Engineering Applications of Artificial Intelligence 41 (2015) 207–222

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

HU-FCF++: A novel hybrid method for the new user cold-start problem in recommender systems

Le Hoang Son *

VNU University of Science, Vietnam National University, 334 Nguyen Trai, Thanh Xuan, Ha Noi, Viet Nam

ARTICLE INFO

Article history:
Received 21 September 2014
Received in revised form
26 December 2014
Accepted 3 February 2015
Available online 16 March 2015

Keywords:
Collaborative filtering
HU-FCF++
Hybrid method
New user cold-start
Recommender systems

ABSTRACT

Recommender system (RS) is a special type of information systems that assists decision makers to choose appropriate items according to their preferences and interests. It is utilized in different domains to personalize its applications by recommending items, such as books, movies, songs, restaurants, news articles, jokes, among others. An important issue in RS namely the new user cold-start problem occurring when a new user migrates to the system has grasped a great attraction of researchers in recent years. Existing researches are faced with the limitations of the relied dataset, the determination of the optimal number of clusters, the similarity metric, irrelevant users and the selection of membership values. In this paper, we present a novel hybrid method so-called HU-FCF++ to deal with these drawbacks by considering the integration of existing state-of-the-arts of several groups of methods in order to combine the advantages of different groups and eliminate their disadvantages by some special procedures. A numerical example on a simulated dataset is given to illustrate the activities of the proposed approach. Experimental validation on the benchmark RS datasets show that HU-FCF++ achieves better accuracy than the relevant methods.

1. Introduction

Recommender system (RS) is a special type of information systems that assists decision makers to choose appropriate items according to their preferences and interests. RS is utilized in different domains to personalize its applications by recommending items, such as books, movies, songs, restaurants, news articles, jokes, among others. It has been applied to e-commerce to learn from a customer and recommend products that he will find most valuable from among the available products; thus helping the customer find suitable products to purchase. Some e-commerce RSs are named but a few (Shapira, 2011; Manouselis et al., 2012). For instance, Amazon.com is the most famous e-commerce RS, structured with an information page for each book, giving details of the text and purchase information. Two recommendations are found herein including books frequently purchased by customers who purchased the selected book and authors whose books are frequently purchased. EBay.com is another example providing the Feedback Profile feature that allows both buyers and sellers to contribute to feedback profiles of other customers with whom they have done business. The feedback consists of a satisfaction rating as well as a specific comment about other customers. In Moviefinder.com, customers can locate movies with a similar "mood, theme, genre or cast" through Match Maker or by their previously indicated interests through We Predict. Obviously, these examples have stressed the importance and practical applications of RS.

In this note, we deal with an important issue in RS namely the *new user cold-start problem* occurring when a new user migrates to the system. Being a new user, he has no prior rating for an item and then it is hard to give the prediction to any item in the system since the basic filtering methods in RS such as the collaborative filtering and the content-based filtering require the historic rating of this user to calculate the similarities for the determination of the neighborhood. For this reason, the new user cold-start problem can significantly affect negatively the recommender performance due to the inability of the system to produce meaningful recommendations (Safoury and Salah, 2013). Example 1 intuitively demonstrates the new user cold-start problem.

Example 1. We have three tables: the users' demographic data (Table 1), the movies' information (Table 2) and the rating (Table 3). In Table 1, Kim (User ID: 6) is a new user so that it is hard to give the prediction for the Titanic movie (ID: 1).

In order to deal with the new user cold-start problem, *existing researches* used one of following techniques: (i) making uses of additional data sources; (ii) choosing the most prominent groups of analogous users; and (iii) enhancing the prediction by hybrid methods (Son, *Information Systems)*. The principal idea of the *first group* is using some additional sources such as the demographic data

* Tel.: +84 904171284; fax: +84 0438623938.
E-mail addresses: sonlh@vnu.edu.vn, chinhson2002@gmail.com

http://dx.doi.org/10.1016/j.engappai.2015.02.003
0952-1976/

Table 1
Users' demographic data.

ID	Name	Age	Gender	Occupation
1	John	23	Male	Student
2	David	30	Male	Doctor
3	Jenny	29	Male	Student
4	Marry	20	Female	Engineer
5	Tom	30	Male	Engineer
6	**Kim**	**25**	**Female**	**Doctor**

Table 2
Movies' information.

ID	Name	Genre	Date	Sales
1	Titanic	Romantic	9/2004	150
2	Hulk	Horror	10/2005	300
3	Scallet	Romantic	6/2009	200

Table 3
Rating data.

User ID	Movie ID	Rating
1	2	4
1	3	2
2	1	4
2	2	3
2	3	1
3	2	2
3	1	1
4	2	3
5	2	3
5	3	2
6	**1**	**?**

(a.k.a. the users' profile), the users' opinions, social tags, etc. for the better selection of the neighbors of the new user. One of the most efficient algorithms in the first group is *MIPFGWC-CS* (Son et al., 2013). It uses a fuzzy geographically clustering algorithm such as MIPFGWC (Son et al., 2012a, 2012b, 2013, 2014; Son, 2014a, 2014b, 2014c, 2015; Son and Thong, 2015; Thong and Son, 2015) for the determination of similar users with respect to all attributes in the demographic data. Since the new user has no prior rating, the demographic data are the only medium to calculate the similarities between users. After finding similar users to the new one, MIPFGWC-CS checks whether they rated the considered item or not. If the ratings are found then consider them as the representative ratings of users. Otherwise, find a similar item to the considered one by the Pearson coefficient and assume that the rating on the similar item is the representative rating. Lastly, the rating of the new user to the considered item is approximated by the weighted average operator of the representative ratings.

The idea of the *second group* is to improve the methods determining the analogous users without the aid of additional data sources. Liu et al. (2014) presented a new user similarity model – *NHSM* to improve the recommendation performance in the cold-start situation that takes into account the global preference of user behaviors besides the local context information of user ratings. This heuristic similarity measure is composed of three factors of similarity such as Proximity, Significance and Singularity. Proximity considers the distance between two ratings. Significance shows that the ratings are more significant if two ratings are more distant from the median rating. Singularity represents how two ratings are different with other ratings. Furthermore, NHSM integrates the modified Jaccard and the user rating preference in the design.

The idea of the *third group* is to use hybrid methods for the calculation of similarity and/or the prediction of rating after determining the most analogous users to the new one. Leung et al. (2008) integrated fuzzy sets theory into association rules mining techniques and applied the proposed work – *FARAMS* to the collaborative filtering of recommender systems. Firstly, the rating data are converted to the transactional database of Association Rule mining and fuzzified by fuzzy memberships of linguistic variables and transformed into the type of transaction ID (TID) – Items where each TID is in the form of {Item, linguistic variable} and each item is a list of users with equivalent fuzzy memberships that opted the {Item, linguistic variable}. Then an Apriori-like algorithm is used to define candidate item sets and possible rules with the support of MinSupp and MinConf thresholds. The difference of this algorithm with the original Apriori algorithm is the uses of Fuzzy Support – $FC_{\langle\langle A,X\rangle,\langle B,Y\rangle\rangle}$ and Fuzzy Confidence $FC_{\langle\langle A,X\rangle,\langle B,Y\rangle\rangle}$ between two items A, B equipped by their memberships X, Y. Once defining the fuzzy rules, the predicting score of recommendable item is calculated and used to give the final rating of the new user. Another efficient algorithm in this group is the *HU-FCF* method (Son, 2014b). It integrates the fuzzy similarity degrees between users based on the demographic data with the hard user-based degrees calculated from the rating histories into the final similarity degrees. As such, those degrees would reflect more exactly the correlation between users in terms of the internal (attributes of users) and external information (interactions between users). Each similarity degree (fuzzy/hard) is accompanied by weights automatically calculated according to the numbers of analogous users. Once the final similarity degrees are calculated, the final rating will be constructed based on the rating values of neighbors of the considered user. Depending on the domain of a specific problem, the final rating will be approximated to its nearest value in that domain accompanied by an error threshold, which is normally smaller than 5%. A list of nearest values with equivalent error thresholds is also given as the prediction ratings of a user for an item.

Nonetheless, the mentioned algorithms have some *drawbacks*. *Firstly*, all algorithms rely either on the demographic or the rating data. If the relied dataset is not available, the algorithms could not work. *Secondly*, the optimal number of clusters for clustering algorithms such as MIPFGWC is undetermined. The exact number of clusters would lead to more accurate results of the similar users to a new user and thus enhancing the accuracy of prediction. Even though other parameters of MIPFGWC were suggested by Son et al. (2013), how to determine the optimal number of clusters is still an on-going research of this algorithm. *Thirdly*, in some algorithms such as MIPFGWC-CS and HU-FCF, defining the similarity metric between items is made through the Pearson coefficient, which has some limitations where there is a poor signal-to-noise ratio and negative spikes. In other words, if the relationship between two variables is non-linear, the Pearson coefficient cannot measure correlation accurately. *Fourthly*, irrelevant users produced by the GFD matrix in the HU-FCF algorithm and other demographic-based methods may be included in the computation of similarities; thus degrading the performance of the prediction. *Lastly*, the fuzzification in FARAMS could lead to inaccurate results of prediction. The question of how to set up the membership functions in an association rules-based algorithm like FARAMS is worth considering. Wrong membership values would result in the activities of the entire algorithm. In fact, not all recommender systems applications require fuzzy parameters so that for the sake of stability and processing time, the fuzzification step should be cut down. Nonetheless, the ideas of FARAMS could be useful to calculate the similarity between items.

From the analyses above, *our idea* in this proposal is to propose an integrated approach of existing standalone algorithms and employ some special procedures to enhance the accuracy of the approach. Specifically, *our contributions* are shortly summarized as follows:

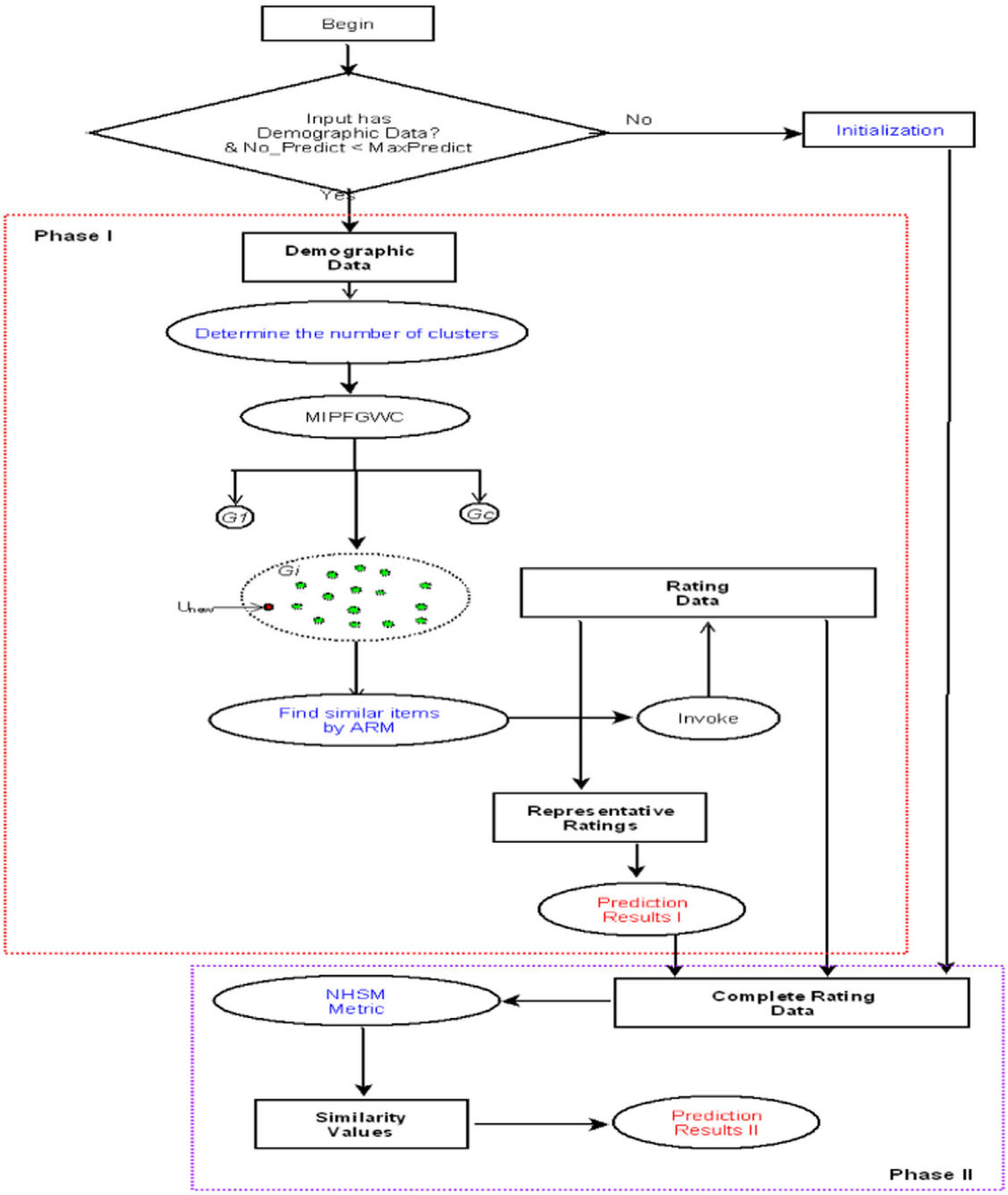

Fig. 1. The HU-FCF++ algorithm. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

a) A combination of HU-FCF (Son, 2014b) and the NHSM metric (Liu et al., 2014) described in Fig. 1 of Section 2.1 is proposed to handle the first limitation of the relied dataset. In this case, both demographic and rating data are employed in the proposal. A novel initialization procedure to create pre-ratings for the Complete Rating Data based on the idea of the most popular rating is attached into the combination;
b) A pre-processing procedure so-called FACA-DTRS (Yu et al., 2014) described in Fig. 1 (Phase I) of Section 2.1 is employed to automatically determine the number of clusters for handling the second limitation;
c) Two different similarity metrics are proposed to deal with the third limitation. Specifically, a novel variation of FARAMS (Leung et al., 2008) so-called Association Rules Mining (ARM) is presented in Fig. 1 (Phase I) of Section 2.1 to find similar items. In Phase II of Section 2.1, the NHSM metric (Liu et al., 2014) is employed for the similar tasks;
d) A combination of the fuzzy geographically clustering method – MIPFGWC (Son et al., 2013), which is the core part in MIPFGWC-CS and the ARM method described in Fig. 1 (Phase I) of Section 2.1 is used to tackle with the fourth limitation. In this case, only users belonged to the same group with the new user are counted for the calculation of similarity;
e) As suggested in the last limitation, the FARAMS method is not used but a variant of this method – ARM is utilized to calculate the similarity between items;
f) The cooperation mechanism of these methods is described in a novel hybrid method named as HU-FCF++ presented in Section 2. The differences and the advantages of HU-FCF++ in comparison with the relevant approaches are also described in this section;

Table 4
The pseudo-code of HU-FCF++ algorithm.

Input
- [Optional] The users' demographic data: $U=\{U_1,\ldots,U_N\}$ where each $U_i=\left\{U_i^1,\ldots,U_i^l\right\}$ $(i=\overline{1,N})$,N is the number of users and l is the number of demographic attributes, U_N is the cold-start user;
- The items set: $I=\{I_1,\ldots,I_M\}$ where M is the number of items;
- The rating data: $R=\{R(U_i,I_j)\,|\,U_i\in U;I_j\in I\}$;
- Parameters of MIPFGWC: threshold ε and other parameters m,η,τ, a_i $(i=\overline{1,3})$, $\gamma_j(j=\overline{1,C})$ where C is the number of clusters; Geographic parameters $\alpha,\beta,\gamma,a,b,c,d$;
- Parameters of ARM: *MinSupp* and *MinConf*;
- *MaxPredict*;
- A list of items – I^* to be predicted where its cardinality is larger than *MaxPredict*;

Output Ratings for I^*;

HU-FCF++

1: *No_Predict*=1;
2: Check whether or not the demographic data are provided in the Input and *No_Predict* < *MaxPredict*. If yes move to Step 3, otherwise move to Step 12;
3: Use the FACA-DTRS procedure to determine the number of clusters from the demographic data;
4: Set the parameters of MIPFGWC as in Son et al. (2013);
5: Use the MIPFGWC procedure to classify the demographic data into C groups. Determine which group U_N falls into;
6: Find the ratings of users in this group for item I^*[*No_Predict*] and consider them as the representative ratings;
7: In cases that a user did not rate for this item, use the ARM procedure to find the most similar rated item to I^*[*No_Predict*] and consider its rating as the representative rating;
8: The Prediction Results I (PR1) for I^*[*No_Predict*] is calculated as follows:

$$R^*=\frac{\sum_i w_i R_i}{\sum_i w_i}, \tag{1}$$

where R_i is a representative rating and w_i is the normalized weight of R_i calculated from the membership value of user i to the group of cold-start user;
9: Append the new rating to the rating data;
10: *No_Predict*=*No_Predict*+1;
11: If *No_Predict* > *MaxPredict* then go to Step 13. Otherwise go to Step 6;
12: [Initialization]:
If the demographic data are not provided then
- Calculate PR1 for I^*[*No_Predict*] as the most popular rating of all users based on the histogram of this item;
- Append the new rating to the rating data;
- *No_Predict*=*No_Predict*+1;
- Repeat Step 12 until *No_Predict* > $|I^*|\times 0.3$ and move to Step 13;

13: Use the NHSM metric to calculate the similarity matrix between the cold-start user U_N and other users in the group;
14: The *Prediction Results II* (PR2) for I^*[*No_Predict*] is calculated as follows:

$$R(a,i^*)=\overline{r_a}+\frac{\sum_{b\in U\setminus\{a\}} SIM(a,b)*\left(r_{b,i^*}-\overline{r_b}\right)}{\sum_{b\in U\setminus\{a\}}|SIM(a,b)|}, \tag{2}$$

where a is the cold-start user and b is a user in the group, $SIM(a,b)$ is the similarity value between these users taken from the similarity matrix, r_{b,i^*} is the rating of user b for the considered item, $\overline{r_a}$ and $\overline{r_b}$ are the average rating of user a and b, respectively;
15: *No_Predict*=*No_Predict*+1;
16: If *No_Predict* > $|I^*|$ then stop the algorithm, otherwise go to Step 13.

g) An illustrated example on a simulated dataset is given in Section 3.1 to demonstrate the activities of HU-FCF++;
h) The proposed HU-FCF++ method is experimentally validated on the benchmark RS datasets in terms of accuracy in Section 3.

The rest of the paper is organized as follows. The proposed hybrid method HU-FCF++ is presented in Section 2 including the difference of HU-FCF++ with the stand-alone approaches and the details of sub-procedures. In Section 3 we firstly give a numerical example on the dataset in Example 1 to illustrate the activities of HU-FCF++ and secondly validate the proposed approach through a set of experiments involving benchmark RS datasets. Finally, Section 4 draws the conclusions and delineates the future research directions.

2. The proposed HU-FCF++ method

In this section, we firstly present the mechanism of the new algorithm – HU-FCF++ in Section 2.1. The FACA-DTRS procedure (Yu et al., 2014) aiming to automatically determine the number of clusters is recalled in Section 2.2. Section 2.3 demonstrates the MIPFGWC algorithm (Son et al., 2013) used to find the group of analogous users to a new one. The novel Association Rules Mining (ARM) method designed to find similar items used in Phase I of Fig. 1 is presented in Section 2.4. Section 2.5 recalls the NHSM metric (Liu et al., 2014) used in Phase II of Fig. 1. Lastly, the differences and the advantages of HU-FCF++ in comparison with the relevant, standalone approaches namely MIPFGWC-CS, NHSM, FARAMS and HU-FCF are described in Section 2.6.

2.1. The algorithm

The limitations in Section 1 motivate us to design a novel hybrid method that combines the advantages of different groups and eliminates their disadvantages by some special procedures. Fig. 1 proposes the design of such the hybrid method. The HU-FCF++ algorithm is used to predict a list of ratings for given movies of the new user. It starts by checking whether or not the demographic data are provided in the data list and the number of predicted rating is smaller than a threshold – *MaxPredict*. If so, the algorithm moves to Phase I. Otherwise, it proceeds to the Initialization step of Phase II. HU-FCF++ has two main phases: Phase I and Phase II where Phase I is designed for the prediction of some first ratings with the support of the demographic data and Phase II is used to predict the last ratings in the list. The results of Phase I and Phase II are the Prediction Results I and II highlighted in red color in Fig. 1. The main activities of *Phase I* are the extensions of the MIPFGWC algorithm, which will be described in Section 2.3 with the provision of some procedures to eliminate the deficiencies of this algorithm. Specifically, the problem of determination of the optimal number of clusters in MIPFGWC is handled by the FACA-DTRS procedure,

Table 5
The pseudo-code of FACA-DTRS algorithm.

Input - The users' demographic data: $U=\{U_1,\ldots,U_N\}$
Output - The number of clusters
FACA-DTRS
1: Calculate $f(C_h,C_g)_N=\left\{f(C_h,C_g)|\ \forall h,g=\overline{1,N}\right\}$ by Eq. (3) and find $f_{max}=MAX_{h,g}(f(C_h,C_g)_N)$ under the condition $h<g$
2: If $f_{max}>0.5$ set $k_1=N$ and go to Step 3. Otherwise return$N$
3: If $k_1=2$ then return 1. Otherwise go to Step 4
4: Select $round\left(k_1-\sqrt{k_1}\right)$ maximal element from $f(C_h,C_g)_{k1}$ in descending order. Merge two clusters into one based on the maximal elements order until getting $round\left(\sqrt{k_1}\right)$ clusters
5: Calculate $f_{max}=f(C_h,C_g)_{\sqrt{k1}}$
6: If $f_{max}>0.5$ then set $k_1=\sqrt{k_1}$and go to Step 3. Otherwise set $k_2=\sqrt{k_1}$ and go to Step 7
7: If $k_2-k_1<2$ then return k_2. Otherwise set $k=(k_1+k_2)/2$ and go to Step 8
8: Select k_1-k maximal element from $f(C_h,C_g)_{k1}$ in descending order. Combine two clusters into one based on the maximal elements order until getting k clusters
9: Calculate $f_{max}=f(C_h,C_g)_k$
10: If $f_{max}>0.5$ then set $k_2=k$. Otherwise set $k_1=k$
11: Go to Step 7

Table 6
Normalized users' demographic data.

ID	Name	Age	Gender	Occupation
1	John	0.766666667	1	0
2	David	1	1	0.333333333
3	Jenny	0.966666667	0	0.666666667
4	Marry	0.666666667	0	1
5	Tom	1	1	1
6	Kim	0.833333333	0	0.333333333

Table 7
The similarity matrix between two pair of elements.

	1	2	3	4	5	6
1	1	0.713005	0.140623	0	0.275708	0.255015
2	0.713005317	1	0.256129	0.120279	0.52977	0.284925
3	0.140622566	0.256129	1	0.683685	0.256129	0.746773
4	0	0.120279	0.683685	1	0.2565	0.515298
5	0.275707776	0.52977	0.256129	0.2565	1	0.144168
6	0.255014924	0.284925	0.746773	0.515298	0.144168	1

Table 8
The P matrix between two pair of elements.

	1	2	3	4	5	6
1	1	0.737024	0.154756929	0	0.30342	0.280647
2	0.737023649	1	0.281872995	0.132369	0.569123	0.313564
3	0.154756929	0.281873	1	0.710157	0.281873	0.767966
4	0	0.132369	0.710156979	1	0.282282	0.555862
5	0.303419927	0.569123	0.281872995	0.282282	1	0.158658
6	0.280647179	0.313564	0.767965504	0.555862	0.158658	1

which is highlighted in blue color in Fig. 1 and will be described in Section 2.2.

After determining the number of clusters, the MIPFGWC algorithm is used to classify the demographic data into groups and specify the group containing the new user. Then we check whether users in this group except the new one have rated the considered item or not. If yes, consider them as the representative ratings. Otherwise, we have to find the similar rated item to the considered one and take its rating as the representative rating. In the MIPFGWC-CS algorithm, the authors used the Pearson coefficient for this task. Yet we have pointed out the limitation of this measure in Section 1 so that it is better to integrate another method therein. Furthermore, we

Table 9
The f matrix between two pair of clusters.

	1	2	3	4	5	6
1	1	0.737	0.155	0	0.303	0.281
2	0.737	1	0.282	0.132	0.569	0.314
3	0.155	0.282	1	0.71	0.282	0.768
4	0	0.132	0.71	1	0.282	0.556
5	0.303	0.569	0.282	0.282	1	0.159
6	0.281	0.314	0.768	0.556	0.159	1

Table 10
The $f(C_h,C_g)_{\sqrt{k1}}$ matrix between two pair of clusters.

	1	2	3
1	0.869	0.194	0.436
2	0.194	0.785	0.241
3	0.436	0.241	1

have shown that FARAMS could be regarded as an efficient method to calculate the similarity between items. Nevertheless, using the fuzzification in the FARAMS method will result in high time complexity and the vagueness in selecting the membership functions. Thus, in order to avoid these limitations, we propose a new procedure to find similar items by Association Rules Mining (ARM) working directly with the rating data. This procedure will be described in Section 2.4. Outputs of the ARM procedure is the most similar item to the considered one accompanied with a rule score of ARM. Once the representative ratings are found, the predictive rating of the new user to the considered item (Prediction Result I) is approximated by the weighted average operator of the representative ratings. Phase I stops an iteration step after the new predicted rating is appended to the rating data (Complete Rating Data) and the number of predicted rating increases by one unit. Once the number of predicted rating is larger than *MaxPredict*, Phase I stops its operations. By using the hybrid method between MIPFGWC, ARM and the FACA-DTRS procedure, this eliminates the weakness of MIPFGWC-CS and FARAMS stated in Section 1. The first limitation of additional data in Section 1 is solved by taking advantages of the hybrid mechanism in Fig. 1. *Phase II* start working when either the number of predicted rating is larger than *MaxPredict* or the demographic data is not provided. In the first case, since the rating data is now completed, we can use the NHSM metric, which will be mentioned in Section 2.5 to calculate the similarity values and make the prediction of ratings for the last items. In the remaining case, the

Table 11
The pseudo-code of MIPFGWC procedure.

Input Geo-demographic data X. The number of elements (clusters) – $N(C)$. The dimension of dataset r. Threshold ε and other parameters m, η, τ, a_i $(i=\overline{1,3})$, $\gamma_j (j=\overline{1,C})$. Geographic parameters $\alpha, \beta, \gamma, a, b, c, d$.

Output Final membership values u'_k and centers $V^{(t+1)}$

MIPFGWC

1: Set the number of clusters C, threshold $\varepsilon > 0$ and other parameters such as $m, \eta, \tau > 1$, $a_i > 0$ $(i=\overline{1,3})$, γ_j $(j=\overline{1,C})$ as in Son et al. (2013)

2: Initialize centers of clusters $V_j, j=\overline{1,C}$ at $t=0$

3: Set geographic parameters $\alpha, \beta, \gamma, a, b, c, d$ satisfying condition (7)

$$\alpha+\beta+\gamma=1. \tag{7}$$

4: Use the formulas (8)–(10) to calculate the membership values, the hesitation level and the typicality values, respectively

$$u_{kj}=\frac{1}{\left(\sum_{i=1}^{C}\|X_k-V_j\|/\|X_k-V_i\|\right)^{2/(m-1)}}, \quad k=\overline{1,N};\ j=\overline{1,C}, \tag{8}$$

$$h_{kj}=\frac{1}{\left(\sum_{i=1}^{C}\left(\|X_k-V_j\|/\|X_k-V_i\|\right)\right)^{2/(\tau-1)}}, \quad k=\overline{1,N};\ j=\overline{1,C}, \tag{9}$$

$$t_{kj}=\frac{1}{1+\left(a_2\|X_k-V_j\|^2/\gamma_j\right)^{1/(\eta-1)}}, \quad k=\overline{1,N};\ j=\overline{1,C}. \tag{10}$$

5: Perform geographic modifications through Eqs. (11) and (12)

$$u'_k=\alpha u_k+\beta\sum_{j=1}^{k-1}w_{kj}u'_j+\gamma\frac{1}{A}\sum_{j=k}^{C}w_{kj}u_j, \tag{11}$$

$$w_{kj}=\begin{cases}\frac{(pop_k\times pop_j)^b\times p_{kj}^c\times IM_{kj}^d}{d_{kj}^a}, & k\neq j\\ 0, & else\end{cases} \tag{12}$$

6: If $\{u'_k\}$ is a completely monotone increasing sequence or $u_k \geq u'_k$ for most $k=\overline{1,C}$ then conclude that there is no suitable solution for given geographic parameters. Otherwise, go to Step 7.

7: Calculate the centers of clusters at $t+1$ by Eq. (13)

$$V_j=\frac{\sum_{k=1}^{N}\left(a_1u_{kj}^m+a_2t_{kj}^\eta+a_3h_{kj}^\tau\right)X_k}{\sum_{k=1}^{N}\left(a_1u_{kj}^m+a_2t_{kj}^\eta+a_3h_{kj}^\tau\right)}, \quad j=\overline{1,C}. \tag{13}$$

8: If the difference $\|V^{(t+1)}-V^{(t)}\|\leq\varepsilon$ then stop the algorithm. Otherwise, assign $V^{(t)}=V^{(t+1)}$ and return to Step 4.

Table 12
The transactional database.

Transactional ID	Movies
1	{2, 3}
2	{1, 2, 3}
3	{1,2}
4	{2}
5	{2,3}
6	NULL

Table 13
The supports of movies.

Movie	Support
1	2
2	5
3	3

Table 14
The supports of 2-candidates item list.

Movie	Support
{1, 2}	2
{1, 3}	1
{2, 3}	3

Initialization step will approximate 30% of the first ratings of the new user in the list by the most popular rating of all users based on the histogram of an item. The new approximated data are appended to the Complete Rating Data. The reason for doing so is to make a pre-knowledge for the prediction of other items in the list based on the most popular rating of other users. The ideas are quite intuitive: "if most people prefer an item then it is likely that the new user will prefer that item". The number of 30 percents is based on heuristic, that is to say it is either not large or not small but adequate to build the pre-knowledge basis. Once the Complete Rating Data has been set up, similar steps to the first case are performed for the last items. The results of Phase II are Prediction Results II highlighted in red color in Fig. 1. By using this mechanism, the disadvantages of NHSM stated in the first and fourth limitations in Section 1 are solved. The proposed HU-FCF++ are able to handle the deficiencies of HU-FCF stated in the first, third and fourth limitations in Section 1 regarding the using of demographic data only, the Pearson coefficient and the irrelevant users. Table 4 shows the pseudo-code of the HU-FCF++ algorithm for the detailed explanation of the working mechanism.

Table 15
The confidence of rules.

Rule	Confidence
1→2	1
3→2	1

Table 16
The scores of rules.

Rule	Score
1→2	2
3→2	3

2.2. FACA-DTRS: the procedure for the determination of number of clusters

In this section, we recall the procedure so-called FACA-DTRS originated from the recent work of Yu et al. (2014) to determine the number of clusters from the demographic data. FACA-DTRS was designed on the basis of the decision-theoretic rough set model for clustering. The basic idea of this hierarchical-like algorithm is to set each object is a cluster and combine two clusters into a unique one in each step until there is only one remaining cluster or the termination

condition is satisfied. The 'distance' between two clusters C_g and C_h is measured by $f(C_g, C_h)$function as follows:

$$f(C_g, C_h) = \frac{1}{|C_h| * |C_g|} \sum_{x_i \in C_h} \sum_{x_j \in C_g} P(x_i, x_j), \tag{3}$$

$$P(x_i, x_j) = \begin{cases} 0.5 + \frac{sim(x_i, x_j) - val}{2 - 2 * val}, & sim(x_i, x_j) \geq val, \\ 0.5 - \frac{val - sim(x_i, x_j)}{2 * val}, & sim(x_i, x_j) < val, \end{cases} \tag{4}$$

$$val = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} sim(x_i, x_j), \tag{5}$$

$$sim(x_i, x_j) = 1 - \frac{\sqrt{\sum_{a_l \in A} \left(x_i^{a_l} - x_j^{a_l}\right)^2}}{max_{i,j} \sqrt{\sum_{a_l \in A} \left(x_i^{a_l} - x_j^{a_l}\right)^2}} \tag{6}$$

where A is the set of possible feature values of x_i and x_j. The authors stated that if $f(C_g, C_h) < \frac{1}{2}$ then should not merge cluster. Thus, if $f_{max} = \{f(C_g, C_h) | \forall g \in [1, \ldots, n], \forall h \in [1, \ldots, n], h \neq g\} < \frac{1}{2}$ then end algorithm. The time complexity of FACA-DTRS is $O(n^2)$. Table 5 describes the pseudo-code of FACA-DTRS.

Example 2. Consider the users' demographic data in Table 1. Transform the discrete values into continuous ones and normalize them, we get the results in Table 6. In this dataset we have $N = 6$ and $l = 3$. Then we calculate the similarity matrix between two pair of element based on Eq. (6) and get the results in Table 7. From Eq. (5), $val = 0.454333666$ and the P matrix is shown in Table 8. From Step 1 of FACA-DTRS we calculate $f(C_h, C_g)_N$ and get the results in Table 9.

Being aware that $f_{max} = 0.768 > 0.5$, the algorithm sets $k_1 = 6$ and moves to Step 3. Because $k_1 = 6 > 2$, it continue to go to Step 4. Since $round\left(\sqrt{k_1}\right) = 3$, we partition objects to clusters in order to have three clusters ($k_1 - round\left(\sqrt{k_1}\right) = 3$). The three clusters are $\{3, 4, 6\}$, $\{1, 2\}$, $\{5\}$. Step 5 calculates $f(C_h, C_g)_{\sqrt{k1}}$ and takes the results in Table 10. From this table, we have $f_{max} = 0.436$ (ignored the elements in the matrix diagonal). Because $f_{max} = 0.436 < 0.5$ the algorithm sets $k_2 = round\left(\sqrt{k_1}\right) = 2$ and moves to Step 7.

In this step, the stopping condition $k_2 - k_1 < 2$ is satisfied. Thus, the FACA-DTRS algorithm finishes with the optimal number of clusters being calculated as 2.

2.3. Determining the group of analogous users by the MIPFGWC algorithm

After determining the optimal number of clusters, a fuzzy geographically clustering algorithm such as MIPFGWC is used to determine the group of analogous users to the new one. MIPFGWC (Son et al., 2013) is the state-of-the-art fuzzy clustering algorithm for demographic segmentation as it showed the advantages in terms of accuracy in comparison with other relevant algorithms such as IPFGWC, FGWC, NE and FCM. MIPFGWC was constructed on the basis of intuitionistics fuzzy sets and possibilistic fuzzy clustering described in Eqs. (8)–(10), (13). It also used the Spatial Interaction – Modification Model (SIM^2) expressed in Eqs. (7), (11), (12) to geographically update the membership values. Being integrated these main parts, MIPFGWC has good clustering quality and is suitable for our considered problem. The pseudo-code of MIPFGWC is highlighted in Table 11.

2.4. ARM: finding similar items by Association Rules Mining

In this section, we present a novel procedure to find similar items by Association Rules Mining (ARM). The basic idea of the proposed procedure is to find all association rules in the form of "$x \rightarrow y$" where x, y are items and y is the single item being considered and regard the item x having largest rule score as the most similar item. In order to determine possible association rules, the APRIORI-like algorithm (Agrawal and Srikant, 1994), which is the most popular mining algorithm, is employed. ARM starts by considering the user-item matrix as the transaction database where the User ID is the transactional ID.

Example 3. Consider the Rating dataset in Table 3. They are converted into the transactional database in Table 12.

Then the support of each item, which in essence is the number of occurrences, is calculated as follows:

$$Supp(X_i) = |X_i|. \tag{14}$$

We keep the item whose support is larger or equal to the *MinSupp* threshold. In Table 13, *MinSupp* = 2 so that all items are frequent. The next step is to generate a list of all pairs of the frequent items and calculate its supports Table 14.

From Table 14, we recognize that frequent sets {1, 2} and {2, 3} hold the *MinSupp* constraint. Since the aim is to find the rule in the form of "$x \rightarrow y$", we stop expanding the candidates item list. From the frequent sets, there are two possible rules if the considered item is 2.

$$\text{Rule 1}: \quad 1 \rightarrow 2 \tag{15}$$

$$\text{Rule 2}: \quad 3 \rightarrow 2 \tag{16}$$

Next we calculate the confidences of those rules by the formula below:

$$Conf(X \Rightarrow Y) = P(Y|X) = Supp(X \cup Y)/Supp(X). \tag{17}$$

Since the *MinConf* = 1, no rule is ignored. Now we calculate the rule scores as follows:

$$Score = Supp(X_i) \times Conf(X_i \Rightarrow Y). \tag{18}$$

Thus, the most similar item to item 2 is 3 (Tables 15 and 16). Outputs of the ARM procedure is the most similar item to the considered one accompanied with a rule score of ARM. Table 17 describes the pseudo-code of ARM.

2.5. The NHSM metric

This section describes the NHSM metric used in Phase II (Fig. 1) of Section 2.1. As mentioned in Section 1, NHSM (Liu et al., 2014) consists of three factors of similarity namely Proximity, Significance and Singularity, which are described in the following equations:

$$sim(u, v)^{NHSM} = sim(u, v)^{JPSS} sim(u, v)^{URP}, \tag{19}$$

$$sim(u, v)^{URP} = 1 - \frac{1}{1 + \exp(-|\mu_u - \mu_v||\sigma_u - \sigma_v|)}, \tag{20}$$

$$sim(u, v)^{JPSS} = sim(u, v)^{PSS} sim(u, v)^{Jaccard}, \tag{21}$$

$$sim(u, v)^{Jaccard} = \frac{|I_u \cap I_v|}{|I_u| \times |I_v|}, \tag{22}$$

$$sim(u, v)^{PSS} = \sum_{p \in I} Proximity(r_{u,p}, r_{v,p}) Significance(r_{u,p}, r_{v,p}) Singularity(r_{u,p}, r_{v,p}), \tag{23}$$

$$Proximity(r_{u,p}, r_{v,p}) = 1 - \frac{1}{1 + \exp(-|r_{u,p} - r_{v,p}|)}, \tag{24}$$

$$Significance(r_{u,p}, r_{v,p}) = \frac{1}{1 + \exp(-|r_{u,p} - r_{med}||r_{v,p} - r_{med}|)}, \tag{25}$$

Table 17
The pseudo-code of ARM.

Input	– The items set: $I=\{I_1,\ldots,I_M\}$ where M is the number of items – *MinSupp* and *MinConf* – The considered item I_+ – The active user U
Output	- The most similar item to I_+
ARM	
1:	Generate the transaction database from the user-item matrix
2:	Calculate the support of each item that was rated by U and remove items smaller than *MinSupp*
3:	Generate the 2-candidates item list
4:	Calculate the support of this list and remove items smaller than *MinSupp*
5:	Generate rules in the form of $I_j \rightarrow I_+$ from the valid list
6:	Calculate the confidences of those rules and remove rules having the confidence smaller than *MinConf*
7:	Among all valid rules, calculate the rules scores and choose the rule having largest value among all. In case of many largest value, choose a ubiquotous rule
8:	If no valid rule if found, choose the item having largest support value as the most similar item

Table 18
The pseudo-code of NHSM.

Input	- The rating dataset
Output	- The new rating dataset
NHSM	
1:	Calculate the similarity matrix between users by the NHSM metric in (19)
2:	Use the following equation for the prediction $r_{u,i}=\overline{r_u}+\frac{\sum_{v\in NG(u)} sim(u,v)^{NHSM}(r_{v,i}-\overline{r_v})}{\sum_{v\in NG(u)} sim(u,v)^{NHSM}}$, (27) where $\overline{r_u}$ ($\overline{r_v}$) is the mean value of rated items of user u (v).$r_{u,i}$ ($r_{v,i}$) is the rating of user u (v) for item i. $NG(u)$ is the set of neighbors of u. u is the new user cold-start.

$$Singularity(r_{u,p},r_{v,p})=1-\frac{1}{1+\exp\left(-\left|(r_{u,p}+r_{v,p}/2)-\mu_p\right|\right)}, \quad (26)$$

where μ_u and σ_u are the mean rating and the standard variance of user u, respectively. I_u represents for the set of ratings of user u. The operator $\times$ means the common ratings between two users. $r_{u,p}$ is the rating of user u on item p. r_{med} is the median value in the rating scale. Table 18 describes the pseudo-code of NHSM.

2.6. The differences of HU-FCF++ with the relevant approaches

In this section, we clearly point out the differences and the advantages of HU-FCF++ in comparison with the relevant approaches stated in Section 1. They are summarized as follows:

- *Firstly*, the proposed HU-FCF++ is the generalization of the existing approaches such as MIPFGWC-CS, NHSM and HU-FCF. Specifically, Phase I of HU-FCF++ mostly employed the ideas of MIPFGWC-CS, Phase II utilized the NHSM-based algorithm, and the cooperation between Phase I and Phase II is done similarly to the ideas of HU-FCF with the Prediction Results I and II being regarded as the outputs of the fuzzy and hard steps in HU-FCF;
- *Secondly*, HU-FCF++ is not a trivial generalization of existing approaches. It makes uses of some special procedures to handle the limitations of those approaches. Specifically, in Phase I the FACA-DTRS procedure is used to determine the number of clusters for the clustering of demographic data in MIPFGWC. MIPFGWC is utilized to specify the group of analogous users to a new one. The novel ARM procedure is employed to find out which is the most similar item to a considered one. In Phase II, the novel Initialization procedure is used to create the Complete Rating Data. The NHSM metric is then selected to make the prediction from this dataset. By using these procedures, some major problems of existing algorithms such as the relied dataset, the determination of the optimal number of clusters, the similarity metric, irrelevant users and the selection of membership values were handled and have been clearly shown in the proposed HU-FCF++ algorithm;
- *Thirdly*, HU-FCF++ could predict the ratings for a number of items. This is difference to those in other algorithms where only one rating is predicted during the activities of algorithms. Nonetheless, the HU-FCF++ algorithm still contains some limitations such as
- HU-FCF++ requires large computational time. As being recognized in Fig. 1, HU-FCF++ invokes many procedures to compute the final ratings and the number of items to be predicted is larger than those of the relevant approaches so that the computational time of HU-FCF++ increases remarkably;
- Setting up the values of parameters in HU-FCF++ is another concerned problem. Even though some parameters such as those of MIPFGWC were suggested in equivalent articles, minor left parameters such as the *MinConf* and *MinSupp* should be paid much attention in order to achieve good performance of the algorithm.

3. Evaluation

3.1. A numerical example

In this section, we give a numerical example on the simulated dataset from Tables 1–3 (Section 1) to illustrate the activities of the HU-FCF++. According to Fig. 1, since the demographic dataset is provided, the algorithm immediately moves to Phase I. In this phase, the FACA-DTRS procedure is invoked to determine the number of clusters from the demographic dataset. The results of Example 2 in Section 2.2 showed that the optimal number of clusters is 2. Using the MIPFGWC algorithm in Section 2.3 to classify the demographic dataset into 2 groups, we receive the membership matrix as follows:

$$U^{(MIPFGWC)}=\begin{pmatrix}0.025868, 0.974132\\ 0.998475, 0.001525\\ 0.962639, 0.037361\end{pmatrix}$$

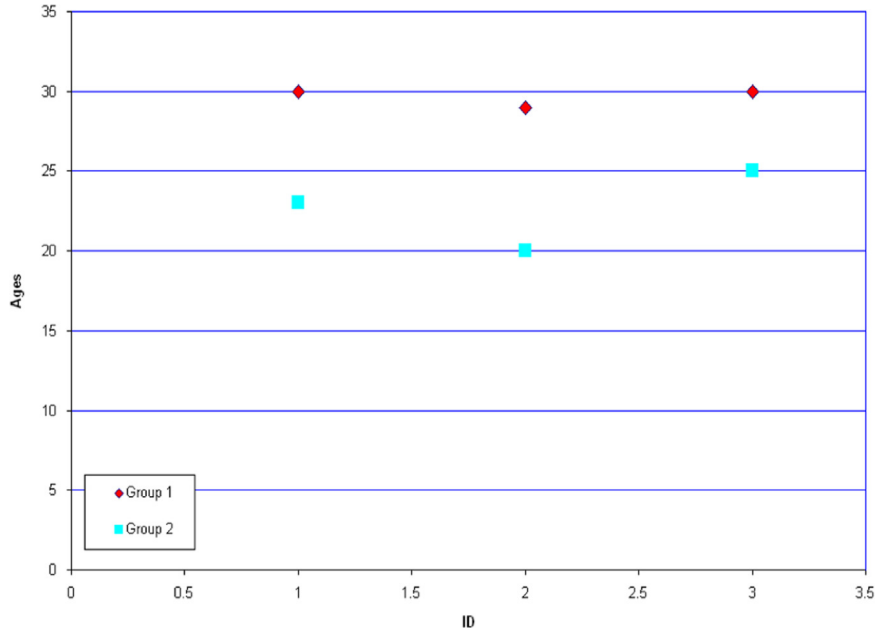

Fig. 2. The distribution of the demographic dataset by groups.

0.079178, 0.920822
0.982933, 0.017067
0.203170, 0.796830). (28)

The distribution of the demographic dataset by groups is depicted in Fig. 2. From Eq. (28), we determine the users of each group.

Group 1: David (User ID: 2), Jenny (User ID: 3), Tom (User ID: 5).
Group 2: John (User ID: 1), Marry (User ID: 4), Kim (User ID: 6).

According to Table 3, we need to make prediction of the new user Kim for the Titanic movie (Movie ID: 1). It could be approximated by the average rating of the similar users to Kim in Group 2. Nonetheless, both John (User ID: 1) and Marry (User ID: 4) did not rate for the Titanic movie (Movie ID: 1) beforehand. Thus, the ARM procedure in Section 2.4 is used to find the most similar rated movie by a given user to the Titanic movie (Movie ID: 1). By similar processes as in Example 3, the most similar movie to Titanic is Hulk (Movie ID: 2). Additionally, Table 3 shows that the representative ratings of John (User ID: 1) and Marry (User ID: 4) are 4 and 3, respectively. Therefore, the Prediction Results I is calculated as

$$R*(6,1)=\left[\frac{0.974123*4+0.920822*3}{0.974123+0.920822}\right]=4, \quad (29)$$

where 0.974123 and 0.920822 are the membership values of John (User ID: 1) and Marry (User ID: 4) in the membership matrix. Phase I stops working.

Now, we continue to make prediction of user Kim for two left movies: Hulk and Scallet. Since $MaxPredict=2$, we perform similar steps in Phase I to calculate the predictive rating of user Kim for Hulk (Movie ID: 2) as follows:

$$R*(6,2)=\left[\frac{0.974123*4+0.920822*3}{0.974123+0.920822}\right]=4. \quad (30)$$

Next, because $No\_Predict=MaxPredict$, the calculation for Scallet movie is executed in Phase II. At this point of time, the rating data have been supplemented by the ratings of Kim to the Titanic and Hulk movies. Therefore, we can use the NHSM metric to calculate the similarity values between Kim and two other users in Group 2 namely John and Marry (Table 19), and make prediction for the Scallet movie.

Table 19
The NHSM values.

Cold-Start user	User	NHSM
Kim	John	0.005
Kim	Marry	0.0246

The Prediction Results II for the Scallet movie is calculated as

$$R*(6,3)=\left[\frac{3.514066+3.514066}{2}+\frac{0.005*(2-3)}{0.005}\right]=3. \quad (31)$$

3.2. Experimental environment

In this section, we describe the experimental environments such as

- *Experimental tools*: We have implemented the proposed HU-FCF++ method in addition to MIPFGWC-CS (Son et al., 2013), NHSM (Liu et al., 2014), FARAMS (Leung et al., 2008) and HU-FCF (Son, 2014b) in C programming language and executed it on a PC Intel Pentium 4, CPU 2.66 GHz, 1 GB RAM, 80 GB HDD.
- *Experimental datasets*: We use the benchmark RS datasets as follows:
  - *MovieLens 1M* (MovieLens, 2013): contains 1,000,209 anonymous ratings of approximately 3900 movies made by 6040 MovieLens users. Ratings are discrete values from 1 to 5. Demographic data are provided in the following form "Gender:: Age:: Occupation:: Zip-code".
  - *Jester* (Jester, 2013): contains ratings of 100 jokes from 73,421 users. Ratings are real values ranging from −10 to 10. The value "99" corresponds to "null"="not rated". Demographic data are no longer support for this dataset.
- *Generating cold-start users*: We adopt the *K*-fold cross validation method to generate the cold-start users with *K* being from 2 to 10. Specifically, the rating data as in Table 3 are converted into a 2D matrix with rows and columns being the users and the items,

respectively. For a given value of K, this matrix is randomly divided into K parts by rows where $(K-1)$ parts are used f or the training set and the rest is the testing set. In the testing set, all ratings of users except two first ones of each user are cleared and assigned predictive values by the algorithms above. The predictive ratings are compared with the accurate ones by evaluation indices to measure the accuracy. This process is analogously performed for other $(K-1)$ randomly division of this dataset. The final accuracies of algorithms are the average results among all divisions of the dataset. By the similar calculation, we also obtain the accuracies of algorithms with other values of K.

- *Evaluation indices*: We use the *Mean Absolute Error* (MAE) and *Root Mean Square Error* (RMSE) for the validation of accuracy.

$$MAE = \frac{1}{N}\sum_{u,i}|p_{u,i} - r_{u,i}|, \tag{32}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(p_{u,i} - r_{u,i})^2}. \tag{33}$$

where $p_{u,i}$ ($r_{u,i}$) is the predicted (real) rating of user u for item i.

- *Parameters setting*: The *MaxPredict* parameter is set as half of the number of items to be predicted.
- *Experimental objectives*:
  - To validate the accuracy of HU-FCF++ in comparison with other algorithms according to evaluation indices;
  - To measure the stability of HU-FCF++ by various datasets and parameters;
  - To verify the drawbacks of HU-FCF++ stated in Section 2.6.

3.3. Assessment

In Tables 20 and 21, the experimental results of algorithms on the MovieLens and Jester datasets are presented respectively. Each table consists of the comparative results in terms of MAE, RMSE and the computational time according to the number of folds (K). In Table 21, the results of MIPFGWC-CS on the Jester dataset are missing since this dataset does not support the demographic, which is essential for the calculation mechanism in MIPFGWC-CS. Figs. 3 and 4 illustrate MAE and RMSE values of algorithms on the MovieLens dataset, respectively. Similarly, Figs. 5 and 6 depict *MAE* and *RMSE* values of algorithms on the Jester dataset, respectively. The experimental results including the tabular and chart representations help us understand the efficiencies of algorithms by various datasets and numbers of folds. Since the K-fold cross validation method is used in the experiments, these results are independent from the constitution of the training and testing datasets and are unbiased and objective in the judgment of the performances of algorithms. Diverse values of K also point out the trends of variation in the accuracies of algorithms, and let us know how these algorithms are efficient in different conditions of testing. The changing of results from a dataset supporting both the demographics and rating like MovieLens to another having the rating only like Jester is clearly shown in the tables and figures.

Obviously, the accuracies of HU-FCF++ in terms of *MAE* and *RMSE* values are better than those of other algorithms. According to Table 20, the average *MAE* values of HU-FCF++, MIPFGWC-CS, NHSM, FARAMS and HU-FCF on the MovieLens dataset are 0.61, 0.74, 0.68, 0.66 and 0.69, respectively. The average *RMSE* values of HU-FCF++, MIPFGWC-CS, NHSM, FARAMS and HU-FCF on the MovieLens dataset are 0.77, 0.95, 0.97, 0.92 and 0.93, respectively. This table clearly shows that the MAE and RMSE values of HU-FCF++ are the smallest among all. We know that the MovieLens dataset consists of both the demographic and the rating. The mechanism of HU-FCF++ as shown in the pseudo-code in Table 4 demonstrates that the proposed algorithm works efficiently with the full dataset like MovieLens since the cooperation between some special procedures in Phase I of HU-FCF++ such as FACA-DTRS,

Table 20
The results of k-fold cross validation on the MovieLens dataset.

Fold	HU-FCF++	MIPFGWC-CS	NHSM	FARAMS	HU-FCF
MAE					
2	0.696	0.804	0.758	0.790	0.806
3	0.650	0.782	0.746	0.679	0.793
4	0.646	0.798	0.725	0.659	0.731
5	0.640	0.745	0.687	0.658	0.698
6	0.625	0.722	0.684	0.707	0.672
7	0.596	0.718	0.665	0.641	0.652
8	0.588	0.703	0.631	0.632	0.641
9	0.566	0.693	0.615	0.605	0.625
10	0.543	0.672	0.603	0.591	0.608
RMSE					
2	0.848	1.203	1.138	1.102	1.163
3	0.825	1.045	1.025	0.989	1.032
4	0.823	0.998	1.036	0.992	0.969
5	0.801	0.968	0.980	0.972	0.912
6	0.759	0.903	0.976	0.850	0.893
7	0.743	0.896	0.982	0.871	0.886
8	0.742	0.882	0.901	0.852	0.842
9	0.738	0.856	0.885	0.849	0.843
10	0.719	0.824	0.801	0.815	0.840
Computational time (s)					
2	525.6	963.26	4.3	48.3	345.32
3	546.3	1123.2	5.6	49.6	356.70
4	598.9	1254.6	6.3	51.6	489.63
5	623.3	1321.8	6.2	54.3	478.34
6	675.0	1345.4	6.8	58.5	498.43
7	690.2	1235.2	7.7	60.8	552.18
8	714.7	1543.6	8.0	65.2	603.22
9	732.4	1537.7	10.3	76.8	643.43
10	768.3	1843.4	10.9	80.3	668.98

Table 21
The results of k-fold cross validation on the Jester dataset.

Fold	HU-FCF++	NHSM	FARAMS	HU-FCF
MAE				
2	0.856	0.898	0.903	1.123
3	0.836	0.844	0.878	1.102
4	0.820	0.825	0.853	1.097
5	0.815	0.847	0.835	1.002
6	0.796	0.814	0.819	0.992
7	0.770	0.795	0.784	0.947
8	0.732	0.747	0.793	0.909
9	0.719	0.745	0.742	0.832
10	0.693	0.703	0.723	0.808
RMSE				
2	0.948	1.203	1.304	1.534
3	0.915	1.102	1.134	1.432
4	0.894	1.003	1.145	1.269
5	0.857	0.993	1.091	1.101
6	0.823	0.987	0.989	0.994
7	0.812	1.002	0.963	0.982
8	0.794	0.982	0.942	0.985
9	0.782	0.972	0.923	0.934
10	0.776	0.897	0.912	0.915
Computational time (s)				
2	256.3	12.4	75.2	304.5
3	289.6	13.4	76.0	334.2
4	301.9	15.2	79.4	365.6
5	323.8	15.6	85.6	398.0
6	345.1	17.2	88.2	415.6
7	368.2	18.4	91.8	405.9
8	389.7	18.9	93.8	425.3
9	390.2	18.5	96.9	420.6
10	413.4	18.4	100.3	489.5

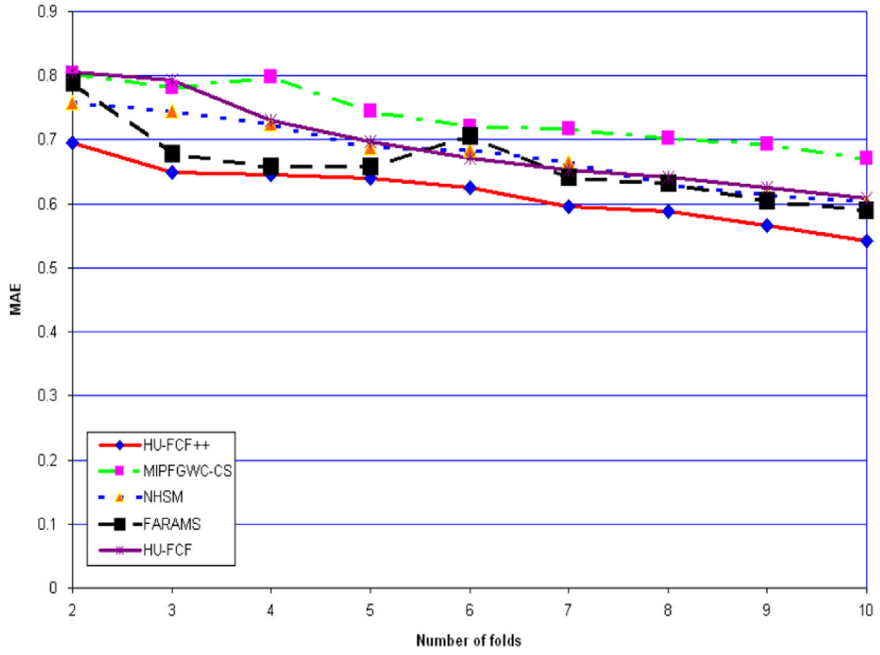

Fig. 3. The *MAE* values of algorithms on the MovieLens dataset.

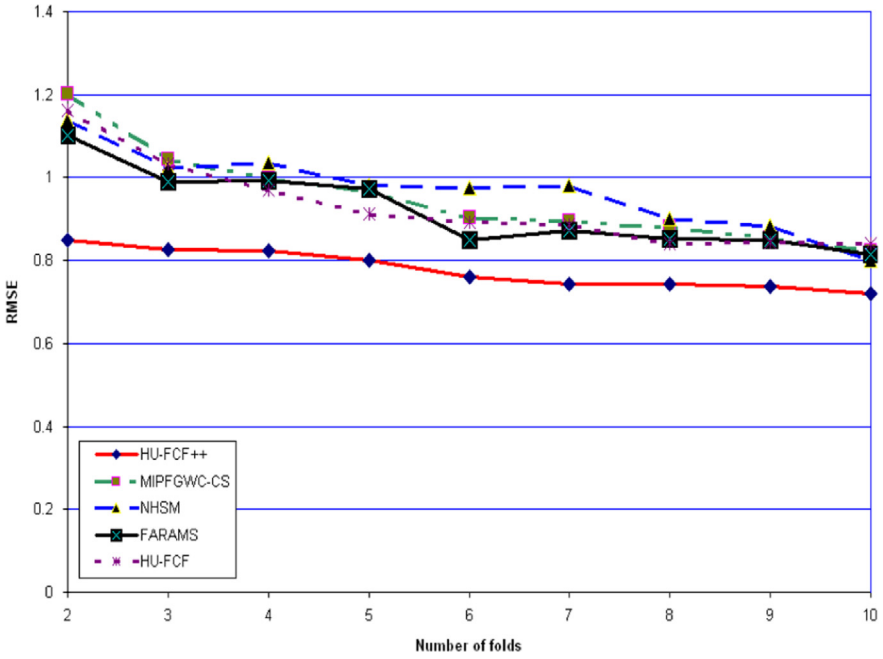

Fig. 4. The *RMSE* values of algorithms on the MovieLens dataset

MIPFGWC and ARM would create good approximation of ratings and then these values are supplemented to the Complete Rating Data, which are the basis to make the prediction in Phase II of the algorithm. The other relevant algorithms depended on a small part of Phase I or Phase II and were not equipped with some special procedures like HU-FCF++ so that their accuracies are less efficient than that of the HU-FCF++ algorithm. In Table 21, we validate the accuracies of algorithms on a dataset that supports the rating only like Jester. In this case, the average *MAE* values of HU-FCF++, NHSM, FARAMS and HU-FCF on the Jester dataset are 0.78, 0.802, 0.81 and 0.98, respectively. The average *RMSE* values of HU-FCF++, NHSM, FARAMS and HU-FCF on the Jester dataset are 0.84, 1.02, 1.04 and 1.13, respectively. Clearly, the *MAE* and *RMSE* values of HU-FCF++ are still better than other algorithms even in the case of the demographic is missing. The reason for this fact lies on the Initialization procedure of the HU-FCF++ algorithm. In case of missing the demographic, HU-FCF++ invokes the Initialization procedure to create the Complete Rating Data. For a given item, contrary to the scanning all ratings that may consist of irrelevant and extraordinary ones, the most popular rating evaluated by many users will be selected as the representation of rating of the new user. The advantage of this process is two-folds: Firstly, the accuracy of selection could be high in the mass; secondly, the computational time is reduced since the most popular rating is opted only. This approximation behavior is then applied to the next item until a number of first items in the Complete Rating Data denoted by *MaxPredicts* in Table 4 are reached. Intuitively, we have a good approximation of the first ratings, which are then used to predict the

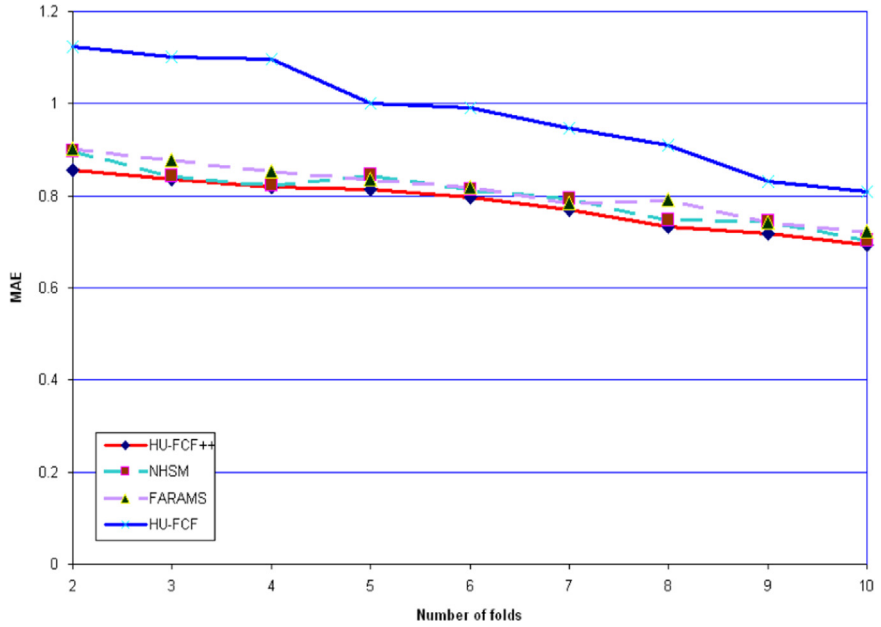

Fig. 5. The *MAE* values of algorithms on the Jester dataset.

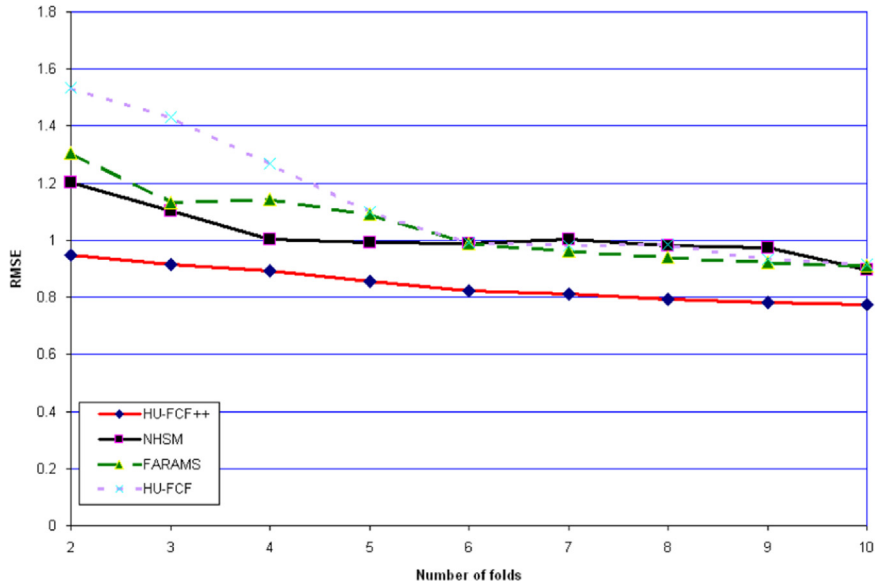

Fig. 6. The *RMSE* values of algorithms on the Jester dataset.

next. In the NHSM algorithm, the scanning-all-rating strategy is applied to only first item in the list so that this may lessen the accuracy of the algorithm. We clearly recognize that using the Initialization procedure, the performance of HU-FCF++ is better than those of NHSM and other algorithms as shown in Table 21. Taking the comparison between the results in Tables 20 and 21, we notice that the accuracy of HU-FCF++ in the dataset having both demographic and rating like MovieLens is better than that in the dataset providing the demographic only. The average *MAE* and *RMSE* values of HU-FCF++ in these tables have confirmed this fact. Thus, a suggestion for the obtaining of high accuracy of the HU-FCF++ algorithm is selecting the appropriate dataset for prediction, which is in this case the mixed dataset of demographic and rating. The figures from Figs. 3–6 demonstrate two remarks: Firstly, HU-FCF++ achieves better accuracy than other algorithms; secondly, this algorithm shows better performance in the MovieLens than in the Jester dataset. A limitation of the HU-FCF++ algorithm is the computational complexity. As being mentioned in Tables 20 and 21, this algorithm takes much time to process a case of experiments, for instance from 342 to 652 s in both datasets. The other algorithms especially NHSM and FARAMS take little processing time, i.e. 16 and 90 s respectively. Thus, we recognize that the proposed algorithm is effective in term of accuracy only. It needs to be enhanced to remedy the problem of computational complexity as shown in the experiments. Through the results in Tables 20 and 21 and from Figs. 3–6, we have the answer for objectives 1 & 3 in Section 3.2.

- The accuracy of HU-FCF++ is better than those of the relevant algorithms especially on the RS data having both the demographic and rating.
- The drawback of HU-FCF++ in comparison with those of other algorithms regarding the computational time should be investigated further.

Table 22
The results of HU-FCF++ by cases of parameters on the MovieLens dataset.

Dataset	*MinSupp=10* *MinConf=0.3* *MaxPredicts=30*	*MinSupp=10* *MinConf=0.3* *MaxPredicts=50*	*MinSupp=10* *MinConf=0.3* *MaxPredicts=70*	*MinSupp=30* *MinConf=0.3* *MaxPredicts=30*	*MinSupp=30* *MinConf=0.3* *MaxPredicts=50*
MAE	0.623	0.612	0.643	0.623	0.642
RMSE	0.829	0.823	0.849	0.623	0.850
	MinSupp=30 *MinConf=0.3* *MaxPredicts=70*	*MinSupp=50* *MinConf=0.3* *MaxPredicts=30*	*MinSupp=50* *MinConf=0.3* *MaxPredicts=50*	*MinSupp=50* *MinConf=0.3* *MaxPredicts=70*	*MinSupp=10* *MinConf=0.5* *MaxPredicts=30*
MAE	0.622	0.622	0.606	0.637	0.624
RMSE	0.828	0.829	0.823	0.849	0.830
	MinSupp=10 *MinConf=0.5* *MaxPredicts=50*	*MinSupp=10* *MinConf=0.5* *MaxPredicts=70*	*MinSupp=30* *MinConf=0.5* *MaxPredicts=30*	*MinSupp=30* *MinConf=0.5* *MaxPredicts=50*	*MinSupp=30* *MinConf=0.5* *MaxPredicts=70*
MAE	0.615	0.664	0.622	0.612	0.660
RMSE	0.835	0.873	0.828	0.830	0.867
	MinSupp=50 *MinConf=0.5* *MaxPredicts=30*	*MinSupp=50* *MinConf=0.5* *MaxPredicts=50*	*MinSupp=50* *MinConf=0.5* *MaxPredicts=70*	*MinSupp=10* *MinConf=0.7* *MaxPredicts=30*	*MinSupp=10* *MinConf=0.7* *MaxPredicts=50*
MAE	0.620	0.607	0.653	0.601	0.576
RMSE	0.826	0.825	0.862	0.809	0.795
	MinSupp=10 *MinConf=0.7* *MaxPredicts=70*	*MinSupp=30* *MinConf=0.7* *MaxPredicts=30*	*MinSupp=30* *MinConf=0.7* *MaxPredicts=50*	*MinSupp=30* *MinConf=0.7* *MaxPredicts=70*	**Average**
MAE	0.611	0.606	0.581	0.614	**0.621**
RMSE	0.837	0.813	0.797	0.838	**0.824**

Table 23
The results of HU-FCF++ by cases of parameters on the Jester dataset.

MaxPredicts (%)	30	50	70	Average
MAE	0.735	0.802	0.724	**0.753**
RMSE	0.863	0.802	0.856	**0.840**

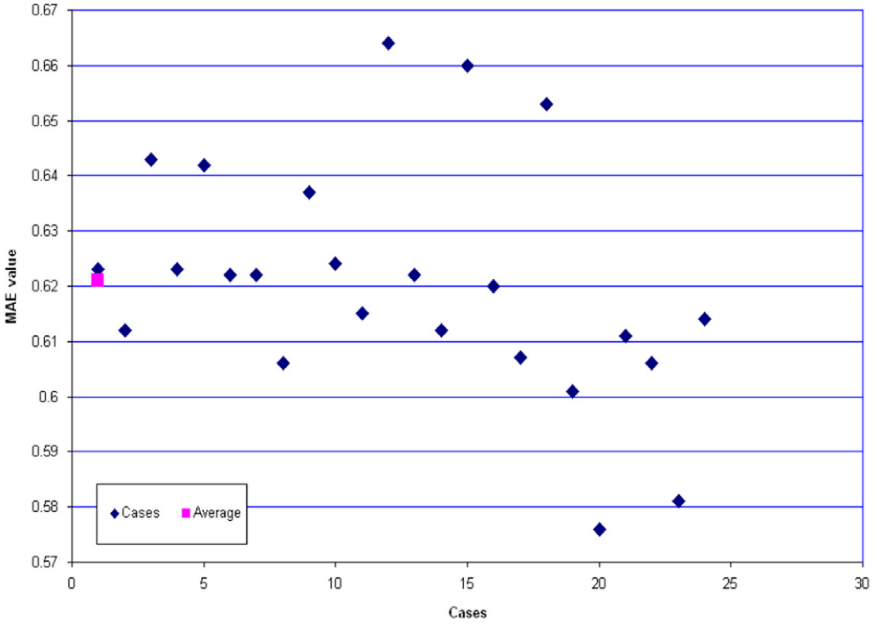

Fig. 7. The *MAE* values of HU-FCF++ by cases on the MovieLens dataset.

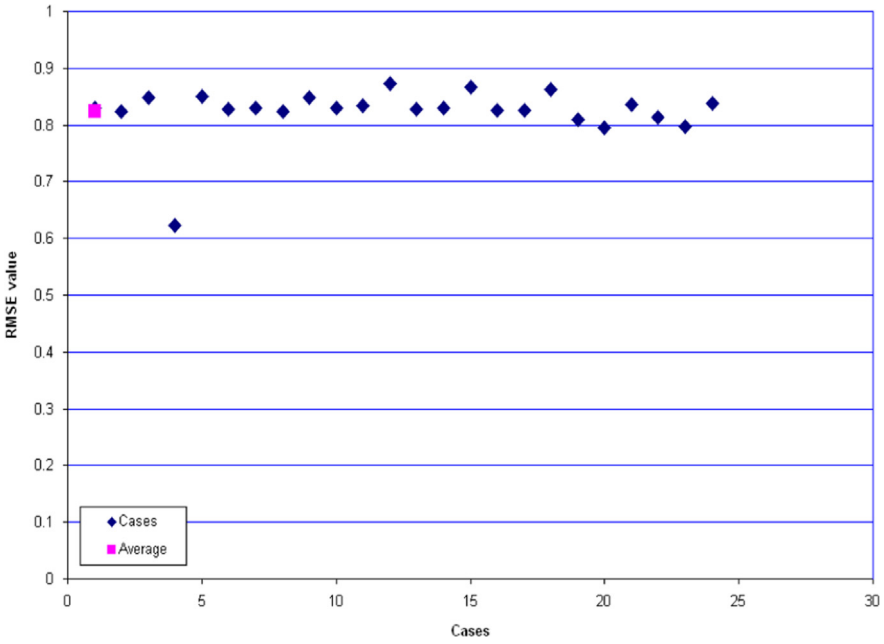

Fig. 8. The *RMSE* values of HU-FCF++ by cases on the MovieLens dataset.

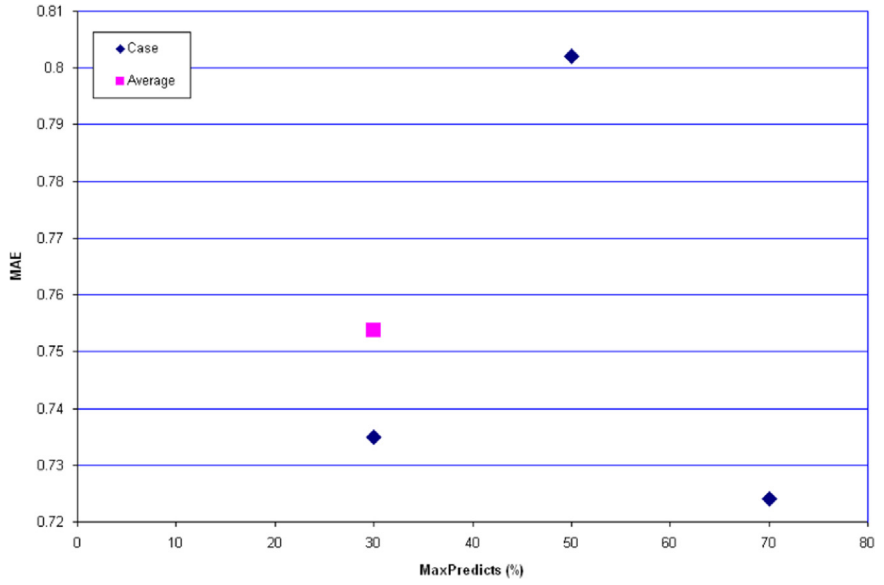

Fig. 9. The *MAE* values of HU-FCF++ by cases on the Jester dataset.

3.4. *The analyses of HU-FCF++ by various datasets and parameters*

In Tables 22 and 23, we have run the HU-FCF++ algorithm by various parameters of the algorithms and by the datasets.

Table 22 describes the experimental results on the MovieLens data, which have both the demographic and the rating so that Phase I and II of the HU-FCF++ algorithm are performed. Phase I requires the *MinSupp* and *MinConf* parameters of the ARM procedure, and Phase II takes the *MaxPredicts* parameter. Thus, the results in Table 22 were taken by various values of those parameters. Similarly, in Table 23 the results are conducted on the *MaxPredicts* parameter only since the Jester data do not have the demographic that is essentially used in Phase I.

The average *MAE* and *RMSE* values of HU-FCF++ on the MovieLens dataset described in Table 22 are 0.621 and 0.824, respectively. They are approximate to the values of other cases of parameters. Similarly, the average *MAE* and *RMSE* values of HU-FCF++ on the Jester dataset described in Table 23 are 0.753 and 0.840, respectively.

In Figs. 7 and 8, we illustrate the *MAE* and *RMSE* values of HU-FCF++ by cases on the MovieLens dataset, respectively. Analogously, Figs. 9 and 10 depict the *MAE* and *RMSE* values of HU-FCF++ by cases on the Jester dataset, respectively. The experimental results have clearly showed that the *MAE* value of HU-FCF++ fluctuates in the interval [0.62, 0.75]. Similarly, the *RMSE* value of HU-FCF++ fluctuates in the interval [0.824, 0.840]. Moreover, the values of HU-FCF++ in different cases are in the small difference and near to the average value. Thus, we have the answer for objectives 2 in Section 3.2 as follows:

- The accuracy of HU-FCF++ is stable by various datasets and parameters.

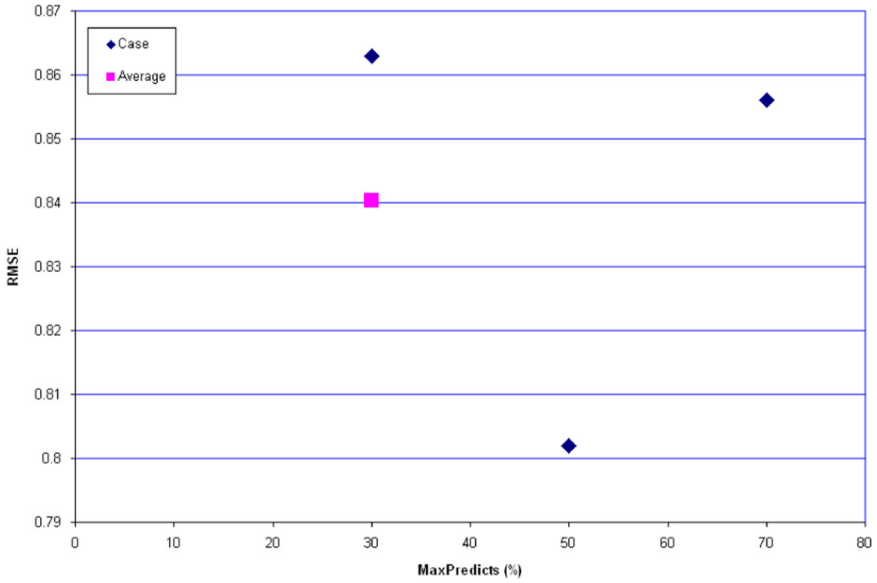

Fig. 10. The *RMSE* values of HU-FCF++ by cases on the Jester dataset.

4. Conclusions

In this paper, we concentrated on the new user cold-start problem that significantly affects negatively the recommender performance due to the inability of the recommender systems to produce meaningful recommendations. A novel hybrid method so-called HU-FCF++ that combines the advantages of different groups of methods for solving the new user cold-start problem was proposed. HU-FCF++ made uses of (i) a pre-processing procedure so-called FACA-DTRS to automatically determine the number of clusters for the finding of similar users; (ii) a fuzzy geographically clustering algorithm namely MIPFGWC to specify the group of analogous users to a new one; (iii) a novel Association Rules Mining (ARM) procedure to find similar items of a given one; (iv) a novel initialization procedure to create pre-ratings for the Complete Rating Data based on the idea of the most popular rating; (v) the NHSM metric to make the prediction from a complete rating dataset. These novel ideas and solutions were packed in the HU-FCF++ method.

As being mentioned in Section 2, the advantages and differences of the proposed work in comparison with the relevant ones are the capability to predict the ratings for a number of items, which is difference to those in other algorithms where only one rating is predicted. Furthermore, HU-FCF++ is proven to be the generalization of the existing approaches as shown in the mechanism in Fig. 1. It could also handle the limitations of previous works described in Section 1. The experimental evaluation on the benchmark recommender systems datasets have shown that (i) the accuracy of HU-FCF++ is better than those of other relevant algorithms; (ii) the accuracy of HU-FCF++ is stable through the variations of datasets and parameters; and (iii) the drawbacks of HU-FCF++ regarding large computational time should be further investigated. A numerical example on a simulated dataset was also presented to demonstrate step-by-step the activities of HU-FCF++. These concluding remarks have clearly pointed out the efficiency of the proposal.

Looking back at Section 1, we recognize the practical implication and insightful of the proposed work to the new user cold-start problem. Therefore, our further researches directions could be lied into the following points: (i) designing parallel mechanisms to reduce the computational costs of HU-FCF++; (ii) improving the association rules to large orders in the ARM procedure; (iii) enhancing the accuracy of HU-FCF++ by considering the simultaneously processing in both Phase I and Phase II; (iv) proposing a general similarity measure that is better than the NHSM metric; (v) investigating applications of HU-FCF++ to the forecasting problems. Those directions will enrich the knowledge of developing hybrid systems and techniques in the fields of recommender systems and applied intelligence in the future.

Acknowledgment

The authors are greatly indebted to the editor-in-chief, Prof. B. Grabot; anonymous reviewers for their comments and their valuable suggestions that improved the quality and clarity of paper. This work is sponsored by the NAFOSTED under Contract no. 102.05-2014.01.

References

Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, vol. 1215, pp. 487–499.

Jester. Jester Online Joke Recommender Dataset. ⟨http://www.ieor.berkeley.edu/~goldberg/jester-data/⟩ (accessed September 2013).

Leung, C.W.K., Chan, S.C.F., Chung, F.L., 2008. An empirical study of a cross-level association rule mining approach to cold-start recommendations. Knowl.-Based Syst. *21* (7), 515–529.

Liu, H., Hu, Z., Mian, A., Tian, H., Zhu, X., 2014. A new user similarity model to improve the accuracy of collaborative filtering. Knowl.-Based Syst. *56*, 156–166.

Manouselis, N., et al., 2012. Recommender systems challenge 2012. In: Proceedings of the 6th ACM Conference on Recommender Systems, pp. 353–354.

MovieLens. Movie Lens dataset. ⟨http://grouplens.org/datasets/movielens/⟩ (accessed September 2013).

Safoury, L., Salah, A., 2013. Exploiting user demographic attributes for solving cold-start problem in recommender system. Lect. Notes Softw. Eng. *1* (3), 303–307.

Shapira, B., 2011. Recommender Systems Handbook. Springer, USA.

Son, L.H., 2014a. Enhancing clustering quality of geo-demographic analysis using context fuzzy clustering type-2 and particle swarm optimization. Appl. Soft Comput. *22*, 566–584.

Son, L.H., 2014b. HU-FCF: a hybrid user-based fuzzy collaborative filtering method in recommender systems. Expert Syst. Appl. *41* (15), 6861–6870.

Son, L.H., 2014c. Optimizing municipal solid waste collection using chaotic particle swarm optimization in GIS based environments: a case study at Danang City, Vietnam. Expert Syst. Appl. *41* (18), 8062–8074.

Son, L.H., 2015. DPFCM: a novel distributed picture fuzzy clustering method on picture fuzzy sets. Expert Syst. Appl. *42* (1), 51–66.

Son L.H., Dealing with the New User Cold-Start Problem in Recommender Systems: A Comparative Review, Information Systems, http://dx.doi.org/10.1016/j.is.2014.10.001.

Son, L.H., Cuong, B.C., Lanzi, P.L., Thong, N.T., 2012a. A novel intuitionistic fuzzy clustering method for geo-demographic analysis. Expert Syst. Appl. *39* (10), 9848–9859.

Son, L.H., Cuong, B.C., Long, H.V., 2013. Spatial interaction-modification model and applications to geo-demographic analysis. Knowl.-Based Syst. *49*, 152–170.

Son, L.H., Lanzi, P.L., Cuong, B.C., Hung, H.A., 2012b. Data mining in GIS: a novel context-based fuzzy geographically weighted clustering algorithm. Int. J. Mach. Learn. Comput. *2* (*3*), 235–238.

Son, L.H., Linh, N.D., Long, H.V., 2014. A lossless DEM compression for fast retrieval method using fuzzy clustering and MANFIS neural network. Eng. Appl. Artif. Intell. *29*, 33–42.

Son, L.H., Minh, N.T.H., Cuong, K.M., Canh, N.V., 2013. An application of fuzzy geographically clustering for solving the cold-start problem in recommender systems. In: Proceeding of 5th IEEE International Conference of Soft Computing and Pattern Recognition (SoCPaR 2013), pp. 44–49.

Son, L.H., Thong, N.T., 2015. Intuitionistic fuzzy recommender systems: an effective tool for medical diagnosis. Knowl.-Based Syst. *74*, 133–150.

Thong, N.T., Son, L.H., 2015. HIFCF: An effective hybrid model between picture fuzzy clustering and intuitionistic fuzzy recommender systems for medical diagnosis. Expert Systems With Applications 42 (7), 3682–3701.

Yu, H., Liu, Z., Wang, G., 2014. An automatic method to determine the number of clusters using decision-theoretic rough set. Int. J. Approx. Reason. *55* (1), 101–115.