison matrix from these types of data requires large investments of resources and computation time [24].

In recent years, hybrid methods for recommendation systems have been studied. These combine techniques such as data mining, ontology, semantics, and clustering. Association rule mining is a technique used to search for relationships or hidden information inside transactions [25]. Using this capability, recommendation systems can predict what items should be recommended for users [26]. As with CF, this technique does not consider the relationships between users. Association rule mining only considers the relation between items, meaning that as long as two or more items occur together they can be recommended to the user, based on the values of the support and confidence thresholds. This generates items which do not match user wants. To pay greater attention to the user, an ontology based on user ontology is proposed. Since each user has their own ontology, the system presents more individualized user preferences. To recommend items between users, an inference engine is used to generate rules. Using these rules, the similarity rank between users may be calculated [25,26]. Hybrid methods such as semantic inference have been proposed to enhance the capability to construct user interest similarity.

Unlike previous approaches, the proposed model is a new method that combines similarity rank with user familiarity based on the user relation to other users. We consider closeness and degree of familiarity between users. Most importantly, recommendation items can be generated from cold start users when considering both user rating and relation to other users.

3. **Proposed Method.** This section describes the operation of our proposed method. We describe the architecture of our system, dynamic rule generation, and calculation of the similarity, familiarity, and similarity rank.

3.1. **Architecture of the system.** We develop a personal recommendation system. Our system collects data from Facebook through a crawler engine using the FacebookAPI. Figure 1 shows our system architecture. Figure 1 includes three main parts. One is data collection; the second is similarity calculation which includes the familiarity rate and similarity rate calculation and the last is recommendation results sorting and user's interface which will describe as follows.

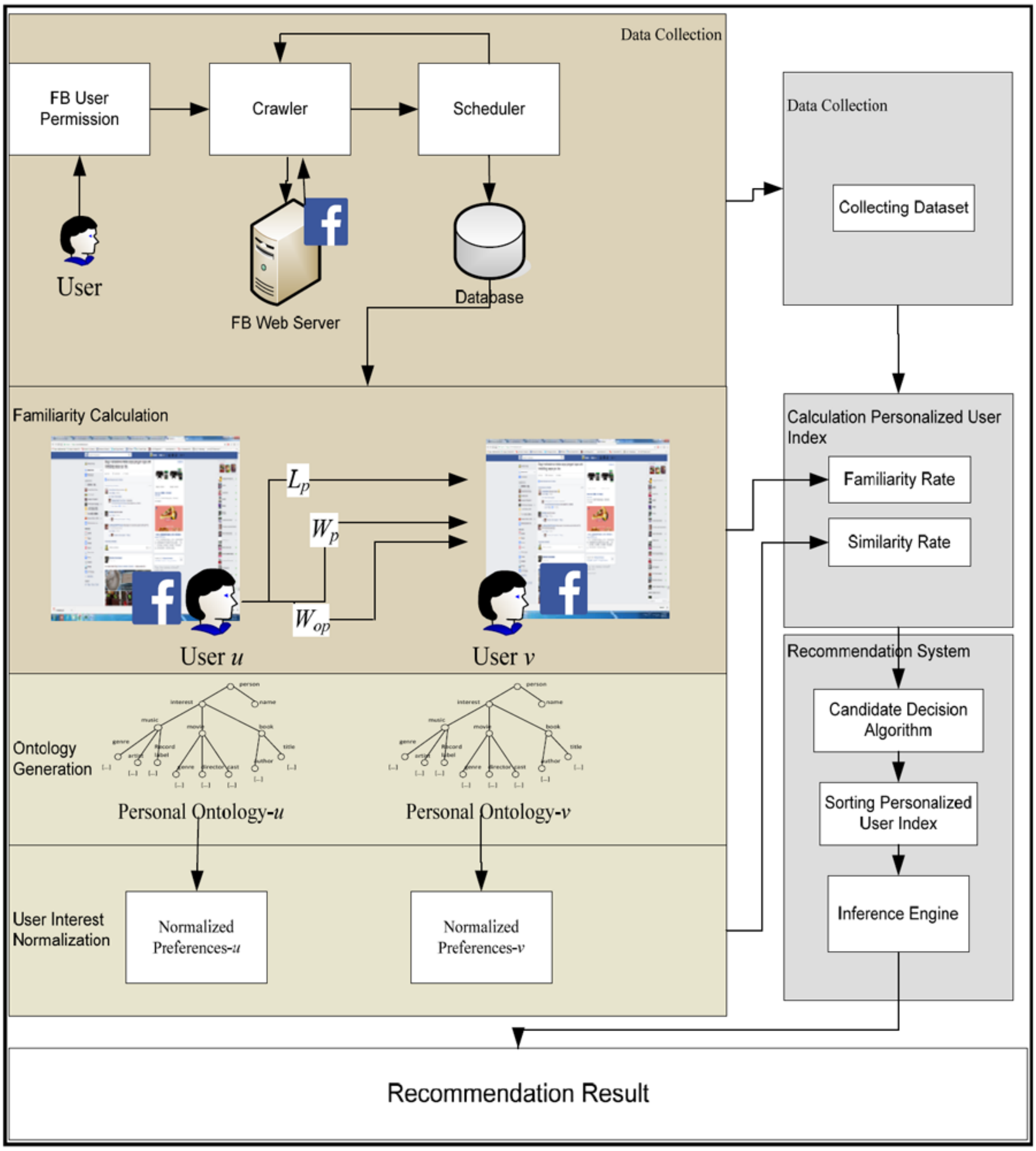

FIGURE 1. System architecture

1) Data from Facebook is protected by copyright and we could not access Facebook user data immediately. The first step is to obtain user permission in order to extract their data. Our system crawls their friend list, basic profile (name, gender, age, profile picture) and explicit preferences based on the movie, music, and book pages which the user follows. After the system collects user data from Facebook, it is saved in a web server database. In the web server database, the system processes the data and re-constructs the relationships.

2.A) The next step is divided into two broad processes. First, the system calculates the user similarity rate between two users. Next, it calculates the familiarity rate between the two users. The EFPUI is proposed based on a combination of these two parts. The similarity rate is the calculation of user preferences about item rating from users. Ontology is generated for each user for a more personalized description of the user. We propose ontology because ontology has the capability to handle unstructured or semi-structured social media data. Chen et al. stated that ontologies have many advantages, such as the ability to process information in open, distributed, heterogeneous, and weakly structured environments [20]. Common types of ontology include: (1) individuals: instances or objects; (2) classes: collections or concepts from types of objects; (3) attributes: aspects, properties, or parameters from types of objects; and (4) relations: ways in which classes and individuals can be related. There are several programs for editing ontologies such as Protégé and OntoEdit, but the process is tedious. Much research focuses on developing ontologies automatically by integrating them with knowledge acquisition and machine-learning technology. After a personal ontology for each user is generated, the system will process it to extract the user preferences. The user preferences are the user profile of what items users are interested in. For example, on a movie page we can classify user preferences by movie genres. The system counts the number of movies which users rate, comments on, or like for each genre. After the system calculates all user preferences, it normalizes the user preference values within the range of 0 to 1.

2.B) The second broad process of the system involves calculating the familiarity rate between users. The familiarity calculation is based on user social activity with other users on social media. The familiarity rate consists of three factors which are user posts, user likes, and user comments. L_p represents total like of user on other user's post, W_p represents weight of user post on other user's post, and W_{op} represents weight of other user post on user's wall. Our system familiarity rate works based on user posts, likes, and comments. For every post, like, or comment on another user's wall, the system will count and save the activity. This activity will later be compared with the relevant topics. A higher value for this social activity means that the user preference is likely to other user preferences. After the similarity and familiarity rates are calculated, the system will calculate the EFPUI value, which combines both the similarity and familiarity rate.

3) The acceptance rate is the threshold value that determines whether current user preferences will be classified as the same group (high similarity interest) or the opposite group. In this research we construct three different candidate groups of users. The first group of users has a high similarity of interest in the item to be recommended. The second has lower similarity of interest, and the last has no interest in the item. The system will sort EFPUI values based on descending order, where the highest EFPUI value will be at the top of the list. This means users with high EFPUI values are likely to be recommended items which are generated by the system. The inference engine then processes the user ontology to generate rules for the recommended item results.

3.2. **Problem formulation.** Symbol and notation used in this paper are shown in Table 1. We develop a system that personalizes the recommendation system based on user preferences and user familiarity, which recommends items for users based on their historical behavior such as posts, likes, and comments, and user preferences for items which they rate in the website.

Table 1. Symbols and descriptions

Symbol	Description	Symbol	Description
DB	Database	AR	Acceptance rate
FR	Familiarity rate	SR	Similarity rate
U	Set of users	u	A user in the set of users
P	Set of items	i	An item in the set of items
$UP_{m\times n}$	Matrix of user preferences	NP	Normalized user preferences
SR_Rank	Similarity Rank	r	Rating of items in the set of ratings
SA	Social Activity of user	W_p	Weight of user posts on user's wall
L_p	Likes of user posts	W_{op}	Weight of user posts on other user's walls
R	Set of ratings	$EFPUI$	Enhance Familiarity Personal User Index

Moreover, we calculate the recommended items for user u for an unknown item i based on the user's history and social activity on social media. In the DB system, we have a set of users and set of items $U = \{u_1, u_2, \ldots, u_m\}$. The ratings given by users expressed on items $P = \{i_1, i_2, \ldots, i_n\}$ are shown by matrix $UP = UP[u, i]_{m\times n}$. In this matrix $UP_{u,i}$ denotes the user preferences of user u for item i. For each user interested in an item 1 will be added to the value. Each $UP_{u,i}$ will be normalized using a normalization function, which generates $NP_{u,i}$. $NP_{u,i}$ is the normalization matrix which is generated from user preferences. Each user has a normalization matrix NP. This matrix is used to calculate the similarity rank SR_Rank between users. Each user u has set of friends' v and the closeness between user u and user v will be calculated as familiarity rate $FR_{u,v}$ between users. The FR calculation is based on the social activity SA between two users. SA consists of user post W_p, likes post L_p, and posts on other user's walls W_{op} about item i.

In this paper we propose the $EFPUI$ method to classify users into their candidate group of users. $EFPUI$ is calculated based on a combination of SR and FR. For item i between users u and v, we also treat their rates $r_{u,v}$ and $r_{u,v}$ as a subset of rating R. If both users u and v rate item i, then the value of $r_{u,v}$ will be 1 and their closeness can be shown by the $EFPUI$ value. Otherwise, if one of them has not rated the item, the value of $r_{u,v}$ will be 0 and the $EFPUI$ value will also be 0. A higher $EFPUI$ value between two users means that these users have similar interests. If items are recommended for one of them, the other user can also be recommended of the items. AR is a threshold value that determines whether the $EFPUI$ passes the minimum requirement. The system will rank the $EFPUI$ value in descending order and pick the higher $EFPUI$ value to be used as the recommendation item for other users.

3.3. **Calculation of similarity rate.** In this subsection we describe the calculation of the similarity rate. Every user u has user preferences for each item i. Matrix $UP_{u,i}$ will be generated from this process. The system will present a normalized user preference matrix $UP_{u,i}$ using a normalization function as shown in Equation (1).

$$NP_{u,i} = \begin{cases} 0, & Normalization(UP_{u,i}) < AR \\ 1, & Normalization(UP_{u,i}) \geq AR \end{cases} \tag{1}$$

AR is the threshold value from the normalization function. All users U will have their normalization user preference value NP for the set of items P. NPu and v are the normalization process values for users whose similarity we want to compare. A smaller SR value between users indicates that users have greater similarity and are closer to candidate user preferences. Equation (2) shows the process using the XOR equation.

$$SR_{u,v} = \sum_{i=1}^{P} XOR\left(NP_{ui}, NP_{vi}\right) \tag{2}$$

After the system calculates all the SR values between users, it will sort all the SR values and rank them from the closest to the farthest. Equation (3) shows the formula, where n is an integer value that stands for the number of users. The SR_Rank is the set of order of candidates recommended by the value of SR_{uv}.

$$SR\_Rank_i = \{SR_{uv,1}, SR_{uv,2}, \ldots, SR_{uv,n}\} \tag{3}$$

3.4. **Calculation of familiarity rate.** To handle the cold start user problem, in which the user has no record of their interests, we propose user familiarity. Familiarity consists of the calculation of comments, likes, and posts. Equation (4) shows the familiarity formula.

$$FR_{u,v} = \begin{cases} 1 + \log(SA_{u,v}) & Wp, Lp, Wop \in SA \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$SA_{u,v}$ is the social activity between users on a social media platform. In our formula we consider user u has social activity with user v if user u has at least one comment, like, or post on the other user's wall; otherwise, the familiarity value is 0. Social activity between users is average social activity for every item i in P. Equation (5) shows the formula of social activity.

$$SA_{u,v} = AVG\left(\sum_{i}^{P}\left(SA_{u,v}^{i}\right)\right) \tag{5}$$

$SA_{u,v}^i$ is social activity between users on a social media platform for item i. Thus, social activity is the average summation of all social activity for each item i which occur in P. Equation (6) shows calculation of social activity for item i.

$$SA_{u,v}^{i} = \frac{Wp_i + Lp_i + Wop_i}{Total\ Activity_i} \tag{6}$$

Wp_i is user posts for item i, Lp_i is user likes for item i and Wop_i is user posts on other user's walls for item i. $Total\ Activity_i$ is the summation of all activity of users on a social media platform for item i. It is interesting to give different weights for each factor between Wp_i, Lp_i, and Wop_i, since we know that user's habits are different and can cause differences in the way they communicate using social media. For this paper, we treat the weight for each factor as identical.

3.5. **Recommendation items generation.** Recommendation item generation begins with the calculation of *EFPUI*. *EFPUI* is constructed by combining the similarity rate and familiarity rate. Equation (7) shows the formula for *EFPUI* calculation.

$$EFPUI_{u,v} = \alpha * r_{u,v} SR_{u,v} + \beta * FR_{u,v} \tag{7}$$

where $r_{u,v}$ is item rate between user u and user v, SR is the similarity rate and FR is familiarity rate. α and β are parameter values we determine derived from experiments to give more weight to the similarity or familiarity rate function. The coefficient values will be determined later by the experimental results for each category. The system then sorts the *EFPUI* values from the highest rank. Equation (8) shows the formula for sorting the *EFPUI* values.

$$EFPUI\_Rank_i = \{EFPUI_{i,1}, EFPUI_{i,2}, \ldots, EFPUI_{i,p}\} \tag{8}$$

The dynamic rules are generated based on *EFPUI_Rank* of item i. The system will load the personal ontology of each user from *DB* and then choose the personal ontology of user based on their distance of similarity from *EFPUI_Rank*. The ontology and inference engine are static and cannot be changed. Because of this disadvantage of ontology, we propose modifying the algorithm to handle the ontology disadvantage by re-using knowledge in which data are changed.

For the cold start user problem, the ontology faces disadvantages since it can generate irrelevant rules because it has no previous knowledge which can be used by the system. Familiarity will help to handle this problem by looking for user relationships with other users. Thus, the system can select which user has a personal ontology for comparison. Our system is shown in Table 2 as simple algorithm of EFPUI.

Table 2. Algorithm for recommendation items

Algorithm of Enhance Familiarity Personal User Index (*EFPUI*)

Initialization: ***DB***
While (***ctr*** $<=$ ***DB***)
    Calculate ***SR*** (using Equation (2))
    Sort ***SR*** to obtain best ***SR_Rank*** (using Equation (3))
    Calculate ***FR*** (using Equation (4))
    Calulate ***EFPUI*** (using Equation (7))
    Sort ***EFPUI*** (Equation(8)) to find optimal user
    ctr++
End.

4. **Experiments.** We conduct our experiments using data from Facebook. The system collects data based on a crawler engine which collects data automatically. Our crawler engine works based on users from our DB. For efficiency our crawler is limited to collecting data only for user basic profiles, movie pages, music pages, and book pages. It is designed to save bandwidth and save time of data collection. Otherwise, the data collection process will be long and resource-demanding. Table 3 shows information about data the system collected from Facebook. We collect Facebook data from 130 users, and divide the data into a training dataset and a testing dataset. We use 100 users' data as our training dataset and the remaining 30 users as our testing dataset. Table 4 shows the data format of the dataset.

Table 3. Dataset collected from Facebook

User	**Movie**		**Music**		**Book**	
130	Total movies	261	Total music	887	Total books	90
	Total genres	83	Total genres	26	Total genres	22
	Total comments	8,921,437	Total comments	74,186,225	Total comments	3,361,022

TABLE 4. Data format of dataset

Tb_user_tree		Tb_movie	
Field	**Type**	**Field**	**Type**
Id	Varchar (20)	Id_movie	Varchar (50)
Name	Varchar (100)	Title	Varchar (50)
Parent_id	Varchar (20)	Genre	Varchar (100)
Level	Float	Release_date	Date
Weight	Float	Movie_artist	Varchar (100)
Quantity	Float	Studio	Varchar (50)
Node_weight (NW)	Float	Director	Varchar (50)
Node_quantity (NQ)	Float	Cover	Varchar (255)
NW_multiply_NQ	Float	Talking_about_count	Int (11)
Edge_quantity (EQ)	Float		
NW_multiply_EQ	Float		
Similarity_weight	Float		

4.1. **Performance measure.** We conducted the experiments using 100 users as the training data, and 30 users as the testing dataset. We collected all the movie, music, and book data from 100 users as the training data. The evaluation metric we used in our experiments is Mean Absolute Error (MAE), as this is a popular metric for measuring the accuracy of the recommendation system [28]. MAE is defined as follows:

$$MAE = \frac{\sum(m,i) \in M \left| R_{m,i} - \hat{R}_{m,i} \right|}{|M|} \tag{9}$$

where $R_{m,i}$ is a real item from user m on item i. $\hat{R}_{m,i}$ is the recommendation item generated by the system. M is the set of all user items in the test set.

4.2. **Evaluation.**

4.2.1. *Parameter setting.* This section focuses on the setting of the acceptance rate threshold value use for the parameters of our algorithm in our model. We have two parameters in our model, α and β. The sum of the two parameters is equal to 1. We set α from 0.1 to 0.9 and set β from 0.9 to 0.1. Later we compare the results for each category. We also compare the results without using familiarity (α value 0.1 to 0.9 and $\beta = 0$).

4.2.2. *Result comparison.* In this section, we compare the results from our experiments for each of our categories. We compare the results from first candidate list through the third candidate list. Table 5(a) lists the results for the movie category. The first candidate list has the best accuracy based on MAE, because the recommendations are generated using similar user interests. We compare the results without familiarity, where $\beta = 0$, and α values range from 0.1 to 0.9. The best result we obtain is $\alpha = 0.7$ and $\beta = 0.3$ with an MAE value of 4.65. Without the familiarity factor, the best result is $\alpha = 0.6$ with an MAE value of 4.98. It is clear that the familiarity factor positively affects the recommendation, but the other parameter value gives a less accurate result than the best parameter value without the familiarity factor.

We obtain the results for the music category shown in Table 5(b). The best result is for $\alpha = 0.3$ and $\beta = 0.7$, with an MAE value of 15.47. Without the familiarity factor, the best result is $\alpha = 0.6$ with an MAE value of 4.98. We obtained the same MAE value

Table 5. MAE comparison between threshold values for (a) the movie category, (b) the music category, (c) the book category

$\beta = 0$	C1 MAE	α β	C1 MAE
$\alpha = 0.1$	5.65	$\alpha = 0.1$ $\beta = 0.9$	5.58
$\alpha = 0.2$	4.99	$\alpha = 0.2$ $\beta = 0.8$	4.92
$\alpha = 0.3$	5.8	$\alpha = 0.3$ $\beta = 0.7$	5.70
$\alpha = 0.4$	5.88	$\alpha = 0.4$ $\beta = 0.6$	5.67
$\alpha = 0.5$	5.78	$\alpha = 0.5$ $\beta = 0.5$	5.09
$\boldsymbol{\alpha = 0.6}$	**4.98**	$\alpha = 0.6$ $\beta = 0.4$	4.85
$\alpha = 0.7$	5.4	$\boldsymbol{\alpha = 0.7}$ $\boldsymbol{\beta = 0.3}$	**4.65**
$\alpha = 0.8$	6.43	$\alpha = 0.8$ $\beta = 0.2$	5.88
$\alpha = 0.9$	5.85	$\alpha = 0.9$ $\beta = 0.1$	5.76

(a)

$\beta = 0$	C1 MAE	α β	C1 MAE
$\alpha = 0.1$	16.44	$\alpha = 0.1$ $\beta = 0.9$	15.87
$\alpha = 0.2$	16.44	$\alpha = 0.2$ $\beta = 0.8$	16.21
$\alpha = 0.3$	16.44	$\boldsymbol{\alpha = 0.3}$ $\boldsymbol{\beta = 0.7}$	**15.47**
$\alpha = 0.4$	16.44	$\alpha = 0.4$ $\beta = 0.6$	16.31
$\boldsymbol{\alpha = 0.5}$	**16.44**	$\alpha = 0.5$ $\beta = 0.5$	16.11
$\alpha = 0.6$	16.44	$\alpha = 0.6$ $\beta = 0.4$	16.17
$\alpha = 0.7$	16.44	$\alpha = 0.7$ $\beta = 0.3$	16.23
$\alpha = 0.8$	16.44	$\alpha = 0.8$ $\beta = 0.2$	15.79
$\alpha = 0.9$	16.44	$\alpha = 0.9$ $\beta = 0.1$	16.01

(b)

$\beta = 0$	C1 MAE	α β	C1 MAE
$\alpha = 0.1$	20.86	$\alpha = 0.1$ $\beta = 0.9$	20.58
$\alpha = 0.2$	20.87	$\alpha = 0.2$ $\beta = 0.8$	20.78
$\alpha = 0.3$	20.88	$\alpha = 0.3$ $\beta = 0.7$	20.13
$\alpha = 0.4$	20.88	$\alpha = 0.4$ $\beta = 0.6$	20.46
$\alpha = 0.5$	20.87	$\alpha = 0.5$ $\beta = 0.5$	20.14
$\alpha = 0.6$	20.88	$\alpha = 0.6$ $\beta = 0.4$	19.95
$\alpha = 0.7$	20.86	$\alpha = 0.7$ $\beta = 0.3$	20.74
$\alpha = 0.8$	20.87	$\alpha = 0.8$ $\beta = 0.2$	20.37
$\boldsymbol{\alpha = 0.9}$	**18.76**	$\boldsymbol{\alpha = 0.9}$ $\boldsymbol{\beta = 0.1}$	**17.88**

(c)

without the familiarity factor for the music category, but when we use the familiarity factor, it affects the results.

The results for the book category are shown in Table 5(c). We obtain the best results for $\alpha = 0.9$ and $\beta = 0.1$ with an MAE value of 17.88. Without the familiarity factor the best result of $\alpha = 0.9$ with an MAE value of 18.76.

4.3. **Discussion.** Comparing the results for each category, we discuss how to obtain the best threshold value α and β using our proposed method. Figures 2 and 3 show the comparisons for each category. We compare the initialization of the threshold value with

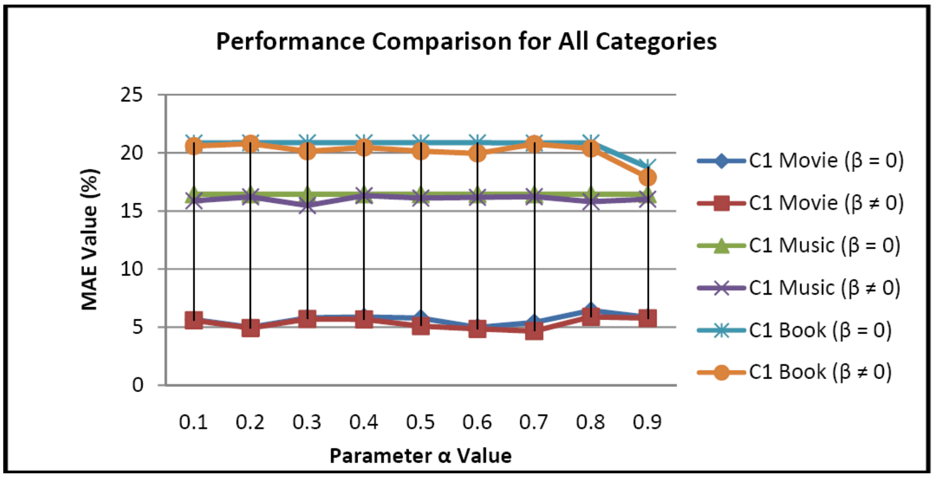

Figure 2. Comparison of performance for all categories

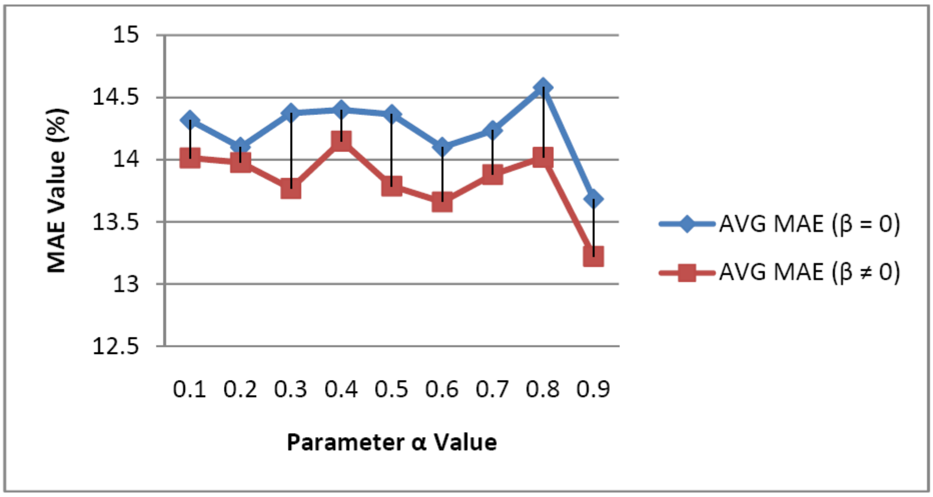

FIGURE 3. Comparison of performance with and without familiarity

the MAE value, and trade-off to determine the best value. We explain each category below.

4.3.1. *Comparison of each category.* Here we compare the results for the movie, music and book categories. We compare the similarity and familiarity factor in our recommendation system model. Figure 2 shows that the best accuracy is for the movie category, followed by the music and book categories. The familiarity factor affects the recommendation and gives better results than not considering the familiarity factor.

The book category shows that the familiarity factor has less of an effect than in the movie or music categories. The familiarity value $\alpha = 0.9$, $\beta = 0.1$ returns the best accuracy for this category. This means that the familiarity between users is less affecting factor for the book category recommendation. Thus, we can pay more attention to the similarity between users. For the movie category the best accuracy for recommendations is $\alpha = 0.7$, $\beta = 0.3$, showing that the familiarity factor affects the recommendation more strongly than for the book category. Thus, similarity and familiarity must be considered in the movie category. The familiarity factor has its strongest effect in the music category where $\alpha = 0.3$, $\beta = 0.7$. In the music category the familiarity between users has a significant effect on our recommendation system model. We also need to consider the similarity factor in the music category to obtain the best recommendation results. Overall the best accuracy is for the movie category with the smallest error value (MAE), while the worst accuracy is for the book category, which has the largest error value (MAE).

4.3.2. *Comparison of familiarity factor through recommendation.* Here we compare the results of our experiment with and without considering the familiarity factor. The system calculates the average from each category to form a single MAE value for each parameter. Figure 3 shows that using both the familiarity and similarity factor enables better results than using the similarity factor only. The best accuracy for recommendations is $\alpha = 0.9$ and $\beta = 0.1$ with an MAE value of 13.22.

4.3.3. *Comparison with other methods.* In this section, we compare our method with XOR similarity to provide recommendation. Figure 4 shows comparison of MAE value between proposed methods. Note that we do preprocess for movie dataset, and we do not do preprocess for music and book dataset. In other words music and book dataset will be sparser and contain more noise than movie dataset. Our propose methods (EFPUI)

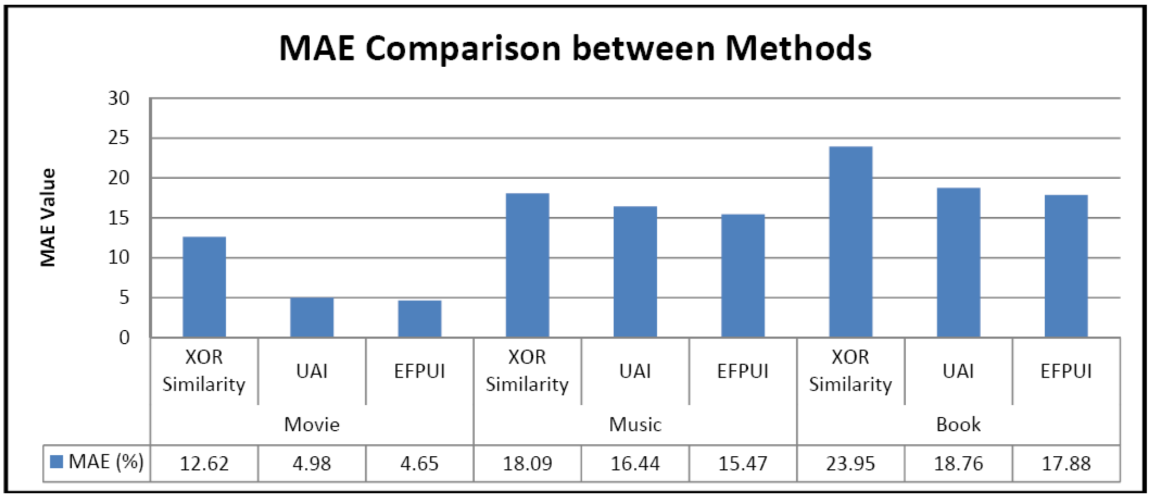

FIGURE 4. Comparison of performance with other methods

provide much better recommendation (reduce the MAE 7.97%) than XOR similarity algorithm and also outperform UAI algorithm by 0.33% in less sparse and less noise movie dataset. In music dataset which is sparser and contains more noise, our proposed method still provides better result although it is not significantly compared to movie dataset. For music dataset EFPUI outperforms XOR similarity algorithm by reducing MAE 2.62%, and also outperforms UAI algorithm by reducing MAE 0.98%. For book category EFPUI outperforms XOR similarity algorithm by reducing MAE 6.07% and also outperforms UAI algorithm by reducing MAE 0.88%.

5. **Conclusions.** This research proposed a personal recommendation system using the factors of similarity and familiarity. The system is developed to generate a candidate list based on the similarity algorithm, but we only study the first candidate list, the group of users who have the best similarity. The best accuracy of the recommendation list for the movie category is $\alpha = 0.7$ and $\beta = 0.3$, with an accuracy of 95.35% (MAE = 4.65%). For the music category the best accuracy is $\alpha = 0.3$ and $\beta = 0.7$, with an accuracy of 84.53% (MAE = 15.47%). For the book category the best accuracy is $\alpha = 0.9$ and $\beta = 0.1$ with an accuracy of 82.12% (MAE = 17.88%). For all categories the best accuracy is $\alpha = 0.9$ and $\beta = 0.1$ with an accuracy of 86.78% (MAE = 13.22%). We also find that using the familiarity factor gives better results for all parameters.

In the future, we will explore external factors such as the popularity of the item on social media. Popularity means how often the item is commented on or liked. We will explore whether this parameter has a significant effect on the recommendation system. Its accuracy will be considered as well.

Acknowledgment. This paper is supported by Ministry of Science and Technology, Taiwan, with project number: MOST103-2221-E-324-028 and MOST104-2221-E-324-019-MY2.

REFERENCES

[1] X. Qian, H. Feng, G. Zhao and T. Mei, Personalized recommendation combining user interest and social circle, *IEEE Trans. Knowledge and Data Engineering*, vol.26, no.7, pp.1763-1777, 2014.

[2] X. Qian, X. Liu, C. Zheng, Y. Du and X. Hou, Tagging photos using user's vocabularies, *Neurocomputing*, vol.111, pp.144-153, 2013.

[3] J. Shen and Z. Cheng, Personalized video similarity measure, *Multimedia System*, vol.17, no.5, pp.421-433, 2011.
[4] J. Shen, D. Tao and X. Li, Modality mixture projections for semantic video event detection, *IEEE Trans. Circuits and Systems for Video Technology*, vol.18, no.11, pp.1587-1596, 2008.
[5] *Chitika Research*, https://searchenginewatch.com/sew/study/2276184/no-1-position-in-google-gets-33-of-search-traffic-study, 2013.
[6] I. Il and H. Alexander, Does a one-size recommendation system fit all? The effectiveness of collaborative filtering based recommendation systems across different domains and search modes, *ACM Trans. Information System*, vol.26, no.1, 2007.
[7] S. Parvatikar and B. Joshi, Online book recommendation system by using collaborative filtering and association mining, *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, pp.1-4, 2015.
[8] Z. Jiang, A. Zhou, S. Wang, Q. Sun, R. Lin and F. Yang, Personalized service recommendation for collaborative tagging systems with social relations and temporal influences, *IEEE International Conference on Services Computing (SCC)*, San Francisco, CA, USA, pp.786-789, 2016.
[9] N. M. María, S. Saddys, F. L. Vivian, D. M. María and L. S. Ángel, Web mining based framework for solving usual problems in recommender systems a case study for movies' recommendation, *Neurocomputing*, vol.176, pp.72-80, 2016.
[10] R. C. Chen, Hendry and C. Y. Huang, A domain ontology in social networks for identifying user interest for personalized recommendations, *Journal of Universal Computer Science*, vol.22, no.3, pp.319-339, 2016.
[11] D. Ahn, T. Kim, S. J. Hyun and D. Lee, Inferring user interest using familiarity and topic similarity with social neighbors in facebook, *Proc. of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pp.196-200, 2012.
[12] D. Zeng, H. Chen, R. Lusch and S. H. Li, Social media analytics and intelligence, *IEEE Intelligent Systems*, vol.25, no.6, pp.13-16, 2010.
[13] H. C. Chen, R. H. L. Chiang and V. C. Storey, Business intelligence and analytics: From big data to big impact, *MIS Quarterly*, vol.36, no.4, pp.1165-1188, 2012.
[14] L. Kaijian and E. Nora, Ontology-based sequence labelling for automated information extraction for supporting bridge data analytics, *Procedia Engineering*, vol.145, pp.504-510, 2016.
[15] J. Masanès, *Web Archiving*, Springer, 2007.
[16] J. Edwards, K. S. McCurley and J. A. Tomlin, An adaptive model for optimizing performance of an incremental web crawler, *Proc. of the 10th Conference on World Wide Web (Hong Kong: Elsevier Science)*, pp.106-113, 2001.
[17] Q. Huang, B. Chen, J. Wang and T. Mei, Personalized video recommendation through graph propagation, *ACM Trans. Multimedia Computing Communication Application*, vol.10, no.4, 2014.
[18] J. Shen, H. Pang, M. Wang and S. Yan, Modeling concept dynamics for large scale music search, *Proc. of the 35th Int. ACM SIGIR Conference Research Development Information Retrieval*, New York, NY, USA, pp.455-464, 2012.
[19] Z. Choiru, P. Mahardhika, L. Edwin and G. A. Sreenatha, Evolving type-2 web news mining, *Applied Soft Computing*, vol.54, pp.200-220, 2017.
[20] R. C. Chen, H. Y. Shih, Y. S. Lin and Hendry, Video recommendation system based on personalized ontology and social network, *Proc. of the ASE BigData & Social Informatics 2015 (ASE BD&SI'15)*, ACM, New York, NY, USA, 2015.
[21] D. Wang, H. Lu and C. Bo, Visual tracking via weighted local cosine similarity, *IEEE Trans. Cybernetics*, vol.45, no.9, pp.1838-1850, 2015.
[22] B.-W. Wang and V. S. Tseng, Improving missing-value estimation in microarray data with collaborative filtering based on rough-set theory, *International Journal of Innovative Computing, Information and Control*, vol.8, no.3(B), pp.2157-2172, 2012.
[23] S. Liang, L. Ma and F. Yuan, A singularity-based user similarity measure for recommender systems, *International Journal of Innovative Computing, Information and Control*, vol.11, no.5, pp.1629-1638, 2015.
[24] I. Barjasteh, R. Forsati, D. Ross, A. H. Esfahanian and H. Radha, Cold-start recommendation with provable guarantees: A decoupled approach, *IEEE Trans. Knowledge and Data Engineering*, vol.28, no.6, pp.1462-1474, 2016.
[25] B. Vo, T.-P. Hong and B. Le, Mining most generalization association rules based on frequent closed itemset, *International Journal of Innovative Computing, Information and Control*, vol.8, no.10(B), pp.7117-7132, 2012.

[26] J. Chen, C. Miller and G. G. Dagher, Product recommendation system for small online retailers using association rules mining, *Proc. of the 2014 International Conference on Innovative Design and Manufacturing (ICIDM)*, Montreal, QC, pp.71-77, 2014.
[27] T. T. Quan, S. C. Hui, A. C. M. Fong and T. H. Cao, Automatic fuzzy ontology generation for semantic web, *IEEE Trans. Knowledge and Data Engineering*, vol.18, no.6, pp.842-856, 2006.
[28] M. Jiang, Social contextual recommendation, *Proc. of the 21st ACM International CIKM*, New York, USA, pp.45-54, 2012.