JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 34, 1273-1286 (2018)
DOI: 10.6688/JISE.201809_34(5).0010

A Reqirements Engineering Approach to Attribute Selection in a Recommendation System

PEI-CHUAN ZHOU, ALAN LIU AND CHING-JUNG LEE
Department of Electronic Engineering
National Chung Cheng University
Chiayi, 621 Taiwan
E-mail: z09620@hotmail.com; aliu@ee.ccu.edu.tw; kgarnett521@hotmail.com

Attributes play an important role in recommendation systems. This paper presents a method to enhance the result of a recommendation system through attribute selection based on user goals. With a limited amount of data, a recommendation system suffers from sparsity problems and cold start problems in collaborative filtering. In order to utilize such data, our method seek to find a set of suitable attributes using goal-driven requirements analysis to provide more timely information of user preferences for the current needs of a user. The process of goal-driven analysis is aided by the Analytic Hierarchy Process in attribute selection. The experiments show that this method is feasible and derives promising results.

Keywords: attributes selection, goal-driven use case analysis, analytic hierarchy process, user preferences, recommendation system

1. INTRODUCTION

Research in recommendation systems has been active for the last two decades, solving information overloading problems to assist users in filtering unnecessary information [1, 2]. Applications of recommendation systems appear in social networks, online shops, search engines, and service promoting products. In a recommendation system, attributes of products and customers are used in calculating the similarity among products and among customers.

Collaborative filtering is used for recommender systems. Taking a product recommender as an example, collaborative filtering techniques use the past ratings of a current customer to find a group of similar customers in order to give recommendations of products that the current customer has not encountered in the past. Despite their usefulness in a wide range of applications, collaborative filtering methods still suffer from sparsity problems and cold start problems. As an example of a sparsity problem, if a product lacks ratings from other customers, then the recommendation of that product may not be given [3]. Moreover, as an example of cold start problem, if the user is a new customer, finding similar customers becomes a problem since there are no previous ratings to compare with the user [4, 5].

A large amount of data is required in a recommending system, and computing all combination of attributes is infeasible. Thus, using a set of suitable attributes becomes important. However, attributes are usually chosen using expert heuristics, and the dynamical change in user preferences is not considered. Instead of solving the sparsity

Received May 15, 2017; revised May 16, 2017; accepted April 16, 2017.
Communicated by Ren-Hung Hwang.

problems or cold start problems by changing the algorithms of similarity measures or categorization, we focus on utilizing the attributes in hand to enhance the result of a recommendation. The contribution of this paper to the recommendation system research field is to leverage the extra information gather through requirements engineering. By following the concept analysis of user preference [6], the idea is to identify user attributes from the user requirements on the service provided by a recommendation system. By having a set of attributes more related to user preferences, extra information can be supplied in order to gain better recommendation within the limit of current techniques in collaborative filtering.

The rest of the paper is organized as follows. The next section states the motivation of this work and investigates the cold start problem, the sparsity problem, and the evaluation method. Our proposed method is presented in Section 3, and the fourth section provides the results of our experiments. Finally, the conclusion is given in the last section.

2. MOTIVATION

The main challenge is brought by the lack of data introduces the problem of finding no similar users for new users. For a new product, it may not be selected if previous ratings are the only feature used in selection. Collaborative filtering relies heavily on user ratings, and the system is not able to compute similarity effectively. In order to address these problems, the hybrid recommendation techniques have been proposed. Hybrid recommendations combine two or more different techniques to gain better efficiency and decrease weaknesses [7]. Most commonly, collaborative filtering is combined with some other technique in an attempt to exploit benefit of each one of these techniques, such as demographic filtering [8], content-based filtering [9] and utility-based filtering [10].

These recommenders provide recommendations via referring to user profiles and item profiles. A user profile is a collection of individual characteristics information associated with a specific user. The information includes demographic information, *e.g.*, age, gender, education, occupation, *etc.*, and represents the user's interests or preferences. Item profiles can be used to record the description of the features of each item; for example, in a movie recommender system, each movie can be represented by actors, director, genre, year of release, *etc.* The recommendation systems give different results based on models of using different features/attributes. Therefore, selecting critical attributes is considered an important step in the successful recommender system [11, 12].

In the process of attribute selection, the prediction is enhanced by taking into consideration user context information [13]. According to the situation analysis, the system can obtain the user's current needs, and then make recommendations. The concept is an application of knowledge-based recommender system [14]. It builds a knowledge base with a model of the users and items in order to apply inference techniques and find matches between need of users and features of items. In our system, we exploit a knowledge-based approach, named GDUC [15], for building recommendation criteria. A detailed discussion of GDUC will appear in the next section.

As for evaluating the recommendation result, the Mean Absolute Error (MAE) is commonly used [16]. By comparing the recommendation with the actual choice of the user, the recommendation is evaluated. The formula in Eq. (1) is given where n is the number of product to predict, p_s the rating of product calculated by the system, r_s the actual rating by the user on the product s, and e_s the difference in ratings.

$$\mathrm{MAE} = \frac{1}{n}\sum_{s=1}^{n}|p_s - r_s| = \frac{1}{n}\sum_{s=1}^{n}|e_s| \tag{1}$$

Looking at Eq. (1), the simple goal is to make the value p_s as close to r_s as possible. The techniques in improving the value of p_s are discussed in the next section.

With the above description of the issues associated with a recommendation system, we consider attribute selection as a requirements engineering problem. The product recommended is something to satisfy user requirements. The user requirements are derived from the user preferences and the context when the user request is made. We can also consider the goal of the user request is to reach that product which satisfy the user request. Thus, the purpose of our approach is not to solve the sparsity problems or the cold start problems by altering the algorithms on similarity measures or categorization, but to provide a set of suitable attributes to enhance the recommendation.

3. PROPOSED METHOD

3.1 Problem Description

The main purpose of our study is to find a set of suitable attributes to be used in a recommendation system. Thus, our recommendation system is to add a method of attribute selection into a typical approach in recommendation systems. The existing approach in recommendation system [17] follows these phases:

(1) Constructing a user rating matrix
(2) Constructing a user similarity matrix
(3) Applying the nearest neighbor method
(4) Producing prediction/recommendation

The main idea behind the above approach is that the item which you may be interested in is recommended through finding persons with high similarity values to you. However, similarity measures depend on good choice of attributes in categorizing users. Based on this reason, we consider preferences of individual users to be used as attributes in similarity measure. Furthermore, user preferences may change at times, so context needs to be considered.

Looking for personal preferences, we have employed a requirements engineering method in the past [18]. Our approach exploits the Goal-Driven Use-Case (GDUC) analysis method [15] in deriving user preferences. It consists of the following four steps in similarity measures, in which the steps 2 and 3 are performed repeatedly.

(1) Identify the user
(2) Identify the goals
(3) Construct the use case model
(4) Evaluate the goals

In order to identify what attributes affect the user goals the most, we apply pairwise analysis with Analytic Hierarchy Process (AHP) to decide the priorities of the attributes which will be used in the recommendation system to improve the performance. Fig. 1 summarizes the steps, where the blue box indicates steps of GDUC method and the yellow box is steps of AHP.

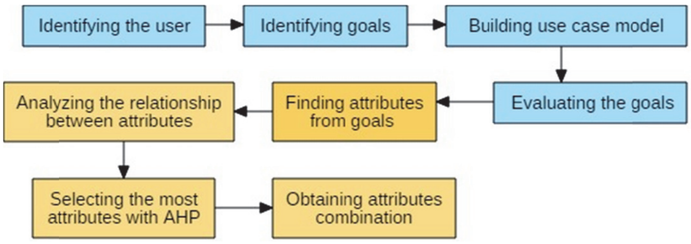

Fig. 1. Steps involved in attribute selection.

Fig. 2 shows our recommended system flow. There are five phases, where the attribute is selected by the GUDC method and AHP before constructing a user similarity matrix. According to key attributes which are selected in a context, we look for the most similar users. This helps recommendation effect.

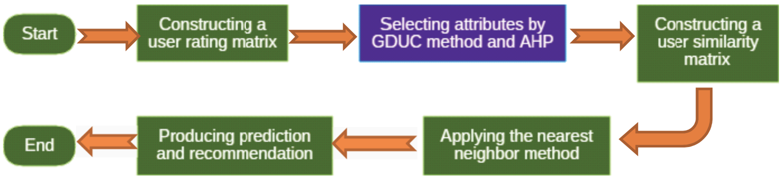

Fig. 2. Flowchart of our recommendation system.

3.2 Dataset

We have evaluated the performance of our methods on two different real datasets, namely MovieLens [19] and LDOS-CoMoDa [20]. MovieLens is a popular dataset used by researchers in the study of recommendation systems. The dataset consists of ratings on movies, and the size of data is large with many user ratings. On the other hand, LDOS-CoMoDa contains less data, but have different user features and contextual factors. Both datasets will be presented to demonstrate the effectiveness of our methods in different situations.

3.3 Model Construction

The dataset, u1.base, is provided by MovieLens, and the following description of model construction uses u1.base as illustration. The user-item matrix can be created with attributes like the user ID, item ID, and associated ratings, and a sample matrix is shown in Table 1. It consists of 100,000 ratings from 943 different users on 1,682 movie items The ratings are given in the range of 1 to 5, and the value 0 indicates that an item has not been rated by the associated user.

Table 1. U1 user-item matrix (u1.uim).

Item id / User id	1	2	...	1681	1682
1	5	3	…	0	0
2	4	0	…	0	0
…	…	…	…	…	…
942	0	0	…	0	0
943	0	5	…	0	0

After deriving the user-item matrix, the similarity among users is calculated. We use the Pearson's correlation coefficient method in obtaining the similarity. Based on [21], the Pearson correlation coefficient is used to define the similarity of two users i and j based on their ratings for common items, as shown in Eq. (2), where s is the item to review, $S_{i,j}$ the common items reviewed by both users i and j, $r_{i,s}$ the rating of the item s given by the user i, and $\overline{r}_i$ the average rating is given by the user i.

$$pcc\_sim(i,j) = \frac{\sum_{s \in S_{i,j}} \left(r_{i,s} - \overline{r_i}\right)\left(r_{j,s} - \overline{r_j}\right)}{\sqrt{\sum_{s \in S_{i,j}} \left(r_{i,s} - \overline{r_i}\right)^2 \sum_{s \in S_{i,j}} \left(r_{j,s} - \overline{r_j}\right)^2}} \tag{2}$$

Table 2 shows the result of processing the similarity between two users. The users in the column are compared to the users in the row. The related values are in the range of −1 to 1. The diagonal shows the total similarity between the same person, which obviously has all 1's, and the upper triangle contains the similarity values in positive and negative values. The lower triangle would be equivalent to the upper triangle, as represented by this symbol "--", so it does not have to calculate. The value *nan* indicates that there are no items that the corresponding pair of users, causing the denominator of the formula to be 0.

Table 2. U1 person's coefficient method in obtaining the similarity between users (u1.pcc_table).

User id	1	2	3	...	942	943
1	1	0.5218	0.525129	...	−0.1778	0.2964
2	--	1	0.14	...	0.2497	0.7124
3	--	--	1	...	0.5133	*nan*
…	…	…	…	…	…	…
942	--	--	--	--	1	−0.3453
943	--	--	--	--	--	1

After obtaining a similarity matrix like Table 2, we use the k-nearest neighbors (k-NN) algorithm to process the similarity among users. For choosing values like 5, 10, 20, 30, *etc.* for k, we can observe the differences in recommendation content. We try to provide more information obtained from user requirements through user attribute analysis in order to ease the problem caused by sparsity.

3.3 User Attributes Analysis

Following the four steps of the GDUC analysis method mentioned above, we need to identify and define a user in order to obtain the status and needs of the user. A use case model is constructed as a base for further analysis. Since a typical recommendation system approach uses a method like the *k*-NN algorithm to process the similarity based on user ratings, we are trying to discover the attributes more fitting to the similarity of the users, which reflect the needs of the users at a particular moment. The status of a user is associated with extended information about the user. Such information includes not only the age and the gender of the user, but also the time of day, the day in the week, the location, the weather of the day, the mood/emotion of the user. The status attributes are summarized in Table 3.

Table 3. User status and information.

User status	Information Categories
age	6-17,18-30, 31-50, > 50 years old
gender	male, female
time	morning (5-12), afternoon (12-18), evening (18:-23), night (23-5)
daytype	weekdays and weekend
location	private bedroom, public room, outside, away from home
weather	sunny, cloudy, rainy
mood	happy, upset, bored *etc*.

A faceted classification scheme in GDUC analysis aides in identifying goals and checking the integrity. A goal is classified with three facets: Competence, View, and Content. For competence, we can see that requirements are either completely satisfied or partially satisfied. This can also be seen as fulfilling a rigid goal or a soft goal. The view facet leads the analyst to see if the requirements are viewed as actor-specific or system-specific. The content facet identifies the goals with functional requirements or nonfunctional requirements.

Using the GDUC approach, the information in Table 3 can be analyzed and associated with goals. After affirming the original goals, the next step is to construct the use case model. The original use case, "Input the user status" has a goal which needs to satisfy system requirements, and this goal is a rigid (R), actor-specific (A), and functional (F) goal. Other goals are soft (S), actor-specific (A), and nonfunctional (N) goals.

- Input the user status: Satisfy system requirements (R, A, F)
- Age: Basic information of age (S, A, N)
- Gender: Gender differences in information (S, A, N)
- Time: Time of day (S, A, N)
- Day type: weekday, weekend, working day, or holiday (S, A, N)
- Location: location specific information (S, A, N)
- Weather: Weather information (S, A, N)
- Mood: Personal emotional information (S, A, N)

The resulting use case model is depicted in Fig. 3. With eight use cases (numbered U1 ~ U8) and eight goals (numbered G1 ~ G8) to analyze, where all use cases inherit use case of "input user status". These are summarized in Table 4. It indicates that the more specific model element shares the same specification with. We can find out the relevant goal by deducing each user case since these are 1-1 relation. We assume that the system can be calculated based on user input and other techniques to know the status. After getting the status, the system begins to identify the goal that satisfy system requirement.

After identifying the goals, the last procedure is to evaluate the goals. There are three concepts to process in this procedures: the relationship between use cases and goals, the interaction among the goals, and the interaction among system goals. In our movie recommendation, these attributes are obtained according to our goals. After analysis, we have derived the attributes shown in Table 5.

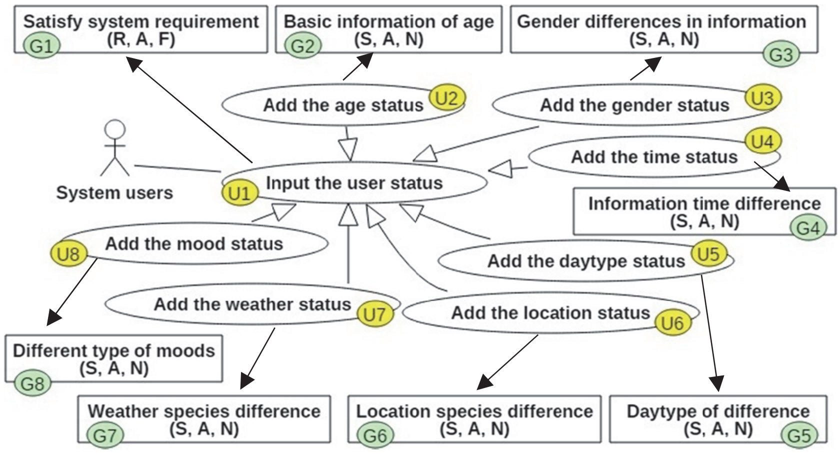

Fig. 3. The result of use-case model using GDUC approach.

Table 4. Use cases and goals.

Use case		Goal	
U1	input user status	G1	satisfy system requirement
U2	add age status	G2	basic information of age
U3	add gender status	G3	gender differences in information
U4	add time status	G4	information of time difference
U5	add daytype status	G5	difference between days of the week
U6	add location status	G6	location specific difference
U7	add weather status	G7	weather specific difference
U8	add mood status	G8	different types of moods

Table 5. Goals and attributes.

Goal		Find attributes
G1	satisfy system requirement	personal information, environmental factor, current situation
G2	basic information of age	personal information
G3	gender differences in information	personal information
G4	information of time difference	environmental factor
G5	difference between days of a week	current situation
G6	location specific difference	current situation
G7	weather specific difference	environmental factor
G8	different types of moods	current situation

3.4 Attribute Selection

The most suitable attributes need to be identified among the attributes derived. For making such decision, we rely on a pairwise comparison method based on AHP. First we can see how much the environment factors (EF) and the current situation (CS) affect the weighting of the personal information (PI) in Eq. (3). As an illustration, assume a person cares more of the current situation and thinks about the personal information the least, we could give the scores of 7 and 3, respectively. As for the environment factors, a neutral score 5 is given. The weighting values is obtained by using Saaty's criterion [22]. By following the process in the AHP, we will have a calculation as below.

$$\begin{array}{c} \\ PI \\ EF \\ CS \end{array} \begin{array}{ccc} \text{Personal information} & \text{Environmental factors} & \text{Current situation} \\ \left[\begin{array}{c} 3/3 \\ 5/3 \\ 7/3 \end{array}\right. & \begin{array}{c} 3/5 \\ 5/5 \\ 7/5 \end{array} & \left.\begin{array}{c} 3/7 \\ 5/7 \\ 7/7 \end{array}\right] \end{array}$$

$$\begin{array}{c} PI \\ EF \\ CS \end{array} \begin{bmatrix} \left(\frac{3}{3} \times \frac{3}{5} \times \frac{3}{7}\right)^{1/3} \\ \left(\frac{5}{3} \times \frac{5}{5} \times \frac{5}{7}\right)^{1/3} \\ \left(\frac{7}{3} \times \frac{7}{5} \times \frac{7}{7}\right)^{1/3} \end{bmatrix} = \underset{Sum = 3.18}{\begin{bmatrix} 0.64 \\ 1.06 \\ 1.48 \end{bmatrix}} \rightarrow \underset{Sum = 1}{\begin{bmatrix} 0.2 \\ 0.33 \\ 0.47 \end{bmatrix}} \tag{3}$$

The produced weights concerning these three attributes are 0.2 for PI, 0.33 EF, and 0.47 CS. With these weights in hand, three plans are proposed to see how the attributes can be used. The plan A uses a single attribute of PI which is what a typical recommendation system uses. The plan B excludes PI and combines EF and CS. The plan C adds CS to PI. Table 6 shows the scores assigned to this concept.

Table 6. Attribute scores for each plan.

Plans	Personal information	Environmental factor	Current situation
Plan A	7	1	1
Plan B	1	5	7
Plan C	3	1	5

Based on these score assignments, we see how each of these attributes weighs in the view of each plan. For PI, Plan A values it as 7 but Plan B has 1 while Plan C treats it 3. The following steps derives the weights of PI in each plan, as show in Eq. (4).

$$\begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 1 & \frac{7}{1} & \frac{7}{3} \\ \frac{1}{7} & 1 & \frac{1}{3} \\ \frac{3}{7} & \frac{3}{1} & 1 \end{bmatrix} \end{array} \rightarrow \begin{bmatrix} \left(1 \times \frac{7}{1} \times \frac{7}{3}\right)^{1/3} \\ \left(\frac{1}{7} \times 1 \times \frac{1}{3}\right)^{1/3} \\ \left(\frac{3}{7} \times \frac{3}{1} \times 1\right)^{1/3} \end{bmatrix} = \underset{Sum = 3.99}{\begin{bmatrix} 2.54 \\ 0.36 \\ 1.09 \end{bmatrix}} \rightarrow \underset{Sum = 1}{\begin{bmatrix} 0.63 \\ 0.1 \\ 0.27 \end{bmatrix}} \tag{4}$$

After repeating the similar calculation above to EF and CS, then a matrix containing three plans (A, B, and C) having different weights on each attributes is derived as below.

$$\begin{array}{cc} & \begin{array}{ccc}\text{Personal information} & \text{Environmental factors} & \text{Current situation}\end{array} \\ \begin{array}{c}A\\B\\C\end{array} & \begin{bmatrix} 0.63 & 0.14 & 0.05 \\ 0.1 & 0.71 & 0.35 \\ 0.27 & 0.14 & 0.59 \end{bmatrix}\end{array}$$

The weights above are applied to the original weights (0.2 for PI, 0.33 EF, and 0.47 CS). The plan B looks most promising and then is chosen based on the calculation of the total weights on the plans calculated as below:

- Plan A: (0.64)∗(0.2)+(0.14)∗(0.33)+(0.05)∗(0.47)=0.1957.
- Plan B: (0.1)∗(0.2)+(0.71)∗(0.33)+(0.35)∗(0.47)=0.4188.
- Plan C: (0.27)∗(0.2)+(0.14)∗(0.33)+(0.59)∗(0.47)=0.3775.

4. EXPERIMENTS

Typically, the majority part of a dataset is used in training the system and the rest of the dataset is used to evaluate how accurate the prediction is. The MAE is a popular way to check the error margin. The smaller the value, the better the performance. MovieLens and LDOS-CoMoDa are two datasets used in our experiments. We use 80% for training and 20% for evaluation.

Two experiments are presented in this section. The first experiment indicates that providing suitable attributes will enhance the performance of the recommendation of our original approach with *k*-NN applying to MovieLens. The goal of the second experiment is to show that although the pure collaborative filtering approach performed poorly for the LDOS-CoMoDa dataset, it was improved after aiding with user attributes. Fig. 4 shows the process of how attributes are obtained and provided.

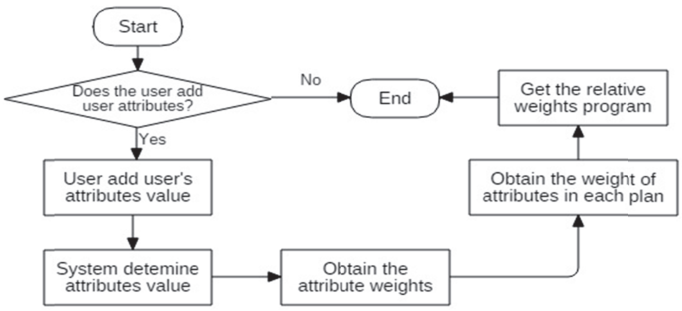

Fig. 4. Obtain and provide attributes.

4.1 Experiment 1

Scenario: The user was distressed after coming back from work. She said the tired mood message and asked the system to recommend a movie and input the. The information gathered from the system is that the user is a female and is 25 years old. It is a working day and the mood is negative. As for time, location, and weather, these are irrelevant at the moment.

As the scenario's illustration, assume the user cares more of the current situation (CS). We give the score of 7 to CS. The personal information (PI) is also important but not the highest, so we give a neutral score of 5. As for the environment factors (EF), it doesn't matter, so we give the lowest score of 1. After analyzing with the AHP, the weights are 0.3846 for personal information, 0.0769 for environment factors, and 0.5384 for current situation. Three plans are proposed. The plan A uses a single attribute which is EF since it is the highest. The plan B combines PI and CS, the two highest. The plan C consists of PI and EF. Table 7 shows the scores assigned to this concept.

Table 7. Attribute scores for each plan.

Plans	Personal information	Environmental factor	Current situation
Plan A	1	7	1
Plan B	5	1	7
Plan C	4	7	1

After calculating the weights of each attributes in each plan, the results are summarized below.

$$\begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{ccc} \text{Personal information} & \text{Environmental factors} & \text{Current situation} \\ \left[\begin{array}{ccc} 0.1 & 0.47 & 0.1 \\ 0.5 & 0.07 & 0.8 \\ 0.4 & 0.47 & 0.1 \end{array}\right. & & \left.\right] \end{array}$$

- Plan A: 0.1∗0.3846+0.47∗0.0769+0.1∗0.5384=0.128
- Plan B: 0.5∗0.3846+0.07∗0.0769+0.8∗0.5384=0.628
- Plan C: 0.4∗0.3846+0.47∗0.0769+0.1∗0.5384=0.244

After executing the recommendation system using three plans separately, we obtained the results shown in Fig. 5, where the MAE for each of three plans are 0.9508, 0.8405, and 0.8714, respectively. By applying the AHP process as shown above, the values obtained by those three plans are 0.128, 0.628, and 0.244. This indicates that the plan B is a better choice, which confirms the experiments.

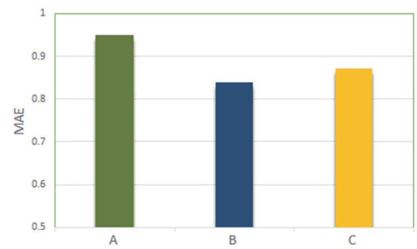

Fig. 5. MAE of different plans.

4.2 Experiment 2

The proposed method uses the Pearson Correlation Coefficient approach in finding the similarity between users, and the performance is affected severely by the sparse matrix shown previously. The LDOS-CoMoDa dataset contains fewer data and the matrix is sparse, which leads to poor performance. The LDOS-CoMoDa dataset contains different kinds of attributes compared to the MovieLens dataset, so we could experiment adding different attributes that are not available in the MovieLens dataset. Table 8 shows the effect of aiding attributes to the system at the settings of different *k* values used for the *k*-NN algorithm for classification. Fig. 6 shows the effect of adding attributes in finding similar users, in which there is a significant improvement made when $k = 20$, but the reverse effect happens at $k = 40$. The results are slightly better for most of the *k* values except for $k = 30$ and $k = 40$. The poorer performance is affected by the model at $k = 20$. That is the over-fitting phenomenon. Each user gives a rating for only few items, so it can produce unreliable similar users. Table 9 shows the *t*-test for adding attributes. The resulting *t*-statistic is statistically significant at the 0.05 level for a one-tail test. It means there is a significant effect of considering user attributes.

Table 8. Change in recommendation due to considering user attributes.

k value	MAE	
	Without user attributes	Considering user attributes
5	0.6851	0.6704
10	0.6823	0.6618
20	0.7349	0.6559
30	0.8645	0.8712
40	0.8428	0.8587
50	0.9053	0.8785
60	0.9110	0.8991
70	0.9425	0.9207

Table 9. The paired *t*-test for Experiment 2.

Standard Error	Degree of freedom	Confidence level	Calculated *T*-Statistic	*t* Critical value, 0.05, one-tail	*P* (*T*<=*t*) one-tail
0.035460763	8–1=7	95%	1.898	1.894	0.0497

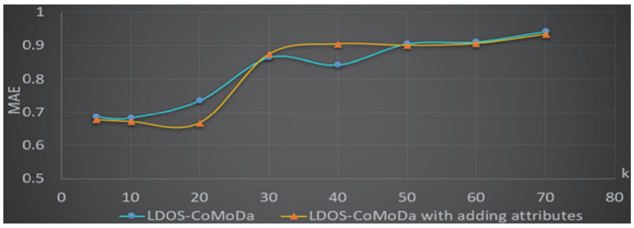

Fig. 6. Compare original dataset to adding attributes.

5. CONCLUSIONS

Even with the presence of the sparsity problem and the cold start problem, choosing a set of suitable attributes for computing results in a better recommendation. With limited information in hand, we produce additional user information gathered through GDUC analysis. By utilizing requirements analysis, more suitable attributes are identified besides the commonly used attributes. In addition, this approach can be used to accommodate personalization.

Our proposed method reveals some promising results in our ongoing research, in which additional information has some effect to the recommendation. However, the current evaluation method is not able to validate the personalization. The current experiments are done for the *k*-NN method. For future work, finding relationship between the user attributes and the classification methods must be conducted.

REFERENCES

1. G. Adomavicius and A. Tuzhilin, "Towards 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, 2005, pp. 734-749.
2. G. Lekakos, and G. M. Giaglis, "Improving the prediction accuracy of recommendation algorithms: Approaches anchored on human factors," *Interacting with Computers*, Vol. 18, 2006, pp. 410-431.
3. H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, Vol. 178, 2008, pp. 37-51.
4. M. Kompan, "Group inspired single-user collaborative filtering," *Information Sciences and Technologies Bulletin of the ACM Slovakia*, Vol. 4, 2012, pp. 35-37.
5. T.-M. Chang, W.-F. Hsiao, and W.-L. Chang, "An ordinal regression model with SVD hebbian learning for collaborative recommendation," *Journal of Information Science and Engineering*, Vol. 30, 2014, pp. 387-401.
6. M. Kompan and M. Bieliková, "Context-based satisfaction modelling for personalized recommendations," in *Proceedings of the 8th IEEE International Workshop on Semantic and Social Media Adaptation and Personalization*, 2013, pp. 33-38.
7. R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, Vol. 12, 2002, pp. 331-370.
8. B. Chikhaoui, M. Chiazzaro, and S. Wang, "An improved hybrid recommender system by combining predictions," in *Proceedings of IEEE Workshops of International Conference on Advanced Information Networking and Applications*, 2011, pp. 644-649.
9. A. B. Barragáns-Martínez, E. CostaMontenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Information Sciences*, Vol. 180, 2010, pp. 4290-4311.
10. P. Adamopoulos and A. Tuzhilin, "Recommendation opportunities: Improving item prediction using weighted percentile methods in collaborative filtering systems," in *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013, pp. 351-354.
11. M. Sun, F. Li, J. Lee, K. Zhou, G. Lebanon, and H. Zha, "Learning multiple-question decision trees for cold-start recommendation," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, 2013, pp. 445-454.

12. X. Yuan, J.-H. Lee, S.-J. Kim, and Y.-H. Kim, "Toward a user-oriented recommendation system for real estate websites," *Information Systems*, Vol. 38, 2013, pp. 231-243.
13. X. Ramirez-Garcia and M. García-Valdez, "Post-filtering for a restaurant context-aware recommender system," in *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, O. Castillo, P. Melin, W. Pedrycz, and J. Kacprzyk, eds., Springer International Publishing, 2014, pp. 695-707.
14. L. O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández, and J. J. Samper-Zapater, "RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes," *Expert Systems with Applications*, Vol. 42, 2015, pp. 1202-1222.
15. J. Lee and N.-L. Xue, "Analyzing user requirements by use cases: a goal-driven approach," *IEEE Software*, Vol. 16, 1999, pp. 92-101.
16. J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, Vol. 46, 2013, pp. 109-132.
17. Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Computing Surveys*, Vol. 47, 2014, pp. 3:1-3:45.
18. C. H. L. Lee and A. Liu, "A goal-driven approach for service request modeling," *International Journal of Intelligent Systems*, Vol. 25, 2010, pp. 733-756.
19. B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl, "MovieLens unplugged: Experiences with an occasionally connected recommender system," in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, 2003, pp. 263-266.
20. O. Ante, *et al.*, "Predicting and detecting the relevant contextual information in a movie-recommender system," *Interacting with Computers*, Vol. 25, 2013, pp. 74-90.
21. M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Journal of Foundations and Trends in Human-Computer Interaction*, Vol. 4, 2011, pp. 81-173.
22. T. L. Saaty, "How to make a decision: the analytic hierarchy process," *European Journal of Operational Research*, Vol. 48, 1990, pp. 9-26.

Pei-Chuan Zhou (周倍全) received the M.S. degree in Electrical Engineering from the National Chung Cheng University in Taiwan. He is a Ph.D. student in the Electrical Engineering Department at National Chung Cheng University. His current research interests are in data mining, machine learning, and recommender systems.

Alan Liu (劉立頌) received the Ph.D. degree in Electrical Engineering and Computer Science from the University of Illinois at Chicago in 1994. He is a Professor in the Department of Electrical Engineering, National Chung Cheng University in Taiwan. His research interests in artificial intelligence and software engineering include knowledge acquisition, requirements analysis, intelligent agents, and applications in embedded systems, and robotic systems. He is also a member of the IEEE, ACM, and TAAI.

Ching-Jung Lee (李靜如) received the M.S. degree in Electrical Engineering from the National Chung Cheng University in Taiwan, in 2015. Her research interests are data mining and recommender systems. She is an Engineer at HIWIN Technologies Corp. in Taichung, Taiwan.