Information Sciences 345 (2016) 313–324

Contents lists available at ScienceDirect

Information Sciences

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

Recommending items to group of users using Matrix Factorization based Collaborative Filtering

Fernando Ortega[a,*], Antonio Hernando[b], Jesus Bobadilla[b], Jeon Hyung Kang[c]

[a] Big True Data Research, Madrid, Spain
[b] Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid, Spain
[c] USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292, USA

ARTICLE INFO

Article history:
Received 30 June 2015
Revised 25 January 2016
Accepted 28 January 2016
Available online 4 February 2016

Keywords:
Recommender system
Collaborative Filtering
Matrix Factorization
Group recommender system

ABSTRACT

Group recommender systems are becoming very popular in the social web owing to their ability to provide a set of recommendations to a group of users. Several group recommender systems have been proposed by extending traditional KNN based Collaborative Filtering. In this paper we explain how to perform group recommendations using Matrix Factorization (MF) based Collaborative Filtering (CF). We propose three original approaches to map the group of users to the latent factor space and compare the proposed methods in three different scenarios: when the group size is small, medium and large. We also compare the precision of the proposed methods with state-of-the-art group recommendation systems using KNN based Collaborative Filtering. We analyze group movie ratings on MovieLens and Netflix datasets. Our study demonstrates that the performance of group recommender systems varies depending on the size of the group, and MF based CF is the best option for group recommender systems.

1. Introduction

Recommender Systems (RS) [1] are a type of information filter to overcome the information overload problem. These systems predict the impact that an unknown item will have on a user. The most popular items to which RS have been applied are movies [43]. However, RS have been applied to a wide variety of fields: music [47], TV [34], e-commerce [40], tourism [21,39], social news [25,29], etc. In recent years e-learning RS have also become very popular [11,41,42,45].

RS have always been focused on computing a set of recommendations to an individual user. Nevertheless, there are certain scenarios in which recommending a set of items to a group of several users is more appropriate than providing several sets of recommendations to each individual user of the group (e.g. recommending a vacation destination to a family or recommending a movie to a group of friends). Group Recommender Systems (GRS) aim to provide a set of recommendations that satisfies the preferences of all users in a group.

RS can be classified according to how the recommendations are computed [8] and Collaborative Filtering (CF) based recommendations show promising results by providing more accurate prediction. In order to compute recommendations, CF uses a wide set of ratings made by hundreds of thousands of users on the required items. CF systems can be classified as follows: (a) KNN based CF [5,9,10], which computes the recommendations using the k most similar users to the target user in

* Corresponding author. Tel.: +34 913365104.
E-mail addresses: fernando@bigtruedata.com (F. Ortega), ahernando@etsisi.upm.es (A. Hernando), jesus.bobadilla@upm.es (J. Bobadilla), jeonhyung@google.com (J.H. Kang).

http://dx.doi.org/10.1016/j.ins.2016.01.083

terms of ratings; (b) model based CF [6,20,46], which uses the rating matrix to build a model to generate recommendations; (c) hybrid CF [28,37,38,44], which is a combination of the previous types. The most popular model applied to CF is Matrix Factorization (MF) [27,32], which provides a decomposition of the rating matrix into two matrices that represent both users and items in a latent factor space.

In this paper we will extend the capabilities of MF based CF by allowing it to provide recommendations to a group of users. The key factor of the proposed method is how to compute the latent factors of the group of users for which the items will be recommended. We propose 3 different approaches to compute latent factors and present the advantages and disadvantages of each one.

The rest of the paper is structured as follows: Section 2 describes the related works on recommending items to a group of users; Section 3 presents the three proposed group recommendation approaches using MF based CF; Section 4 shows experimental results of the proposed methods; and Section 5 contains conclusions and future works.

2. Related works

Increasing relevance of groups of users in the social web has led to a significant expansion of GRS [12,22]. Most recent RS surveys have included important sections to explain the state-of-the-art GRS: [30] classifies GRS based on the type of items that are recommended: text based items (books, documents and webpages), multimedia items (music and movies), or tourism items (attractions, accommodation and restaurants); [8] presents different alternatives to make recommendations to a group of users using KNN based CF. The authors classified them according to the stage in which the data of the users of the group is unified into the data of the group of users. [16] introduces GRS and discusses the challenges of group recommendations.

MF based CF is the most popular algorithm to compute single user recommendations; however, in spite of its popularity and the increasing relevance of GRS, based on our knowledge, there are no methods that attempt to combine both ideas (GRS and MF). None of the most recent surveys [8,16,30] include any references about GRS using MF based RS. The most similar approach is [14], which proposes to modify the MF model to include a wide variety of sociological factors such as cohesion, social similarity and social centrality.

GRS has been used in different areas: tourism [2,17,31], entertainment [13,15,48], web [36], among others. The most popular method to compute recommendations for a group of users is KNN based CF. [3] uses lists of individual recommendations to aggregate recommendations for group individuals into one recommendation list for the group of users. [4] proposes aggregating the users' predictions to build the group's predictions of unknown items. [7] defines the set of neighbors of the group as the intersection of the sets of neighbors of each user of the group. [35] presents a similarity metric to compute the KNN set of the group of users. [18] improves GRS by resolving the data sparsity problem of KNN based CF using a support vector machine learning model that computes similarities between items.

From the algorithmic perspective, many different approaches have been proposed to generate group recommendations. [49] suggests a collaborative methodology for searching, selecting, rating and recommending learning objects using a hybrid RS that combines voting aggregation strategies and meta-learning techniques. [26] focuses on the issue of recommending items to groups that are specifically formed in social media systems. To deal with that issue, they present a stochastic method that makes recommendations based on a link-structure analysis of a graph model profiting from fruitful tagging information. [23] constructs a virtual user representing a group by taking transitive precedence of items of all members into consideration. And, subsequently, the profile of the virtual user has been used to represent the group's profile.

3. Recommendation method

3.1. Method overview

In this paper we present a method to make recommendations to a group of users using MF based CF. The main idea of MF models is to factorize the original rating matrix into two or more matrices that represent user-item interactions. Fig. 1 shows a graphical representation of the factorization process. The rating matrix (left side of the figure) is factorized into two matrices (right side of the figure): the first one represents the users in the latent factor space and the other represents the items.

The key factor of the proposed method is to compute the group's factors representing the group-item interactions in the latent factor space. We define three original approaches to compute these factors. These approaches can be classified according to when the users' data is unified into the group's data: (a) before factorization (BF and WBF); or (b) After Factorization (AF). Fig. 2 summarizes the proposed methods.

- After Factorization (AF): This is the simplest way to compute recommendations to a group of users using an MF model. It computes the group's factors by merging the factors of the users that belong to the group. This approach can be seen as the baseline of the GRS using MF based CF.
- Before Factorization (BF): It models the group of users by building a virtual user representing the item preferences of the users of the group. To compute the group's factors, it uses the folding-in technique on the virtual user to add it to the factorized model.

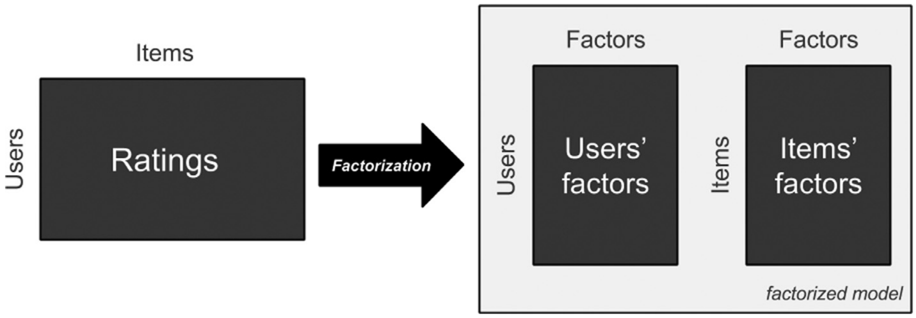

Fig. 1. Graphical example of Matrix Factorization process.

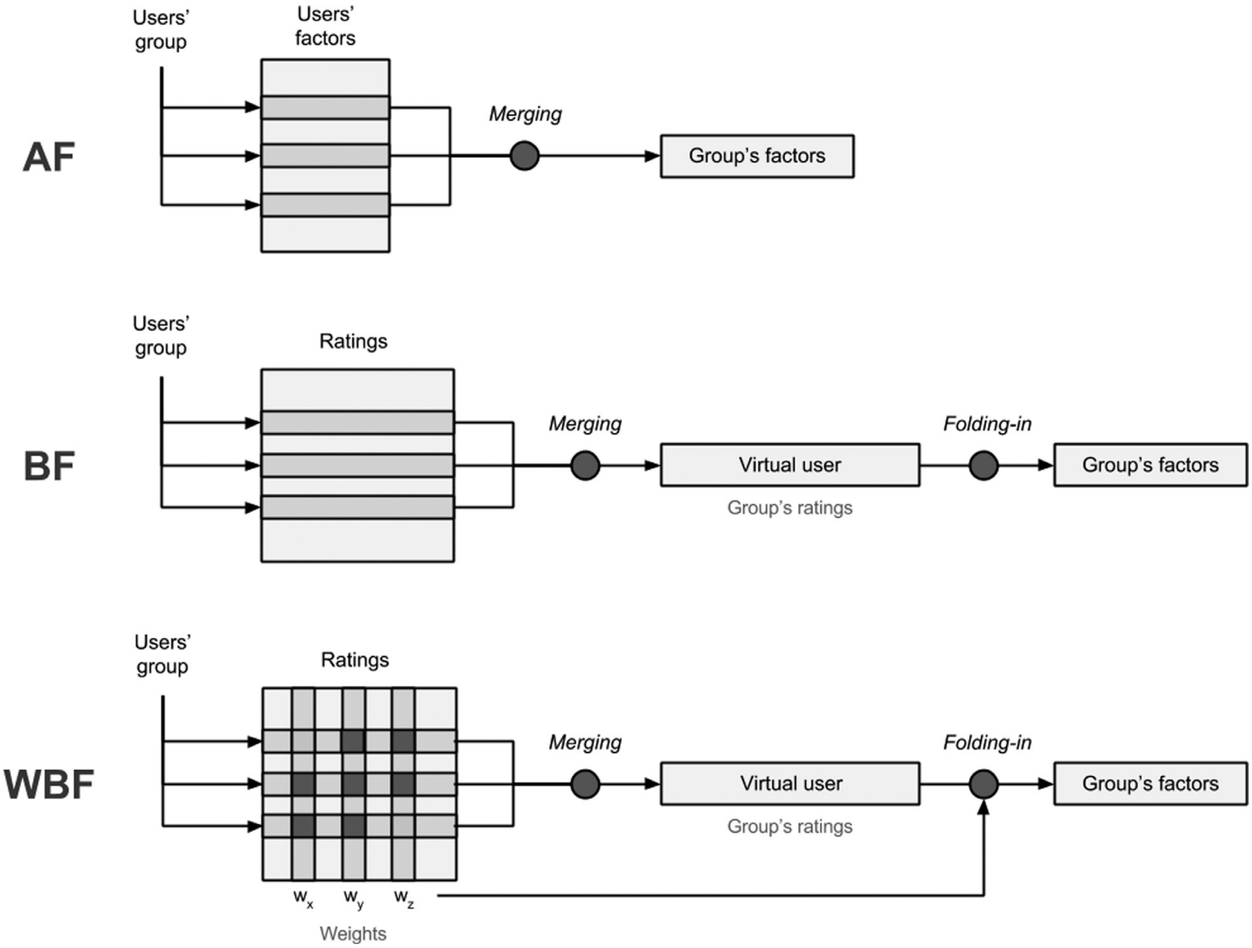

Fig. 2. Matrix Factorization based CF group recommendation approaches.

- Weighted BF (WBF): This is an extension of BF. It adds a weight to each item that the virtual user has 'rated'. These weights will be computed based on the number of ratings that each item has received from the group's users and the consensus of these ratings. The items with the highest weights will contribute more when we compute the group latent factors than those with a low weight.

In the experimental section we will compare these three approaches and evaluate them using the performance on recommendation predictions. Our hypothesis is that the BF and WBF approaches should be more precise than the AF approach as the virtual user is a better representation of the group than the aggregation of the users' factors.

3.2. Matrix Factorization model

MF models map users and items to a joint latent factor space to represent user-item interactions. The model used in this paper [27] factorizes the rating matrix into the following elements:

- $\vec{q}_i = (q_{i,1} \ldots q_{i,K})$ represents the factor vector of the item i.
- b_i represents the bias of the item i independent of any interaction.
- $\vec{p}_u = (p_{u,1} \ldots p_{u,K})$ represents the factor vector of the user u.
- b_u represents the bias of the user u independent of any interaction.

To learn the factor vectors ($\vec{q}_i$ and $\vec{p}_u$) and the biases (b_i and b_u) the system minimizes the following expression for a set of known ratings:

$$\min_{\vec{p}_u,\vec{q}_i,b_u,b_i} \sum_{r_{u,i} \neq \bullet} (r_{u,i} - \mu - b_u - b_i - \vec{p_u}^{\top}\vec{q}_i)^2 + \lambda(||\vec{p}_u||^2 + ||\vec{q}_i||^2 + b_u^2 + q_i^2) \tag{1}$$

Where $r_{u,\,i}$ is the training rating of the user u to the item i, μ is the rating average of the dataset and λ is a parameter that controls the training process.

Once the MF is learnt, prediction for the user u to the item i ($m_{u,\,i}$) can be computed using the following expression:

$$m_{u,i} = \mu + b_i + b_u + \vec{p_u}^{\top}\vec{q}_i \tag{2}$$

In order to generate group predictions we need to compute the group's factor vector ($\vec{p}_G$) and the group's bias (b_G). We will describe how to compute these values in the following sections. Once the group is factorized, prediction for the group G to the item i can be computed as follows:

$$m_{G,i} = \mu + b_i + b_G + \vec{p_G}^{\top}\vec{q}_i \tag{3}$$

Using the predicted values we compute the recommendations for the group G (R_G) as the set of N items (4) not rated for any user of the group (5) and with the prediction values (6).

The following expressions must be true:

$$\#R_G \leq N \tag{4}$$

$$\forall i \in R_G, \forall u \in G : r_{u,i} = \bullet \tag{5}$$

$$\forall i \in R_G, \forall j \notin R_G : m_{G,i} \geq m_{G,j} \tag{6}$$

3.3. After factorization approach

The After Factorization (AF) approach factorizes the group of users by merging factors of each user belonging to the group. This approach only uses the information generated after factorization, not the rating information, so the users are unified into the group when the MF model is built.

Let $G = \{u_1 \ldots u_n\}$ be the set of users that belongs to the group G, let $\vec{p}_u = (p_{u,1} \ldots p_{u,K})$ be the factor vector of the user u and let b_u be the bias of the user u.

We define $\vec{p}_G$ as the factor vector of the group G:

$$\vec{p}_G = \begin{pmatrix} h(p_{u_1,1}, \cdots, p_{u_n,1}) \\ \vdots \\ h(p_{u_1,K}, \cdots, p_{u_n,K}) \end{pmatrix} \tag{7}$$

We define b_G as the bias of the group G:

$$b_G = h(b_{u_1}, \cdots, b_{u_n}) \tag{8}$$

Where $h(x_1, \ldots, x_n)$ is an aggregation function that combines the $x_1 \ldots x_n$ values into one representative value (i.e.: average, maximum, etc).

3.4. Before factorization approach

The Before Factorization (BF) approach is based on the idea of representing the preferences of the group of users, $G = \{u_1, \ldots, u_n\}$, by means of a virtual user, u_G, aggregating the ratings of all the users in the group G. The aggregation is performed with the information we have before the MF process (the ratings). This approach is based on two steps:

- Simulating the ratings that the virtual user u_G makes on the items, $r_{G,\,i}$ (Fig. 2, BF "merging"). This step is performed by means of a specific aggregation function h in the following way:

  $$r_{G,i} = h(r_{u_1,i}, r_{u_2,i}, \ldots, r_{u_n,i}) \tag{9}$$

  where $r_{u_1,i}, \ldots, r_{u_n,i}$ are the observed ratings of the users of the group G to the item i.

- Calculating the virtual user factors vector ($\vec{p}_G = (p_{G,1}, \ldots, p_{G,K})$) and the virtual user bias (b_G) once the ratings ($r_{G,\,i}$) are defined (Fig. 2, BF "folding-in"). This step is achieved by solving the expression (10) where $\vec{q}_i$, b_i and μ are set to the values previously calculated in the learning phase (Section 3.2):

$$\min_{\vec{p}_G, b_G} \sum_{r_{u_G,i} \neq \bullet} (r_{G,i} - \mu - b_G - b_i - \vec{p_G}^{\top} \vec{q}_i)^2 + \lambda(||\vec{p}_{u_G}||^2 + ||\vec{q}_i||^2 + b_{u_G}^2 + q_i^2) \tag{10}$$

The previous expression can be simplified:

$$\min_{\vec{p}_G, b_G} \sum_{r_{G,i} \neq \bullet} \left((r_{G,i} - \mu - b_i) - (\vec{p_G}^{\top} \vec{q}_i + b_G)\right)^2 + \lambda(||\vec{p}_G||^2 + b_G^2) \tag{11}$$

We define:
- $s_{G,i} = r_{G,i} - \mu - b_i$.
- $\vec{p_G}^* = (\vec{p}_G, b_G) = (p_{G,1}, \ldots, p_{G,K}, b_G)$ extends the vector $\vec{p}_G$.
- $\vec{q}_i^* = (\vec{q}_i, 1) = (q_{i,1}, \ldots, q_{i,K}, 1)$ extends the vector $\vec{q}_i$.

According to the definition, the previous expression is derived as follows:

$$\min_{\vec{p_G}^*} \sum_{s_{G,i} \neq \bullet} (s_{G,i} - \vec{p_G}^{*\top} \vec{q}_i^*)^2 + \lambda||\vec{p_G}^*||^2 \tag{12}$$

As we can see, this minimization corresponds to the definition of ridge regression. In order to simplify the notation, we will consider that the virtual user u_G has rated the items $\{1, \ldots n_G\}$, and we define the matrix A:

$$A = \begin{pmatrix} q_{1,1} & \ldots & q_{1,K} & 1 \\ q_{2,1} & \ldots & q_{2,K} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ q_{n_G,1} & \ldots & q_{n_G,K} & 1 \end{pmatrix} \tag{13}$$

We have that:

$$\begin{pmatrix} \vec{p}_G \\ b_G \end{pmatrix} = \begin{pmatrix} p_{G,1} \\ p_{G,2} \\ \vdots \\ p_{G,K} \\ b_G \end{pmatrix} = (A^T A + \lambda I)^{-1} A^T \begin{pmatrix} s_{G,1} \\ s_{G,2} \\ \vdots \\ s_{G,n_G} \end{pmatrix} \tag{14}$$

3.5. Weighted before factorization approach

Unlike the previous BF approach, the Weighted Before Factorization (WBF) approach will weight each item according to the ratings of the users of the group. Here we will give more importance to those items that: (a) have been most rated by the users and (b) have similar ratings to the users of the group. We define $w_{G,i}$ as the weight of the item i for the group G as :

$$w_{G,i} = \frac{\#\{u \in G | r_{u,i} \neq \bullet\}}{\#G} \cdot \frac{1}{1 + \sigma_{G,i}} \tag{15}$$

where $\sigma_{G,\,i}$ is the standard deviation of the group G members' ratings for the item i and # denotes the cardinality of a set.

This approach only differs from the BF approach in the second step where we calculate the virtual user vector ($\vec{p}_G$) and the virtual user bias (b_G). In this case, our objective function is:

$$\min_{\vec{p}_G, b_G} \sum_{r_{G,i} \neq \bullet} w_{G,i}(r_{G,i} - \mu - b_G - b_i - \vec{p_G}^{\top} \vec{q}_i)^2 + \lambda(||\vec{p}_{u_G}||^2 + ||\vec{q}_i||^2 + b_{u_G}^2 + q_i^2) \tag{16}$$

Following the same reasoning used in the previous section, we conclude that this expression is equivalent to a weighted ridge regression. Consequently, we have that:

$$\begin{pmatrix} \vec{p}_G \\ b_G \end{pmatrix} = \begin{pmatrix} p_{G,1} \\ p_{G,2} \\ \vdots \\ p_{G,K} \\ b_G \end{pmatrix} = (A^T W A + \lambda I)^{-1} A^T W \begin{pmatrix} s_{G,1} \\ s_{G,2} \\ \vdots \\ s_{G,n_G} \end{pmatrix} \tag{17}$$

Table 1
Rating matrix of the example RS with 8 users and 10 items.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
U_1	5	3		1		2	5		3	
U_2		2		4		2		5		
U_3	5		3	1		3				5
U_4	4				2	2		5		4
U_5		2			3		4	4	4	
U_6			3	1			4			5
U_7	5		2		2			4	3	
U_8		3		2			5		4	5

Table 2
Users' factors and bias.

$p_{u,\,k}$	F_1	F_2	F_3	b_u
U_1	−0.988814	0.582736	0.417410	−0.044211
U_2	0.765410	0.710435	−0.395533	0.102630
U_3	−0.718895	−0.450749	0.328001	0.004569
U_4	0.113950	0.587010	−0.680070	0.000034
U_5	0.513911	−0.012850	0.283797	−0.011456
U_6	−0.753587	−0.419765	0.264213	−0.019968
U_7	−0.727785	−0.428044	−0.603294	−0.214561
U_8	−0.321865	0.126082	0.573243	0.409063

Table 3
Items's factors and bias.

$q_{u,\,k}$	F_1	F_2	F_3	b_i
I_1	−0.993608	−0.261176	−0.130855	0.856596
I_2	−0.664219	−0.089604	−0.092027	−0.902178
I_3	0.175229	0.409434	0.762585	−0.300954
I_4	1.252735	0.409759	−0.625687	−1.011783
I_5	0.227013	−0.299668	0.681047	−0.678858
I_6	0.003825	−0.838561	0.080504	−0.791824
I_7	−0.303405	0.701578	0.606703	0.636074
I_8	0.106912	0.553922	−0.624930	0.792658
I_9	0.534776	−0.168291	0.170963	0.287533
$I_1 0$	−0.669308	−0.249170	0.152452	0.953500

where W is the diagonal matrix:

$$W = \begin{pmatrix} w_{G,1} & 0 & \dots & 0 \\ 0 & w_{G,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{G,n_G} \end{pmatrix} \tag{18}$$

3.6. Running example

The following running example aims to clarify how the proposed approaches compute the group's factors.

We defined an RS with 8 users and 10 items. Table 1 contains the rating matrix. The users can rate the items on a scale of 1 (don't like) to 5 (love it).

We computed our factors using a derived algorithm and decomposed the rating matrix into two matrices. We applied 3 factors ($K = 3$) and $lambda = 0.03$ for this experiment. Table 2 contains the users' factors and the users' bias. Table 3 contains the items' factors and the items' bias.

We defined a group G of three users: $G = \{U_1, U_2, U_3\}$.

To compute the group's factors using the AF approach we must aggregate the factors of each user in the group. Eq. (19) shows an example of how to compute F_1 using the average as h function. Table 4 contains the factors and the bias of the group G with the AF approach.

$$p_{G,1} = \frac{p_{U_1,1} + p_{U_2,1} + p_{U_3,1}}{3} = -0.3140996667 \tag{19}$$

Table 4
Group's factors and bias using AF approach.

$p_{G,\,1}$	$p_{G,\,2}$	$p_{G,\,3}$	b_G
−0.3140996667	0.2808073333	0.116626	0.020996

Table 5
Virtual user's ratings using average as aggregation function.

$r_{G,\,i}$	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
G	5	2,5	3	2		2,33	5	5	3	5

Table 6
Group's factors and bias using BF approach.

$p_{G,\,1}$	$p_{G,\,2}$	$p_{G,\,3}$	b_G
−0.7705556797	0.7431251244	−0.2790598325	0.1535254538

To represent the group's factors using the BF approach we compute the factors of virtual user for the group of users. Table 5 contains the virtual user's ratings using the average as h function.

To compute the group's factors we have to solve Eq. (14). Eq. (20) contains the A matrix for the virtual user of the group G. Eq. (21) shows how to compute the first element of the column vector. Once Eq. (14) has been solved we obtain the factors and bias of the group of users. Table 6 contains the factors and the bias of group G if the BF approach is used.

$$A = \begin{pmatrix} q_{I_1,F_1} & q_{I_1,F_2} & q_{I_1,F_3} & 1 \\ q_{I_2,F_1} & q_{I_2,F_2} & q_{I_2,F_3} & 1 \\ q_{I_3,F_1} & q_{I_3,F_2} & q_{I_3,F_3} & 1 \\ q_{I_4,F_1} & q_{I_4,F_2} & q_{I_4,F_3} & 1 \\ q_{I_6,F_1} & q_{I_6,F_2} & q_{I_6,F_3} & 1 \\ q_{I_7,F_1} & q_{I_7,F_2} & q_{I_7,F_3} & 1 \\ q_{I_8,F_1} & q_{I_8,F_2} & q_{I_8,F_3} & 1 \\ q_{I_9,F_1} & q_{I_9,F_2} & q_{I_9,F_3} & 1 \\ q_{I_{10},F_1} & q_{I_{10},F_2} & q_{I_{10},F_3} & 1 \end{pmatrix} = \begin{pmatrix} -0.993608 & -0.261176 & -0.130855 & 1 \\ -0.664219 & -0.089604 & -0.092027 & 1 \\ 0.175229 & 0.409434 & 0.762585 & 1 \\ 1.252735 & 0.409759 & -0.625687 & 1 \\ 0.003825 & -0.838561 & 0.080504 & 1 \\ -0.303405 & 0.701578 & 0.606703 & 1 \\ 0.106912 & 0.553922 & -0.624930 & 1 \\ 0.534776 & -0.168291 & 0.170963 & 1 \\ -0.669308 & -0.249170 & 0.152452 & 1 \end{pmatrix} \tag{20}$$

$$s_{G,I_1} = r_{G,I_1} - \mu - b_{I_1} = 0,78 \tag{21}$$

To compute the group's factors using the WBF approach we must define the matrix W containing the weights of each item rated by the virtual user. Eq. (22) contains the W matrix used in this example. To simplify the example, we defined the percentage of users of the group that have rated the item as a weight.

$$W = \begin{pmatrix} 2/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 \end{pmatrix} \tag{22}$$

To compute the group's factors and bias we have to solve Eq. (17). Table 7 contains these factors and bias.

Table 7
Group's factors and bias using WBF approach.

$p_{G,\,1}$	$p_{G,\,2}$	$p_{G,\,3}$	b_G
−1.2977996234	0.9610452568	−0.0417925619	−0.1384674549

Table 8
Parameters used to factorize MovieLens and Netflix databases.

Parameter	MovieLens	Netflix
K (number of factors)	15	30
λ	0.055	0.025
Max. iterations	500	500

4. Evaluation

4.1. Experimental setup

All experiments were carried out using both the MovieLens [19] and Netflix [33] databases. The MovieLens database contains 1,000,209 anonymous ratings of 3,706 movies made by 6,040 users who joined MovieLens in 2000. The Netflix database contains 100,480,507 anonymous ratings of 17,770 movies made by 480,189 users.

We estimate the latent factor of the MF model to use a learned model for recommendation predictions. We use the data to estimate the MF parameters and select the one providing the best results. Table 8 shows the parameters used to factorize both datasets.

The proposed recommendation approaches are capable of working with different h function implementations. We tried each approach with six classical aggregation functions: average, weighted average, mode, median, least misery and most pleasure. For the AF approach the best result is achieved using weighted average (the number of ratings of each user has been used as a weight) for MovieLens and median for Netflix. For the BF and WBF approaches the best results are achieved using least misery in all cases.

Neither MovieLens nor Netflix datasets contain information about how the users are grouped, so we generated around 1,500 groups of users fixing their sizes from 2 to 12 users. These groups of users were generated randomly, but those users who have more training ratings in common have a higher probability of being grouped together [24]. We selected 30% of group ratings as test ratings and these ratings were not used during the learning phase.

We performed two main experiments. The first one checks which of the three proposed approaches reports better recommendation quality. The second one compares the proposed method with other alternatives. There are no previous publications about GRS using MF based CF, so we compared our algorithm with the most recent and accurate GRS using KNN based CF [35]. In each experiment we generated the set of recommendations (we tried from 10 to 30 recommendations) for each group of users and we computed their precision and recall (Section 4.2). We averaged the evaluation metrics values based on the group size. We set three different types of groups: small groups (2 to 4 users), mid-size groups (5 to 8 users) and large groups (9 to 12 users). Obviously it is easier to make a valuable recommendation for a small group than for a larger one. We aim to analyze the impact of the group size on the quality of recommendations.

To avoid fluctuations generated by the random selection of the groups and the random initialization of the latent factors of the MF model we repeated each experiment 50 times and averaged the results.

4.2. Evaluation metrics

In order to check the quality of recommendations computed for the group of users, we define precision and recall for the group G as:

$$precision_G = \frac{\#TP_G}{\#(TP_G \cup FP_G)} \tag{23}$$

$$recall_G = \frac{\#TP_G}{\#T_G} \tag{24}$$

Where TP_G, FP_G and T_G denote the true positive, false positive and expected recommendations sets respectively:

$$FP_G = \{i \in R_G | \exists g \in G \text{ such that } \hat{r}_{g,i} < \theta\} \tag{25}$$

$$TP_G = \{i \in R_G | \exists g \in G \text{ such that } r_{g,i} \neq \bullet \text{ and } \forall u \in G\ \hat{r}_{u,i} \neq \bullet \rightarrow \hat{r}_{u,i} \geq \theta\} \tag{26}$$

$$T_G = \{i \in I | \exists g \in G \text{ such that } r_{g,i} \neq \bullet \text{ and } \forall u \in G\ \hat{r}_{u,i} \neq \bullet \rightarrow \hat{r}_{u,i} \geq \theta\} \tag{27}$$

where R_G represents the set of items recommended to the group G, $\hat{r}_{u,i}$ denotes the test-rating of the user u to the item i (the test ratings are not used during the recommendations computation phase), and θ is a threshold to measure whether a user likes or dislikes an item (our test databases have a rating scale from 1 star to 5 stars, so we have used $\theta = 4$ as threshold).

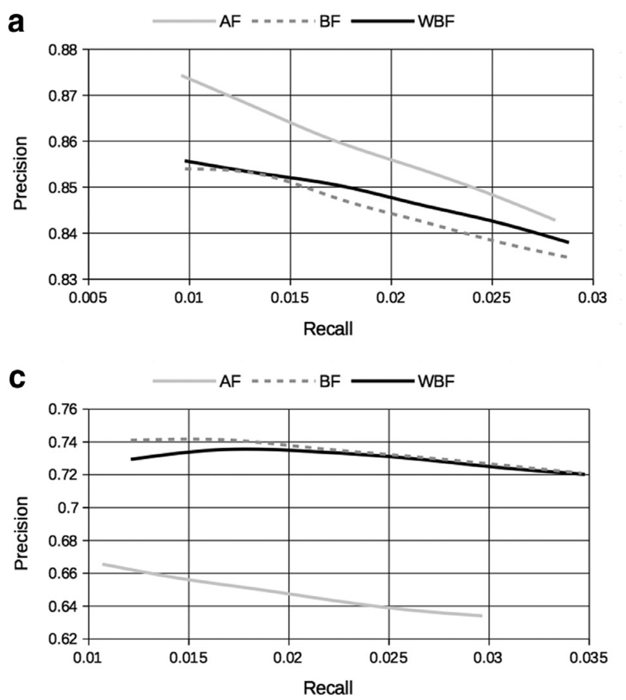

Fig. 3. MF based recommendation approaches comparison using MovieLens. Graphs are classified based on the group size: (a) small group, (b) mid size group, and (c) large group.

Table 9
Precision comparison between proposed method and best memory based CF method.

Group size	2–4		5–8		9–12	
Recommendation method	KNN	MF	KNN	MF	KNN	MF
MovieLens	0.61	0.84	0.54	0.77	0.49	0.73
Netflix	0.61	0.88	0.59	0.78	0.58	0.73

4.3. Experimental results

Fig. 3 contains a comparative of precision and recall values for the three proposed MF based recommendation approaches using the MovieLens dataset. Graph (a) contains the comparison for small groups (2–4 users), Graph (b) for mid size groups (5–8 users), and Graph (c) for large groups (9–12 users).

Fig. 4 contains a comparative of precision and recall values for three proposed MF based recommendations approaches using Netflix dataset. Graph (a) contains the comparison for small groups (2–4 users), Graph (b) for mid size groups (5–8 users), and Graph (c) for large groups (9–12 users).

For small groups (2–4 users) MovieLens obtains better recommendations when AF is used and Netflix obtains better recommendations when BF or WBF (they obtained the same precision) are used. Even though AF is a simplistic and trivial solution, it can make good recommendations when the dataset and the group of users are small. However, when more data are used (like with the Netflix dataset), the AF approach does not work properly (the aggregation of the users' factors is not a good representation of the group preferences) and BF and WBF make better recommendations.

For mid size groups (5–8 users) BF and WBF are very superior to AF. When the group size grows more data are involved in the group recommendation process and the virtual user is a better representation of the group of users than the aggregation of the users' factors. Both BF and WBF obtain approximately the same precision in the two databases tested. Nevertheless WBF is slightly better in MovieLens and BF is slightly better in Netflix.

For large groups (9–12 users) BF is the best recommendation approach in all scenarios studied. BF is better than WBF because it is harder to find items in which all the group's users' ratings are similar when there are too many users in the group. Therefore, the weights obtained when large groups are used are not very representative (all the rated items have approximately the same weight) and the weights have a negative effect on the recommendations.

Table 9 compares the precision of the method (MF) proposed in this paper (we selected the best approach for each group size) with the precision of the best KNN based CF recommendation method proposed by [35] (UGSM). The precision has

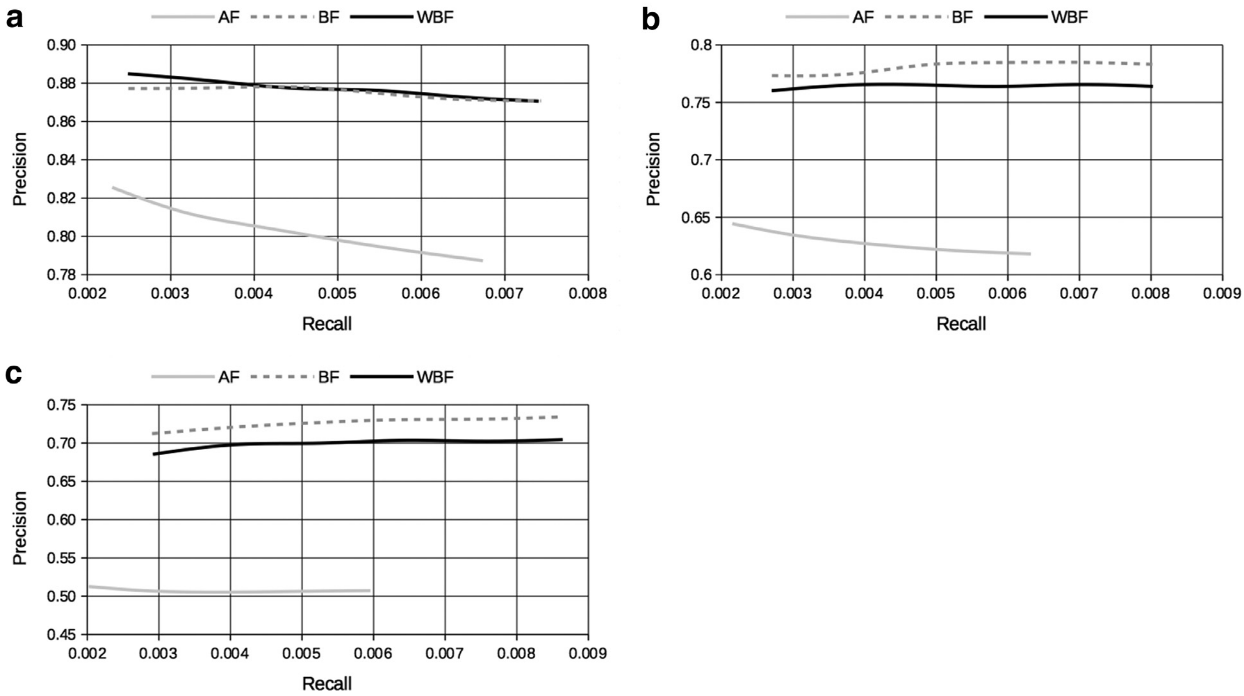

Fig. 4. MF based recommendation approaches comparison using Netflix. Graphs are classified based on the group size: (a) small group, (b) mid size group, and (c) large group.

Table 10
Algorithm's complexity comparison.

	AF	BF	WBF	UGSM
Complexity	$O(GK)$	$O(max(K^2I_G, K^3))$	$O(max(K^2I_G, K^3))$	$O(UI_GG)$

been computed for 20 recommendations. For KNN we used 200 and 300 neighbors for MovieLens and Netflix respectively. The method presented in this paper hugely increases the quality of the recommendations in the three scenarios studied (the new method is around 125% to 150% better in all cases).

4.4. Computational complexity

The complexity of the tested algorithms can be seen in the Table 10. Where K denotes the number of factors of the MF model, G represents the group size, I_G is the number of items rated by at least one user of the group, and U is the number of users of the entire RS.

The computational complexity of the BF and WBF approaches is $O(max(K^2I_G, K^3))$, but, I_G will commonly be larger than K, so the computational complexity of both approaches is $O(K^2I_G)$. Furthermore, BF and WBF have the same complexity because the only difference between both approaches is the matrix W of the WBF approach. This matrix is a diagonal matrix, so it has no effect on the algorithm's computational complexity.

The least complex method is AF followed by BF/WBF and finally UGSM. However, although AF is faster than BF/WBF, the sizes of K (between 10 and 50 factors), G (between 2 and 15 users) and I_G (between 50 and 500 items) are so small that, in a real RS, both methods are equally fast. On the contrary, UGSM is much slower than the MF based methods because the size of U is usually huge (between 10,000 and 1,000,000 users).

5. Conclusions and future work

In this paper we proposed three MF based CF approaches for group recommendations. We tested each approach for different group sizes and compared them with the traditional KNN based CF. Experimental results show that each proposed approach is the most accurate in a different scenario. If the dataset is small and the groups are small the best recommendation approach is AF due to its simplicity and good precision. If the groups are mid size or the dataset is large, we should use the WBF approach. Finally, if the dataset is huge or the groups are very large, BF is the best recommendation method.

In a real GRS we need to test which of the proposed approaches best suits the features of our GRS and choose one of them (we can also select different recommendation approaches according to the type of recommendations that the users request).

In the comparison between the traditional KNN based CF and the proposed MF based CF for groups of users, we can observe a significant increase in the quality of recommendations. MF based CF has reported better recommendation quality than KNN based CF when single-user recommendations were computed, however, in this paper, we also proved that MF based CF is the best option when we want to make recommendations to a group of users.

The proposed approach has an additional advantage: It can be computed using the factorization information without updating the factorization process. Therefore, the proposed approach can be easily incorporated to each MF based RS implementation.

As future work we propose to modify the computation of the items' weights in the WBF approach. In this paper, we defined the weights as a combination of the number of ratings that the item has received and the standard deviation of those ratings, however different weight definitions can be formulated. On the other hand, we can also include a weight system for how the group users represent the social structure of the real groups of users, in which some people have more influence in the group than others. Finally, we can test if the proposed approaches work properly with different factorization algorithms from the one used in this paper.

Acknowledgment

This research was supported by the Spanish Ministerio de Economía y Competitividad TIN2012-32682 project.

To Kristina Lerman's research team (Information Sciences Institute, University of Southern California) for their feedback to improve this work.

Appendix. MF parameters inference

We have used a stochastic gradient descent optimization (expression (1)) to inference the parameters of the MF model. These parameters has been initialized randomly with values in the range $[-1, 1]$. We repeat the following algorithm until the parameters converge: for each known rating of the training set, the error $e_{u,\,i}$ between the real rating and the predicted one is computed:

$$e_{u,i} = r_{u,i} - \mu - b_u - b_i - \vec{p_u}^{\top}\vec{q_i} \tag{28}$$

Then the parameters are modified in the opposite direction of the gradient:

$$\vec{p}_u \leftarrow \vec{p}_u + \gamma \cdot (e_{u,i} \cdot \vec{q}_i - \lambda \cdot \vec{p}_u) \tag{29}$$

$$\vec{q}_i \leftarrow \vec{q}_i + \gamma \cdot (e_{u,i} \cdot \vec{p}_u - \lambda \cdot \vec{q}_i) \tag{30}$$

$$b_u \leftarrow b_u + \gamma \cdot (e_{u,i} - \lambda \cdot b_u) \tag{31}$$

$$b_i \leftarrow b_i + \gamma \cdot (e_{u,i} - \lambda \cdot b_i) \tag{32}$$

where γ is a parameter that controls the speed of the learning process.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.

[2] L. Ardissono, A. Goy, G. Petrone, M. Segnan, P. Torasso, Intrigue: personalized recommendation of tourist attractions for desktop and hand held devices, Appl. Artif. Intelligence 17 (8–9) (2003) 687–714.

[3] L. Baltrunas, T. Makcinskas, F. Ricci, Group recommendations with rank aggregation and collaborative filtering, in: Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, 2010, pp. 119–126.

[4] S. Berkovsky, J. Freyne, Group-based recipe recommendations: analysis of data aggregation strategies, in: Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, 2010, pp. 111–118.

[5] J. Bobadilla, F. Ortega, A. Hernando, A collaborative filtering similarity measure based on singularities, Inf. Process. Manag. 48 (2) (2012) 204–217.

[6] J. Bobadilla, F. Ortega, A. Hernando, J. Alcalá, Improving collaborative filtering recommender system results and performance using genetic algorithms, Knowl. Based Syst. 24 (8) (2011) 1310–1316.

[7] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, Generalization of recommender systems: collaborative filtering extended to groups of users and restricted to groups of items, Expert Syst. Appl. 39 (1) (2012) 172–186.

[8] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl. Based Syst. 46 (2013) 109–132.

[9] J. Bobadilla, F. Ortega, A. Hernando, G. Glez-de Rivera, A similarity metric designed to speed up, using hardware, the recommender systems k-nearest neighbors algorithm, Knowl. Based Syst. 51 (2013) 27–34.

[10] J. Bobadilla, F. Serradilla, J. Bernal, A new collaborative filtering metric that improves the behavior of recommender systems, Knowl. Based Syst. 23 (6) (2010) 520–528.

[11] J. Bobadilla, F. Serradilla, A. Hernando, et al., Collaborative filtering adapted to recommender systems of e-learning, Knowl. Based Syst. 22 (4) (2009) 261–265.

[12] L. Boratto, S. Carta, State-of-the-art in group recommendation and new approaches for automatic identification of groups, in: Proceedings of the Information Retrieval and Mining in Distributed Environments, Springer, 2011, pp. 1–20.

[13] D.L. Chao, J. Balthrop, S. Forrest, Adaptive radio: achieving consensus using negative preferences, in: Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work, ACM, 2005, pp. 120–123.
[14] I. Christensen, S. Schiaffino, Matrix factorization in social group recommender systems, in: Proceedings of the 12th Mexican International Conference on Artificial Intelligence (MICAI), IEEE, 2013, pp. 10–16.
[15] I.A. Christensen, S. Schiaffino, Entertainment recommender systems for group of users, Expert Syst. Appl. 38 (11) (2011) 14127–14135.
[16] M.D. Ekstrand, J.T. Riedl, J.A. Konstan, Collaborative filtering recommender systems, Found. Trends Human Comput. Interact. 4 (2) (2011) 81–173.
[17] I. Garcia, L. Sebastia, E. Onaindia, On the design of individual and group recommender systems for tourism, Expert Syst. Appl. 38 (6) (2011) 7683–7692.
[18] S. Ghazarian, M. Nematbakhsh, Enhancing memory-based collaborative filtering for group recommender systems, Expert Syst. Appl. 42 (7) (2014) 3801–3812.
[19] GroupLens, Movielens dataset. (http://grouplens.org/datasets/movielens/), (accessed: 08.04.2015). (2000).
[20] T. Hofmann, Latent semantic models for collaborative filtering, ACM Trans. Inf. Syst. (TOIS) 22 (1) (2004) 89–115.
[21] R.-H. Hwang, Y.-L. Hsueh, Y.-T. Chen, An effective taxi recommender system based on a spatio-temporal factor analysis model, Inf. Sci. 314 (2015) 28–40.
[22] A. Jameson, B. Smyth, Recommendation to groups, in: The Adaptive Web, Springer, 2007, pp. 596–627.
[23] V.R. Kagita, A.K. Pujari, V. Padmanabhan, Virtual user approach for group recommender systems using precedence relations, Inf. Sci. 294 (2015) 15–30.
[24] J.-H. Kang, K. Lerman, Using lists to measure homophily on twitter, in: Proceedings of the AAAI Workshop on Intelligent Techniques for Web Personalization and Recommendation, 2012.
[25] J.-H. Kang, K. Lerman, La-ctr: A limited attention collaborative topic regression for social media, in: Proceedings of the the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.
[26] H.-N. Kim, A. El Saddik, A stochastic approach to group recommendations in social media systems, Inf. Syst. 50 (2014) 76–93.
[27] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer (8) (2009) 30–37.
[28] Y. Li, L. Lu, L. Xuefeng, A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in e-commerce, Expert Syst. Appl. 28 (1) (2005) 67–77.
[29] C. Lin, R. Xie, X. Guan, L. Li, T. Li, Personalized news recommendation via implicit social experts, Inf. Sci. 254 (2014) 1–18.
[30] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: A survey, Decis. Support Syst. 74 (2015) 12–32.
[31] K. McCarthy, M. Salamó, L. Coyle, L. McGinty, B. Smyth, P. Nixon, Cats: a synchronous approach to collaborative group recommendation., in: Proceedings of the Conference on FLAIRS, vol. 2006, 2006, pp. 86–91.
[32] A. Mnih, R. Salakhutdinov, Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems, 2007, pp. 1257–1264.
[33] Netflix, Netflix dataset. (http://www.netflixprize.com), 2009, (accessed: 18.09.2015).
[34] J. Oh, S. Kim, J. Kim, H. Yu, When to recommend: A new issue on tv show recommendation, Inf. Sci. 280 (2014) 261–274.
[35] F. Ortega, J. Bobadilla, A. Hernando, A. Gutiérrez, Incorporating group recommendations to recommender systems: Alternatives and performance, Inf. Process. Manag. 49 (4) (2013) 895–901.
[36] M.J. Pazzani, D. Billsus, Content-based recommendation systems, in: The Adaptive Web, Springer, 2007, pp. 325–341.
[37] D.M. Pennock, E. Horvitz, S. Lawrence, C.L. Giles, Collaborative filtering by personality diagnosis: a hybrid memory-and model-based approach, in: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2000, pp. 473–480.
[38] C. Porcel, A. Tejeda-Lorente, M. Martínez, E. Herrera-Viedma, A hybrid recommender system for the selective dissemination of research resources in a technology transfer office, Inf. Sci. 184 (1) (2012) 1–19.
[39] F. Ricci, Travel recommender systems, IEEE Intell. Syst. 17 (6) (2002) 55–57.
[40] J.B. Schafer, J. Konstan, J. Riedl, Recommender systems in e-commerce, in: Proceedings of the 1st ACM Conference on Electronic Commerce, ACM, 1999, pp. 158–166.
[41] J. Serrano-Guerrero, E. Herrera-Viedma, J.A. Olivas, A. Cerezo, F.P. Romero, A Google wave-based fuzzy recommender system to disseminate information in university digital libraries 2.0, Inf. Sci. 181 (9) (2011) 1503–1516.
[42] J. Serrano-Guerrero, F.P. Romero, J.A. Olivas, Hiperion: A fuzzy approach for recommending educational activities based on the acquisition of competences, Inf. Sci. 248 (2013) 114–129.
[43] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Adv. Artif. Intell. 2009 (2009) 4.
[44] B. Suryavanshi, N. Shiri, S. Mudur, A fuzzy hybrid collaborative filtering technique for web personalization, in: Proceedings of 3rd International Workshop on Intelligent Techniques for Web Personalization (ITWP 2005), 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), 2005, pp. 1–8.
[45] Á. Tejeda-Lorente, C. Porcel, E. Peis, R. Sanz, E. Herrera-Viedma, A quality based recommender system to disseminate information in a university digital library, Inf. Sci. 261 (2014) 52–69.
[46] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, Z. Chen, Scalable collaborative filtering using cluster-based smoothing, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2005, pp. 114–121.
[47] K. Yoshii, M. Goto, K. Komatani, T. Ogata, H.G. Okuno, An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model, IEEE Trans. Audio Speech Lang. Process. 16 (2) (2008) 435–447.
[48] Z. Yu, X. Zhou, Y. Hao, J. Gu, Tv program recommendation for multiple viewers based on user profile merging, User Model. User-Adapt. Interact. 16 (1) (2006) 63–82.
[49] A. Zapata, V.H. Menéndez, M.E. Prieto, C. Romero, Evaluation and selection of group recommendation strategies for collaborative searching of learning objects, Int. J. Human-Comput. Stud. 76 (2015) 22–39.