Information Sciences 185 (2012) 1–17

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Collaborative filtering based on significances

Jesús Bobadilla *, Antonio Hernando, Fernando Ortega, Abraham Gutiérrez

FilmAffinity.com Research Team
Universidad Politecnica de Madrid, Crta. De Valencia, Km. 7, 28031 Madrid, Spain

ARTICLE INFO

Article history:
Received 2 September 2010
Received in revised form 27 May 2011
Accepted 8 September 2011
Available online 19 September 2011

Keywords:
Recommender systems
Collaborative filtering
Significances
Items

ABSTRACT

It seems reasonable to think that there may be some items and some users in a recommender system that could be highly significant in making recommendations. For instance, the recent and much-advertised Apple product may be regarded as more significant compared with an outdated MP3 device (which is still on sale). In this paper, we introduce a new method to improve the information used in collaborative filtering processes by weighting the ratings of the items according to their importance. We provide here a formalisation of the collaborative filtering process based on the concept of significance. In this way, the *k*-neighbours are calculated taking into account the ratings of the items, the significance of the items and the significance of each user for making recommendations to other users. This formalisation includes extensions of the concepts related to similarity measures and prediction/recommendation quality measures. We will show also the results obtained from a set of experiments using Movielens and Netflix. The results confirm the advantage of introducing the concept of significance in general recommender systems and especially in recommender systems in which it is easy to determine the relative importance of the items: for example, most widely sold products in e-commerce, most widely commented news items in web-news, most widely watched programs on TV, and the latest sports champions.

1. Introduction

Traditional Recommender Systems (RS) use a Collaborative Filtering (CF) process based on considering all items and all users to have the same importance. For these traditional RS systems, there is neither a difference between films nor a difference between users.

This assumption may be controversial because a high-budget film premiere may in many ways be considered more significant compared with a little-known older film. In the same way, the latest product of Apple (which has been strongly advertised) may be regarded as more significant compared with an equivalent cheap product made by an unknown manufacturer. Although there may be users who consider that an MP3 or a little-known film (which has been scarcely rated) may be just as important as Apple's latest products or a film premiere, most users of an RS consider that a recommendation error will be more significant when connected to a more important item. In this way, users mistrust much more those RSs that make mistakes on recommendations of significant items than those that make mistakes on recommendations of non-significant items.

The hypothesis of our paper is based on the assumption that neither all of the items have the same importance nor all of the users have the same importance when making recommendations to other users. To improve recommendations according

* Corresponding author. Tel.: +34 3365133; fax: +34 913367522.
E-mail addresses: jesus.bobadilla@upm.es (J. Bobadilla), ahernando@eui.upm.es (A. Hernando), fortegarequena@gmail.com (F. Ortega), abraham@eui.upm.es (A. Gutiérrez).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.09.014

to a significance measure, we provide in this paper similarity measures and quality measures that take into account the significance of the users and items.

The question now is how to measure the significance of items and users. For some RSs, this measure may be easily formulated. In Geographic RSs, the significance of items (such as restaurants or shops) can be established taking into account how far these items are from the user. However, this type of rule cannot be applied to every type of RS. In this paper, we propose a general method for calculating the significance of items and users independently of the RS.

As far as the significance of items is concerned, we have considered here different parameters to calculate the significance. Regarding the CF processes, the significance of items involves the following steps:

1. We must define a significance measure of items that can be calculated considering only the ratings that each user has made on each item.
2. We must define new similarity measures based on significance measures previously defined, to improve the recommendations made over the most significant items.
3. We must define new quality measures that take into account the significance of the recommended items, to study the errors of the recommendations made on these items. As we have previously stated, an error in a very significant item is much more serious than in a less-significant item.

As far as the significance of users is concerned, we must state that this significance is dealt with differently. Unlike the significance of items, we should here distinguish between two types of significance for users on behalf of their role in a CF based on users: the significance associated with users to whom recommendations are made and the significance associated with users who are used to make recommendations to other users. Defining a significance measure for users is based on the idea that there may be users in the RS who are much more important compared with others for making recommendations. It seems reasonable to think that those users who have rated thousands of films are much more significant in their recommendations compared with a user who has rated, say, only 25 films (the latter one seems not to have seen enough films to make trusty recommendations). In the same way, users who have rated in a balanced way are much more significant in their recommendations compared with a user who has rated thousands of items with the maximum possible value (this user seems to like every film, and consequently, we should mistrust his recommendations). We will define this measure by taking into account only the ratings made by users.

In Section 2, we will review some related work and overview concepts related to RS. In Section 3, we will describe our approach based on the concept of significance in an RS. In Section 4, we will show the experiments performed, which confirm the advantages of our approach. Finally, in Section 5, we will discuss our conclusions.

2. Preliminaries and related work

Recommender Systems are systems that are able to provide personalised recommendations through different types of algorithms (usually termed "filtering"). We will consider the following types of algorithms:

1. Content-based filtering [2]: Recommendations to users are made taking into account the information related to the items (trips, films, books, etc.) that other users have rated previously. In this way, a recommender system for books will strongly recommend a well-considered historical novel to a user who has bought any other historical novel.
2. Demographic filtering [30]: Recommendations to users are made taking into account the positive ratings made by users sharing stats such as geographical location, age, qualifications, etc.
3. Collaborative filtering (CF) [1,3,8,26,22,32,39]: Recommendations to users are made taking into account how other users have rated items (which are stored in databases). To make recommendations to a user, the RS tries first to find users with common tastes (users who have rated items in a way similar to the item being considered; such a user will be named a neighbour of the item under consideration). Once the neighbours are found, the RS recommends those items that have been highly rated by the user's neighbours.
4. Hybrid filtering methods [15,5,17]: Recommendations are made through techniques based on both demographic filtering and CF. Because these hybrid methods use different sources of knowledge, they can tackle situations in which there is a small database of users' ratings ("cold-start" problem [27,31,38]).

Among those filtering methods, CF is the most used because it provides the best results. To make good recommendations, CF needs a very extensive database containing exclusively the ratings made by users over the items. This extensive database usually contains thousands of users (or even hundreds of thousands of users), thousands of items and hundreds of thousands of ratings (or even millions of ratings). Moreover, because each user normally rates very few items, there is a high degree of sparsity [7,35].

CF needs to choose a similarity function [11,20,21,12,37] that measures how similar two users are. Some of the most well-known similarity measures are: Pearson correlation, cosine, constrained Pearson's correlation, Spearman rank correlation, mean squared differences and Jaccard. Recently, other similarity measures have been proposed [8,9], leading to better empirical results. Once the similarity measure is chosen, we can make recommendations to any user, A, through the following steps:

1. Obtain a set of K users similar (through the chosen similarity measure) to the user A. These K users are termed neighbours of user A.
2. Predict the ratings that user A would make over items that user A has not rated yet. These predictions are made through the K neighbours calculated in the previous step. The methods to calculate these predictions are usually termed "aggregation approaches"; the best-known of these: the average, the weighted sum and the adjusted weighted aggregation (deviation-from-mean).
3. Recommend to user A those N items that have been predicted to be rated by user A with the highest values.

To compare several RSs, different quality measures have been proposed:

- Prediction quality measures:
  - Accuracy. This measure informs us of how far the predictions are from the real ratings made over the items. The most frequently used accuracy measures [16,18,23,34] are: MAE (Mean Absolute Error), MSE (Mean Squared Error), and RMSE (Root Mean Squared Error).
  - Coverage. This measure informs us of how many predictions can be made. We cannot predict with which value the user would rate an item in the case that none of its K neighbours has rated the item. Indeed, this measure is the ratio of the total number of predictions that can be calculated to the number of items that have not yet been rated.
  - Percentage of perfect predictions, percentage of bad predictions, etc.
- Recommendation quality measures:
  - Precision. This measure informs us about the ratio of the number of relevant recommendations made to the total number of recommendations made (N).
  - Recall. This measure informs us about the ratio of the number of relevant recommendations made to the total number of relevant recommendations that can indeed be made.
  - Fall-out, specificity, novelty, trust, etc. [25,41].

Researchers have proposed different methods for tackling some difficulties related to how RSs perform evaluation. In [22], some key factors are considered for evaluating the CF in RS: the user tasks; the type of analysis and datasets that are used; the ways in which prediction quality is measured; and the user-based evaluation of the system as a whole. In [23], a method has been recently proposed that makes a distinction between interactive and non-interactive subsystems. There are other lines of research that review the most commonly accepted metrics, aggregation approaches and evaluation measures. In [19], researchers identify different features in RS that are not related to the evaluation. In [10], there is an evaluation of the accuracy of different methods studied over a set of representative problems. Both [11,38] review methods related to CF.

Different papers have proposed a general framework for CF. In [20], the following issues are studied: similarity weighting, significance weighting, variance weighting, selecting neighbours and rating normalisation. In [23], a framework based on two different subsystems is described: a subsystem for guiding users and a subsystem for providing interesting items. The framework proposed recently in [29] uses different levels of abstraction to get a much more flexible recommender system.

Due to the fact that these RSs make no consideration of the significance of the difference between users, items and ratings can be clearly seen in the way that the MAE is measured (Eq. (1), below). As may be seen in Eq. (1), there is no distinction between users, items and ratings:

$$\frac{1}{\#\text{users} \times \#\text{items}} \sum_{\text{users}} \sum_{\text{items}} \text{error}_{\text{user, item}} | \exists \text{error}_{\text{user, item}} \iff \exists \text{vote}_{\text{user, item}} \wedge \exists \text{prediction}_{\text{user, item}} \tag{1}$$

In general, the methods proposed assume implicitly that there is no difference between the following elements in a RS:

1. There is no difference between users in the RS. However, some research has shown the convenience of distinguishing between users. In the case of e-learning RS [6,13,24], different types of users should be considered, because teachers and advanced students usually provide better recommendations than novice students. A Trust-aware RS (TARS) is an RS that measures the trust of a user through its competence and behaviour during a period of time [41]. In this sense, [36] provides a model for obtaining trust measures through similarity functions. Fuzzy relations provide an outstanding tool for modelling trust networks. In [33], a description is given of two fuzzy logic applications (fuzzy inference system and fuzzy MCDM), which manage the quality and the veracity of peers' contributions. The research described in [40] uses a trust model whose trust scores are pairs (trust, distrust) obtained from a bilattice.
2. There is no difference between items in the RS. Errors in recommendations are dealt with in the same way regardless of the importance of the item considered. Nevertheless, there are some specific research lines proposing the use of a timestamp in which each rating is made with the aim of improving the accuracy of the system through time-weight techniques [14,28]. In this way, those ratings within a shorter time interval generate greater significances in their associated items. The method proposed here seems to be closer to the one presented in [27]. In the latter paper, a description is given of an optimisation algorithm to compute automatically the weights of different items through the ratings

made by a set of training users. However, our approach is altogether different, because [27] makes use of a clustered distribution for users in the space of items. The method presented in [4] also differs from ours, inasmuch as it is based on selecting the highest-weighted items so as to use them to make predictions.

3. There is no difference between the ratings a user makes. The present RS only normalises the ratings of each user by considering: (a) standardisation processes such as z-scores (or normal scores) [42]; (b) baseline techniques because some users are often rated more benevolently than others; and (c) the arithmetic average, as it is used in some similarity measures (such as the Pearson Correlation).

3. Formalism of the new approach

3.1. Obtaining the significances

In the next tables, we show the parameters (Table 1), measures (Table 2) and sets (Table 3) used in this section.

We will consider a recommender system with l users and m items. Users rate items with a discrete range of possible values $\{min, min+1, \ldots, max-1, max\}$. We will use the symbol $\bullet$ to state that a user has not rated an item yet. We define the following sets:

$$U = \{u \in NaturalNumbers | 1 \leqslant u \leqslant l\}, \text{set of users} \tag{2}$$

$$I = \{i \in NaturalNumbers | 1 \leqslant i \leqslant m\}, \text{set of items} \tag{3}$$

$$W = \{w \in NaturalNumbers | \min \leqslant w \leqslant \max\} \cup \{\bullet\}, \text{ set of possible votes} \tag{4}$$

$$R_u = \{(i, v) | i \in I, v \in W\}, \text{ ratings of user } u \tag{5}$$

$$\text{We use } r_{u,i} \text{ to represent the rating made by the user } u \text{ over the item } i. \tag{6}$$

We define the cardinality of a set C as the number of its elements.

$$\#C = \#\{x \in C | x \neq \bullet\} \tag{7}$$

$$\#R_u = \#\{i \in I | r_{u,i} \neq \bullet\} \tag{8}$$

Table 1
Parameters.

Name	Parameters descriptions
l	# Users
m	# Items
min	# Min rating value
max	# Max rating value
k	# Neighbourhoods
N	# Recommendations
θ	Recommendation threshold
z	# Similar items

Table 2
Measures.

Name	Measures descriptions	Usage
$r_{u,i}$	Rating of the user on the item	Ratings and significances
$s_{u,i}$	Significance of the item to the user	
s_i	Significance of the item	
s_u	Significance of the user to recommend other users	
$p_{u,i}$	Prediction to the user on the item	
$sim(i_x,i_y)$	Item to item	Similarity measures
$sim^*_s(u_x,u_y)$	User to user	
mae_u	Mean Absolute Error of the user u	Prediction quality
mae	Mean Absolute Error of the RS	
c_u	Coverage$_s$ on the user	
c	Coverage$_s$ of the RS	
h_u	Recommendation precision on the user	Recommendation quality
h	Recommendation precision of the RS	
x_u	Recommendation recall on the user	
x	Recommendation recall of the RS	

Table 3
Sets.

Name	Sets descriptions	Parameters
U	Users	*l*
I	Items	*m*
W	Rating values	*min, max*
A_i	Users who have voted for the item	*Item*
R_u	User ratings	*User*
V	Relevant values of the set W	
V^c	Not relevant values of the set W	
D_u	Relevant items for the user	*User*
E_u	Not relevant items for the user	*User*
S_i^z	z-neighbours similar to the item	*Item*, *z*
F_u	Items $\in S_i^z$ rated for the user and having significances	*User*
$B_{x,y}$	Items rated simultaneously by users x and y	*User1, user2*
$B_{x,y}^*$	Items which simultaneously have significances on users x and y	*User1, user2*
K_u	Neighbourhoods of the user	*User, k*
P_u	Predictions to the user	*User, k*
$G_{u,i}$	User's neighbourhoods which have rated the item	*User, item*
X_u	Top recommended items to the user	*User, k,* θ
Z_u	Top N recommended items to the user	*User*, *k*, *N*, θ
H_u	Set of items not rated by user u which significance exists	
L_u	Items that the user has not voted for and on which predictions exist	*User*
M_u	Items that the user has not voted for	*User*
Y_u	Set of recommended relevant items (true-positives)	*User*
N_u	Set of recommended not relevant items (false-positives)	*User*
O_u	Set of not recommended relevant items (false-negatives)	*User*

3.1.1. Significance of an item i (s_i)

$$\text{Let } A_i = \{u \in U | r_{u,i} \neq \bullet\} \text{ be the set of users who have rated the item } i. \tag{9}$$

We will define the significance of item i according to the following expression:

$$s_i = \left(\frac{1}{\#A_i}\sum_{u\in A_i} r_{u,i}\right)\left(\frac{\#A_i}{\#U}\right) = \frac{\sum_{u\in A_i} r_{u,i}}{\#U}, \quad r_{u,i} \in [0,1], \quad s_i \in [0,1] \tag{10}$$

The first term of the expression calculates the significance, taking into account all of the ratings made over the item i. The second term of the expression calculates the significance taking into account the ratio of the number of ratings over the item i to the total number of users. Finally, the significance of an item can be expressed as the division of the sum of the ratings made over the item i into the number of users in the RS.

3.1.2. Significance of a user (s_u) to recommend to other users

$$\text{Let } V = \{i, \ldots, \max\} \text{ be the subset of } W \text{ with elements that are regarded as relevant ratings.} \tag{11}$$

$$\begin{aligned}\text{Let } V^c &= W - V - \{\bullet\} \\ &= \{\min, \ldots, i-1\} \text{ be the subset of } W \text{ with elements that are regarded as non-relevant ratings.}\end{aligned} \tag{12}$$

In the case that $W = \{1,2,3,4,5,\bullet\}$, we may consider that $V = \{4,5\}$ is the set of relevant ratings and $V^c = \{1,2,3\}$ is the set of irrelevant ratings.

$$\text{Let } D_u = \{i \in I | r_{u,i} \in V\} \text{ be the set of items that user } u \text{ has rated with a relevant value.} \tag{13}$$

$$\text{Let } E_u = \{i \in I | r_{u,i} \in V^c\} \text{ be the set of items that user } u \text{ has rated with a non-relevant value.} \tag{14}$$

To weight the importance of a rating, we use the following factors:

- $f_1 = 1 - \frac{\#D_u}{\#D_u + \#E_u}$ .According to the factor f_1, the lower the number of relevant ratings made by user u, the higher the significance of this user's u relevant ratings.
- $f_2 = \frac{\#D_u + \#E_u}{\#I}$.

According to this factor, the higher the number of ratings made by user u (as may be seen, $\#D_u$+$\#E_u$ is the precise number of ratings made by user u), the higher the significance of user u to make recommendations.

Finally, we define the significance of a user to recommend to other users as:

$$s_u = \left(1 - \frac{\#D_u}{\#D_u + \#E_u}\right)\left(\frac{\#D_u + \#E_u}{\#I}\right), \quad s_u \in [0,1]. \tag{15}$$

Eq. (15) provides a measure of the significance of a user u for recommending to others. This measure is calculated on behalf of the following:

- The ratio of the relevant ratings made by the user u to the number of ratings made by u.
- The ratio of the number of ratings made by u to the number of possible ratings that a user may make (that is to say, the number of items).

3.1.3. Significance of item i for user u ($s_{u,i}$)

Next, we define the significance of an item i for a user u by means of three possible cases. In the first case, we will combine the value of the rating that user u has made over item i with the significance of user u and the significance of item i. We will consider a second case if user u has not rated item i yet. In this second case, we will try to predict the rating user u would make over item i by analysing the ratings user u has made over those items similar to i. However, we were not able to calculate this prediction because there are no items similar to i that have been rated by user u. In this third case, we consider that no significance is assigned to item i for user u. Next, we describe this formally:

Case (1)

$$s_{u,i} = r_{u,i} s_u s_i \iff r_{u,i} \neq \bullet \wedge s_u \neq \bullet \wedge s_i \neq \bullet, \quad r_{u,i} \in [0,1], \quad s_{u,i} \in [0,1]. \tag{16}$$

case (2) Let S_i^z be the set of the z items most similar to i. We know that this set fulfils the following:

$$\#S_i^z = z \quad \text{and} \quad \forall j \in S_i^z \quad \forall t \notin S_i^z sim(j,i) \geqslant sim(t,i), \tag{17}$$

where sim (a,b) indicates the similarity between the items a and b. Let F_u be the subset of items of S_i^z that have been rated by user u.

$$\text{Let } F_u = \{j \in S_i^z \wedge r_{u,j} \neq \bullet\}; \tag{18}$$

$$s_{u,i} = s_u s_i \frac{1}{\mu} \sum_{j \in F_u} s_j r_{u,j} sim(j,i) \iff \tag{19}$$

$$r_{u,i} = \bullet \wedge s_u \neq \bullet \wedge s_i \neq \bullet \wedge F_u \neq \emptyset, \quad r_{u,j} \in [0,1], \quad s_{u,i} \in [0,1],$$

$$\text{where}: \mu = \sum_{j \in F_u} sim(j,i). \tag{20}$$

case (3)

$$s_{u,i} = \bullet \iff r_{u,i} = \bullet \wedge F_u = \emptyset \tag{21}$$

Fig. 1 shows the significance of measures s_i, s_u and $s_{u,i}$ obtained for the typical RS database: Movielens.

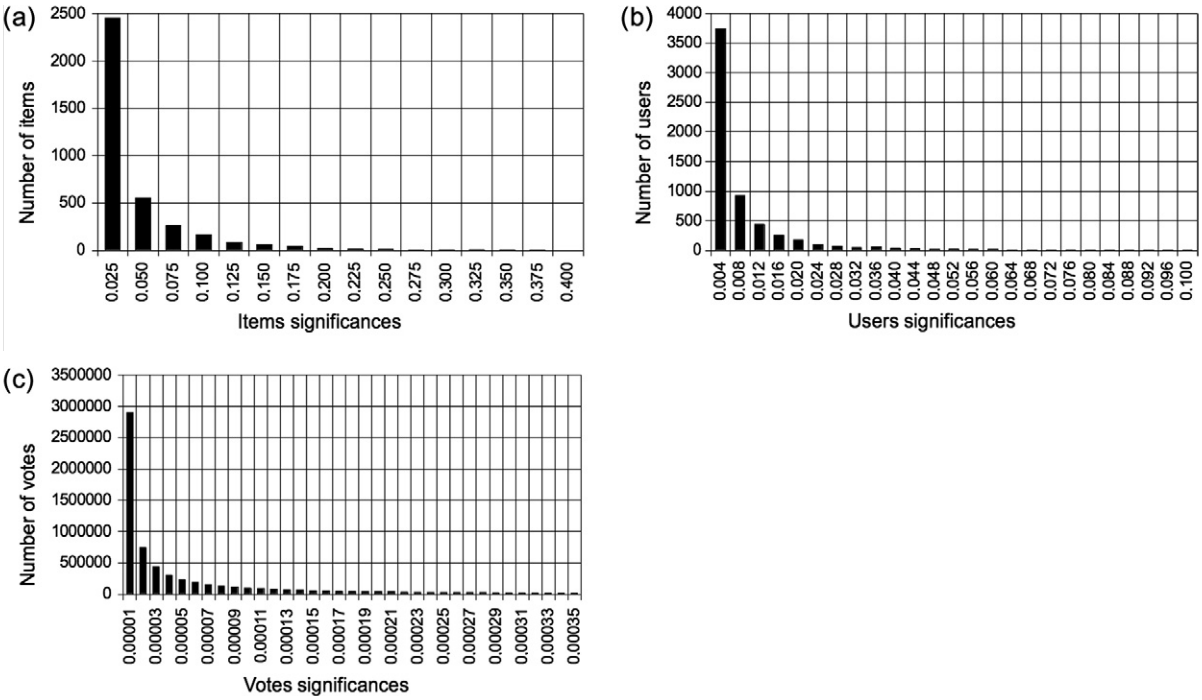

Fig. 1. s_i, s_u and $s_{u,i}$ typical measures (obtained for the Movielens database).

3.1.4. Running example

Table 4 shows a micro-running example that helps the reader to understand the equations formulated in this paper.

Each cell in Table 4 (except for those corresponding to the last row and the last column) provides the value of the ratings made by each user. The values on the left side of each "/" symbol inform us about the rating made by the user. The values on the right side show the value of the ratings normalised in the interval [0, 1]. The last row shows the significance of the items, and the last column shows the significance of the users.

As can be observed from Table 4, user u_5 makes only relevant ratings. Consequently, his significance is 0. Situations involving users such as user u_5 can be found in RS with the cold-start problem and with ill-intentioned users.

$$U = \{u \in NaturalNumbers | 1 \leqslant u \leqslant 5\}, \quad I = \{i \in NaturalNumbers | 1 \leqslant i \leqslant 10\}$$
$$W = \{w \in NaturalNumbers | 1 \leqslant w \leqslant 5\} \cup \{\bullet\}, \quad V = \{4, 5\}$$

To obtain the significances of the ratings ($s_{u,i}$), we need to determine the k-neighbours of each item. With the purpose of simplifying the calculations, we will use in this example a very simple similarity measure:

$$\text{Let } J_{x,y} = \{u \in U | r_{u,x} \neq \bullet \wedge r_{u,y} \neq \bullet\}, sim(x,y) = 1 - \frac{1}{\#J_{x,y}} \sum_{u \in J_{x,y}} |r_{u,x} - r_{u,y}|.$$

Table 5 shows the similarity values between items that have been obtained; 1.0 indicates the maximum similarity.

Based on the values in Table 5, we obtain the k-neighbours of each item, as shown in Table 6.

Table 4
Ratings (absolute rating/normaliszed rating). Greay areas: more significant users and items.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	s_u
u_1	•	•	5/1.0	•	•	5/1.0	•	•	•	3/0.6	0.10
u_2	2/0.4	5/1.0	•	•	•	5/1.0	2/0.4	•	3/0.6	•	0.3
u_3	•	•	4/0.8	•	•	3/0.6	•	•	•	4/0.8	0.10
u_4	•	1/0.2	3/0.6	2/0.4	5/1.0	•	•	•	•	4/0.8	0.3
u_5	4/0.8	5/1.0	•	4/0.8	5/1.0	4/0.8	4/0.8	5/1.0	•	5/1.0	0
s_i	0.24	0.44	0.48	0.24	0.4	0.68	0.24	0.2	0.12	0.64	

Table 5
Similarity between items. The symbol "—"informs us of about the impossibility to of measuringe the similarity between two items; it is impossible to measure the similarity between two items when since there are no users who have simultaneously rated both items.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
i_2	0.6								
i_3	–	0.6							
i_4	1.0	0.8	0.8						
i_5	0.8	0.6	0.6	0.6					
i_6	0.7	0.9	0.9	1.0	0.8				
i_7	1.0	0.6	–	1.0	0.8	0.7			
i_8	0.8	1.0	–	0.8	1.0	0.8	0.8		
i_9	0.8	0.6	–	–	–	0.6	0.8	–	
i_{10}	0.8	0.7	0.8	0.7	0.9	0.73	0.8	1.0	–

Table 6
The 3-neighbours of each item.

i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
{i4, i7, i5}	{i8, i6, i4}	{i6, i4, i10}	{i1, i6, i7}	{i8, i10, i1}	{i4, i2, i3}	{i1, i4, i5}	{i2, i5, i10}	{i1, i7, i2}	{i8, i5, i1}

Table 7
Significances su,i. GrayGrey colorcolour: processed using item neighborsneighbours.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	s_u
u_1	•	0.030	0.048	0.016	0.015	0.068	•	0.007	•	0.038	0.10
u_2	0.029	0.132	0.097	0.054	0.011	0.204	0.028	0.026	0.021	0.018	0.3
u_3	•	0.018	0.038	0.010	0.020	0.041	•	0.010	•	0.051	0.10
u_4	0.053	0.026	0.086	0.028	0.12	0.056	0.053	0.040	0.003	0.153	0.3
u_5	0	0	0	0	0	0	0	0	0	0	0
s_i	0.24	0.44	0.48	0.24	0.4	0.68	0.24	0.2	0.12	0.64	

Table 7 shows the significance values of each rating. The numerical values with a white background are obtained by applying Eq. (16); the values shown in grey are obtained by applying the equations from (17)–(20); if the significance measure cannot be calculated (see Eq. (21)), then we use the symbol. In the example, the value of $s_{u4,i7}$ has been calculated by means of the following expression:

$$s_{u4,i7} = 0.3 \times 0.24 \frac{1}{1+0.8}(0.24 \times 0.4 \times 1 + 0.4 \times 1 \times 0.8).$$

3.2. Definition of the significance metrics (s-measures)

Once we have defined different types of significance measures, we can then define similarity measures between users, sim_s, which take into account the significance measures previously defined.

The traditional similarity metrics between users x and y are defined over the items that both users have rated:

$$B_{x,y} = \{i \in I | r_{x,i} \neq \bullet \wedge r_{y,i} \neq \bullet\}. \tag{22}$$

Our proposed s-similarity measures between users x and y are defined over the items to which a significance measure can be defined for users x and y (see Section 3.1.3):

$$B^*_{x,y} = \{i \in I | s_{x,i} \neq \bullet \wedge s_{y,i} \neq \bullet\}. \tag{23}$$

Please note that $B_{x,y} \subseteq B^*_{x,y}$. (24)

In the example, the Pearson$_s$, cosine$_s$ and MSD$_s$ s-measures are respectively defined through the Pearson, cosine and MSD, as follows:

Pearson correlation:

$$Pearson(x,y) = \frac{\sum_{i \in B_{x,y}} (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in B_{x,y}} (r_{x,i} - \bar{r}_x)^2 \sum_{i \in B_{x,y}} (r_{y,i} - \bar{r}_y)^2}}, \tag{25}$$

$$\text{where} : C = \{j \in I | r_{z,j} \neq \bullet\}, \quad \bar{r}_z = \frac{1}{\#C} \sum_{i \in C} r_{z,i}. \tag{26}$$

$$Pearson_s(x,y) = \frac{\sum_{i \in B^*_{x,y}} (s_{x,i} - \bar{s}_x)(s_{y,i} - \bar{s}_y)}{\sqrt{\sum_{i \in B^*_{x,y}} (s_{x,i} - \bar{s}_x)^2 \sum_{i \in B^*_{x,y}} (s_{y,i} - \bar{s}_y)^2}}, \tag{27}$$

$$\text{where} : C^* = \{j \in I | s_{z,j} \neq \bullet\}, \quad \bar{s}_z = \frac{1}{\#C^*} \sum_{i \in C^*} s_{z,i}. \tag{28}$$

Cosine:

$$Cos(x,y) = \frac{\sum_{i \in B_{x,y}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in B_{x,y}} r^2_{x,i}} \sqrt{\sum_{i \in B_{x,y}} r^2_{y,i}}}; \tag{29}$$

$$Cos_s(x,y) = \frac{\sum_{i \in B^*_{x,y}} s_{x,i} s_{y,i}}{\sqrt{\sum_{i \in B^*_{x,y}} s^2_{x,i}} \sqrt{\sum_{i \in B^*_{x,y}} s^2_{y,i}}}. \tag{30}$$

Mean Squared Differences:

$$MSD(x,y) = 1 - \frac{1}{\#B_{x,y}} \sum_{i \in B_{x,y}} (r_{x,i} - r_{y,i})^2; \tag{31}$$

$$MSD_s(x,y) = 1 - \frac{1}{\#B^*_{x,y}} \sum_{i \in B^*_{x,y}} (s_{x,i} - s_{y,i})^2. \tag{32}$$

3.3. Prediction and recommendation processes

To generate the top N recommendations for a given user u, we will use the selected s-similarity measure. First, we will obtain the k-neighbours for this user through the s-similarity measure; next, we will perform the prediction and recommendation process used in the traditional CF. Now, we present the mathematical formalisation of both processes.

Let K_u be the set of k neighbours of user u. Assuming there exist at least k neighbours, the following statements must hold (as may be seen, we use the s-similarity measures):

$$K_u \subset U \wedge \#K_u = k \wedge u \notin K_u; \tag{33}$$

$$\forall x \in K_u, \quad \forall y \in (U - K_u), \ sim_s(u,x) \geqslant sim_s(u,y). \tag{34}$$

The prediction $p_{u,i}$ of the rating that a user u would make over item i is calculated through the ratings made by the k-neighbours.

Once the set of k-neighbours of a user u, K_u, has been calculated, the prediction $p_{u,i}$ is calculated by means of an aggregation function. The most common aggregation functions are: the average (36), the weighted sum (37) and the adjusted weighted aggregation (deviation-from-mean) (38).

Previously, we will first define the set $G_{u,i}$ as the set of k-neighbours of the user u who have rated item i:

$$G_{u,i} = \{n \in K_u | \exists r_{n,i} \neq \bullet\}. \tag{35}$$

Next, we will define the following common aggregation functions:

$$p_{u,i}^{av} = \frac{1}{\#G_{u,i}} \sum_{n \in G_{u,i}} r_{n,i} \Longleftrightarrow G_{u,i} \neq \emptyset; \tag{36}$$

$$p_{u,i}^{ws} = \mu_{u,i} \sum_{n \in G_{u,i}} sim_s(u,n) r_{n,i} \Longleftrightarrow G_{u,i} \neq \emptyset; \tag{37}$$

$$p_{u,i}^{DFM} = \bar{r}_u + \mu_{u,i} \sum_{n \in G_{u,i}} sim_s(u,n)(r_{n,i} - \bar{r}_n) \Longleftrightarrow G_{u,i} \neq \emptyset, \tag{38}$$

where μ serves as a normalising factor, usually calculated as:

$$\mu_{u,i} = 1 \Bigg/ \sum_{n \in G_{u,i}} sim_s(u,n) \Longleftrightarrow G_{u,i} \neq \emptyset. \tag{39}$$

If item i has not been rated yet by any k-neighbours, the prediction $p_{u,i}$ cannot be calculated:

$$p_{u,i} = \bullet \Longleftrightarrow G_{u,i} = \emptyset. \tag{40}$$

To calculate the best N recommendations, we define first the set X_u as the set of items not rated by user u and over which we can predict the rating this user u would make.

$$X_u \subset I \wedge \forall i \in X_u, r_{u,i} = \bullet, p_{u,i} \neq \bullet, \tag{41}$$

Having defined X_u, we define the set of the N best recommendations, Z_u, as the set of N items in X_u that are predicted to be rated with the highest value. Assuming that, for these items, there are at least N recommendations:

$$Z_u \subseteq X_u, \quad \#Z_u = N, \quad \forall x \in Z_u, \quad \forall y \in X_u : p_{u,x} \geqslant p_{u,y}. \tag{42}$$

Running example: We can calculate the similarity between users through the significances obtained in Table 4. We will use the following s-similarity measure:

Using Eq. (23): $B^*_{x,y} = \{i \in I | s_{x,i} \neq \bullet \wedge s_{y,i} \neq \bullet\}$,

$$sim^*_s(u_x, u_y) = 1 - \frac{1}{\#B^*_{x,y}} \sum_{i \in B^*_{x,y}} |s_{u_x,i} - s_{u_y,i}|.$$

Table 8 shows the similarity measures between two users in terms of the significance measures described in Tables 7 and 9 shows the 2-neighbourhood of each user with the similarities presented in Table 8.

Because there is only one item (item 6) that has been rated by user u_1 and user u_2, the similarity between both users is maximal (i.e., 1.0) when it is calculated through traditional measures (in which the significance measure is not taken into account). This similarity measure seems to be unreasonable because it is calculated through only one common item.

Table 8
Similarities between users.

	u_1	u_2	u_3	u_4
u_2	0.947			
u_3	0.990	0.937		
u_4	0.954	0.938	0.954	
u_5	0.968	0.938	0.973	0.938

Table 9
A user's 2-neighbours, processed using significant items.

u_1	u_2	u_3	u_4	u_5
{u3, u5}	{u1, u4}	{u1, u5}	{u1, u3}	{u3, u1}

Table 10
Predictions.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	4.0	5.0	4.0	4.0	5.0	3.5	4.0	5.0	•	4.5
u_2	•	1.0	4.0	2.0	5.0	5.0	•	•	•	3.5
u_3	4.0	5.0	5.0	4.0	5.0	4.5	4.0	5.0	•	4.0
u_4	•	•	4.5	•	•	4.0	•	•	•	3.5
u_5	•	•	3.5	•	•	4.0	•	•	•	3.5

However, the similarity between these users using our *s*-similarity metric (which is related to significance) takes more items into consideration and therefore is different from 1 (0.965).

Now we can calculate the predictions $p_{u,i}$ using Eqs. (35) and (36) over the data in Tables 4 and 9. Table 10 shows the values of these predictions.

3.4. Prediction quality measures with significance weighting

3.4.1. MAE with significance weighting (MAE_s)

In this section, we will define a measure MAE_s for considering how efficient the RS is while making predictions (taking into account the significance of items).

To calculate the MAE_s, we will weight the prediction errors according to the significance of the items. In this way, we can obtain a low MAE if the errors in our predictions over the significant items are low.

Let H be the following set of pairs (user, item):

$$H = \{< u, i > | r_{u,i} \neq \bullet \wedge p_{u,i} \neq \bullet \wedge s_i \neq \bullet \wedge s_u \neq \bullet\}. \quad (43)$$

We define MAE_s as:

$$MAE_s = \frac{1}{\sum_{<u,i>\in H} s_u s_i} \sum_{u\in U} \sum_{i\in I} |r_{u,i} - p_{u,i}| s_u s_i| < u, i > \in H. \quad (44)$$

3.4.2. $Coverage_s$

In traditional RS, the coverage of an RS is defined as the percentage of predictions that can be calculated (a prediction can be made on an item if this item has been rated by at least a *k*-neighbour).

We will redefine the coverage so that coverage takes into account the significance of items. Like MAE_s, we will define $coverage_s$ by weighting the significances of the items. In this way, the $coverage_s$ will be increased significantly if the RS can predict the ratings over the most significant items.

We will introduce the previous two sets of items needed to make a formal definition of $coverage_s$.

Let L_u be the set of items not rated by user u over which predictions can be calculated:

$$L_u = \{i \in I | r_{u,i} = \bullet \wedge G_{u,i} \neq \emptyset\}. \quad (45)$$

Let M_u be the set of items not rated by user u:

$$M_u = \{i \in I | r_{u,i} = \bullet\}. \quad (46)$$

The $coverage_s$ of user u is defined as follows:

$$c_u = 100 \times \frac{\sum_{i\in L_u} s_i}{\sum_{i\in M_u} s_i} \iff M_u \neq \emptyset, \quad c_u = \bullet \iff M_u = \emptyset. \quad (47)$$

Now, we will define the $coverage_s$ of the RS as follows:

$$\text{Let } L = \{(u,i) | u \in U, i \in I, r_{u,i} = \bullet, G_{u,i} \neq \emptyset\}; \quad (48)$$
$$\text{Let } M = \{(u,i) | u \in U, i \in I, r_{u,i} = \bullet\}. \quad (49)$$

The $coverage_s$ of the RStem is defined as follows:

$$c = 100 \times \frac{\sum_{i\in L} s_i}{\sum_{i\in M} s_i} \iff M \neq \emptyset, \quad c = \bullet \iff M = \emptyset. \quad (50)$$

3.4.3. Running example

Table 11 shows both the errors made in the predictions obtained in Table 10 and the values of MAE_s for each user and for the system. The values (u_x, i_y) in the table show the error in the prediction $(|r_{u,i} - p_{u,i}|)$ calculated through Eq. (44); The symbol '•' shows that either there are no ratings ($r_{u,i} = \bullet$) or there is no prediction ($p_{u,i} = \bullet$). The penultimate row shows

Table 11
Errors and MAEs obtained using significances.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	MAE_u
u_1	•	•	1.0	•	•	1.5	•	•	•	1.5	1.367
u_2	•	4.0	•	•	•	0.0	•	•	•	•	1.571
u_3	•	•	1.0	•	•	1.5	•	•	•	0.0	0.833
u_4	•	•	1.5	•	•	•	•	•	•	0.5	0.928
u_5	•	•	•	•	•	0.0	•	•	•	1.5	0.727
s_i	0.24	0.44	0.48	0.24	0.4	0.68	0.24	0.2	0.12	0.64	
										MAE	1.212

Table 12
The coverages of each user and of the system.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	c_u
u_1	•	•		•	•		•	•	•		93.6%
u_2			•	•	•			•		•	89.8%
u_3	•	•		•	•		•	•	•		93.6%
u_4	•					•	•	•	•		45.9%
u_5			•						•		80.0%
s_i	0.24	0.44	0.48	0.24	0.4	0.68	0.24	0.2	0.12	0.64	
										Coverage	74.3%

the significance measure of each item (s_i) calculated through Eqs. (9) and (10). The last column shows the measure MAE_s of each user. The MAE_s of the RS is shown in the last row.

Table 12 shows the information needed to obtain the $coverage_s$ of the system. An empty cell is used for describing that an item has been at least rated by a user; the symbol • is used for representing that a user has not rated an item; a grey cell is used for describing that an item has not been rated by a user even though its rating can be predicted (see Table 10).

By applying equations from (45)–(47), we obtain the $coverage_s$ of each user (they are calculated in the last column in Table 12). By applying equations from (48)–(50), we obtain the $coverage_s$ of the system (calculated in the last row in Table 12).

3.5. Recommendation quality measures with significance weighting ($precision_s$ & $recall_s$)

In this section, we will study quality measures related to the recommendations made by the RS, taking into account the significance measure. In Section 3.3, we define the set Z_u (see Eq. (42)) as the set of items that the RS recommends to the user u. In this section, we will introduce a constant θ, which is used to define those items that can be indeed recommendable for a given user.

When studying recommendation quality measures, the constraint $p_{u,i} \geqslant \theta$ (where θ is a threshold constant) is usually also considered so that an item i can be regarded as recommendable for a user u.

Next, we will define two recommendation quality measures related to significance: the significance precision measure, $precision_s$ (h_u), and the significance recall measure, $recall_s$ (x_u). Assuming that all users accept N test recommendations, we will define the following set of items:

Let Y_u be the set of items (true-positives), $i \in I$, such that RS derives that they are recommendable for user u ($i \in Z_u$) and that they are indeed items recommendable for user $u(r_{u,i} \geqslant \theta)$:

$$Y_u = \{i \in Z_u | r_{u,i} \geqslant \theta\}. \tag{51}$$

Let N_u be the set of items (false-positives) $i \in I$ such that RS derives that they are recommendable for user u ($i(\in Z_u$) and that they are indeed items non-recommendable for user $u(r_{u,i} < \theta)$:

$$N_u = \{i \in Z_u | r_{u,i} < \theta\}. \tag{52}$$

Let O_u be the set of (false-negatives) $i \in I$ such that RS derives that they are not recommendable for user u ($i \notin Z_u$) and that they are indeed items recommendable for user $u(r_{u,i} \geqslant \theta)$:

$$O_u = \left\{i \in Z_u^c | r_{u,i} \geqslant \theta\right\}. \tag{53}$$

The $precision_s$, either for a user, h_u (see Eq. (54)), or for the whole RS (see Eq. (55)), h, is defined as the ratio of the true-positive significances to the total recommended significances.

$$h_u = \frac{\sum_{i \in Y_u} s_{u,i}}{\sum_{i \in Y_u} s_{u,i} + \sum_{i \in N_u} s_{u,i}}; \tag{54}$$

$$h = \frac{1}{\#U} \sum_{u \in U} h_u. \tag{55}$$

Table 13
Contents of the figures; quality and similarity metric combinations tested in the experiments. COR: Pearson-correlation, sCOR: Pearson-correlations, COS: cosine, sCOS: cosines, CORC: constrained Pearsoncorrelation, sCORC: constrained Pearson-correlations, SR: Spearman, sSR: Spearmans, MSD: Mean Squared Differences, sMSD: Mean Squared Differencess

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	h_u
u_1			5			5				3	0.43
u_2	2	5				5	2		3		1.00
u_3			4			3				4	0.41
u_4		1	3	2	5					4	0.57
u_5	4	5		4	5	4	4	5		5	1.00
s_i	0.24	0.44	0.48	0.24	0.4	0.68	0.24	0.2	0.12	0.64	
										Precision	0.69

Table 14
Main parameters of the databases used in the experiments.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	x_u
u_1			5			5				3	0.41
u_2	2	5				5	2		3		1.00
u_3			4			3				4	0.43
u_4		1	3	2	5					4	0.62
u_5	4	5		4	5	4	4	5		5	0.43
s_i	0.24	0.44	0.48	0.24	0.4	0.68	0.24	0.2	0.12	0.64	
										Recall	0.71

Table 15
Main parameters used in the experiments.

		Improvements					Databases	
		Similarity metrics					Movielens	Netflix
Quality measures	MAE	COR-sCOR	COS-sCOS	CORC-sCORC	MSD-sMSD	SR-sSR	Fig. 2	Fig. 3
	Coverage	sCOR-COR	sCOS-COS	sCORC-CORC	sMSD-MSD	sSR-SR		
	Precision	sCOR-COR	sCOS-COS	sCORC-CORC	sMSD-MSD	sSR-SR		
	Recall	sCOR-COR	sCOS-COS	sCORC-CORC	sMSD-MSD	sSR-SR		
Quality measures with significance	MAE_s	COR-sCOR	COS-sCOS	CORC-sCORC	MSD-sMSD	SR-sSR	Fig. 4	Fig. 5
	$Coverage_s$	sCOR-COR	sCOS-COS	sCORC-CORC	sMSD-MSD	sSR-SR		
	$Precision_s$	sCOR-COR	sCOS-COS	sCORC-CORC	sMSD-MSD	sSR-SR		
	$Recall_s$	sCOR-COR	sCOS-COS	sCORC-CORC	sMSD-MSD	sSR-SR		

The $recall_s$, either for a user, x_u (see Eq. (56)), or for the whole RS (see Eq. (57)), x, is defined as the ratio of true-positive significances to the total relevant significances.

$$x_u = \frac{\sum_{i \in Y_u} s_{u,i}}{\sum_{i \in Y_u} s_{u,i} + \sum_{i \in O_u} s_{u,i}}; \tag{56}$$

$$x = \frac{1}{\#U} \sum_{u \in U} x_u; \tag{57}$$

4. Experimental evaluation

4.1. Planning the experiments

In this section, we will compare some similarity functions (Pearson correlation, cosine, Spearman, Mean Squared Differences and Constrained Pearson) with their corresponding s-similarity functions (Pearson $correlation_s$, $cosine_s$, $Spearman_s$, Mean Squared $Differences_s$ and Constrained $Pearson_s$) through quality measures of predictions (MAE, MAE_s coverage, $coverage_s$), which we have seen in Section 3.4, and the quality measures of recommendations (precision, $precision_s$, recall and $recall_s$), which we have seen in Section 3.5.

Table 16

	MovieLens	NetFlix
#Users	6040	480189
#Movies	3706	17770
#Ratings	1000209	100480507
Min & max values	1 to 5	1 to 5

Table 17

	K(MAE$_s$,coverage$_s$)		Precision$_s$ Recall$_s$			Common parameters			
	Range	Step	N	θ	K	Test users (%)	Test items (%)	Relevant ratings (V)	#Similar items (z)
Movielens 1 M	{50,...,750}	50	{1,...,15}	4	160	20	20	{4,5}	100
Netflix	{50,...,750}	50	{1,...,15}	5	160	5	20	{4,5}	100

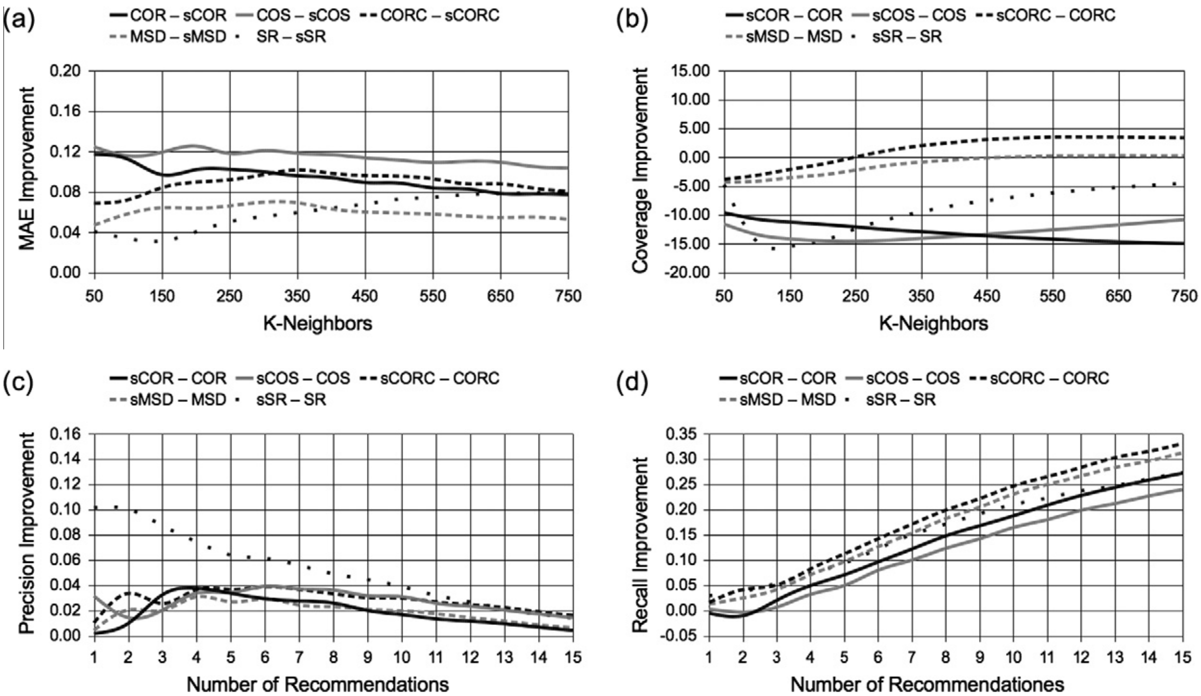

Fig. 2. Traditional quality results using Movielens 1 M: (a) MAE improvements, (b) coverage improvements, (c) precision improvements, (d) recall improvements, $K \in [50,\ldots,750]$ step 50, $V = \{4,5\}$, $z = 100$, 20% test users, 20% test items.

To illustrate that the *s*-metrics provide better results compared with the traditional metrics (with no consideration of significance), the quality measures will be calculated with and without significance levels for both *s*-metrics and metrics. The levels of improvement will be shown in graphs.

Table 13 sums up the combinations of metrics and *s*-metrics with traditional quality measures and the quality measures with significances, which will be evaluated in the experiments. The results (shown in four figures) are calculated considering different subsets of the database.

The experiments will be carried out on both Movielens 1 M and on Netflix, with the aim of providing a general aspect to the results obtained. Table 14 shows the general characteristics of these two databases.

We will perform experiments that consider different numbers of neighbours (*k*-neighbours), from 50 to 750. In the cross-validation process, the number of test items will be the same in all of the experiments; however, due to the high volume of data in Netflix, the number of test users involved will be lower in Netflix (5%) than in Movielens (20%). In Table 15, we show fundamental parameters associated with the experiments (see Tables 16 and 17).

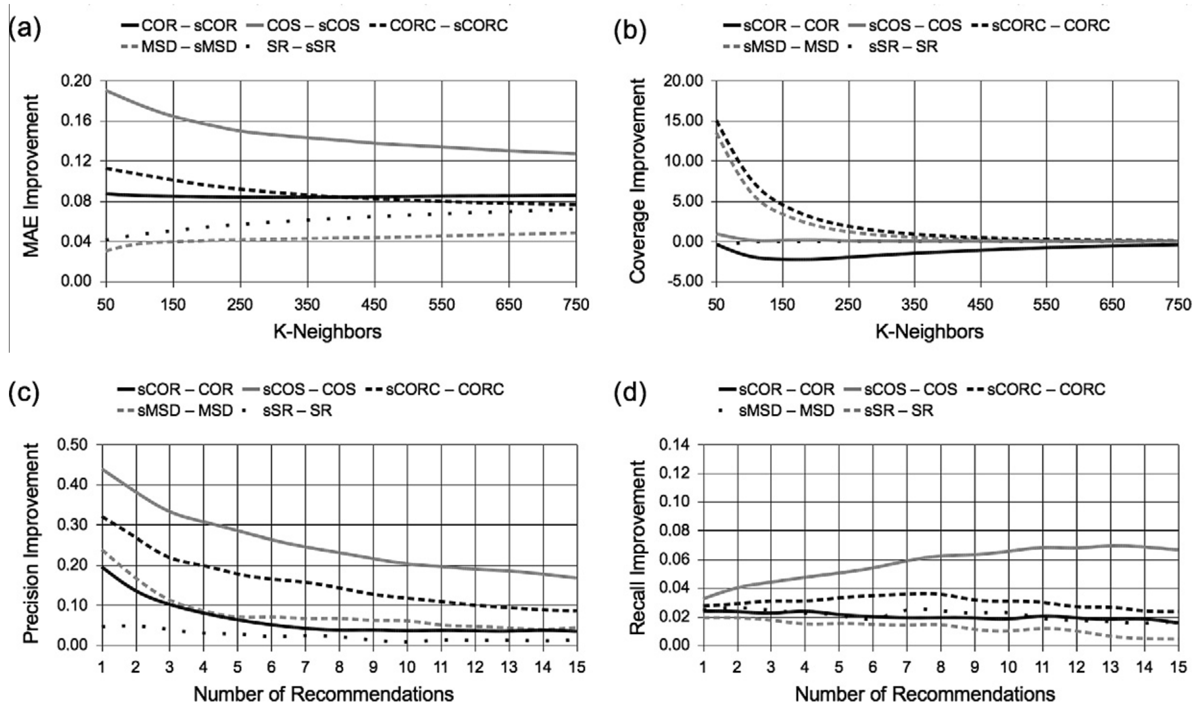

Fig. 3. Traditional quality results using Netflix, (a) MAE improvements, (b) coverage improvements, (c) precision improvements, (d) recall improvements, $K \in [50,\ldots,750]$ step 50, $V = \{4,5\}$, $z = 100$, 5% test users, 20% test items.

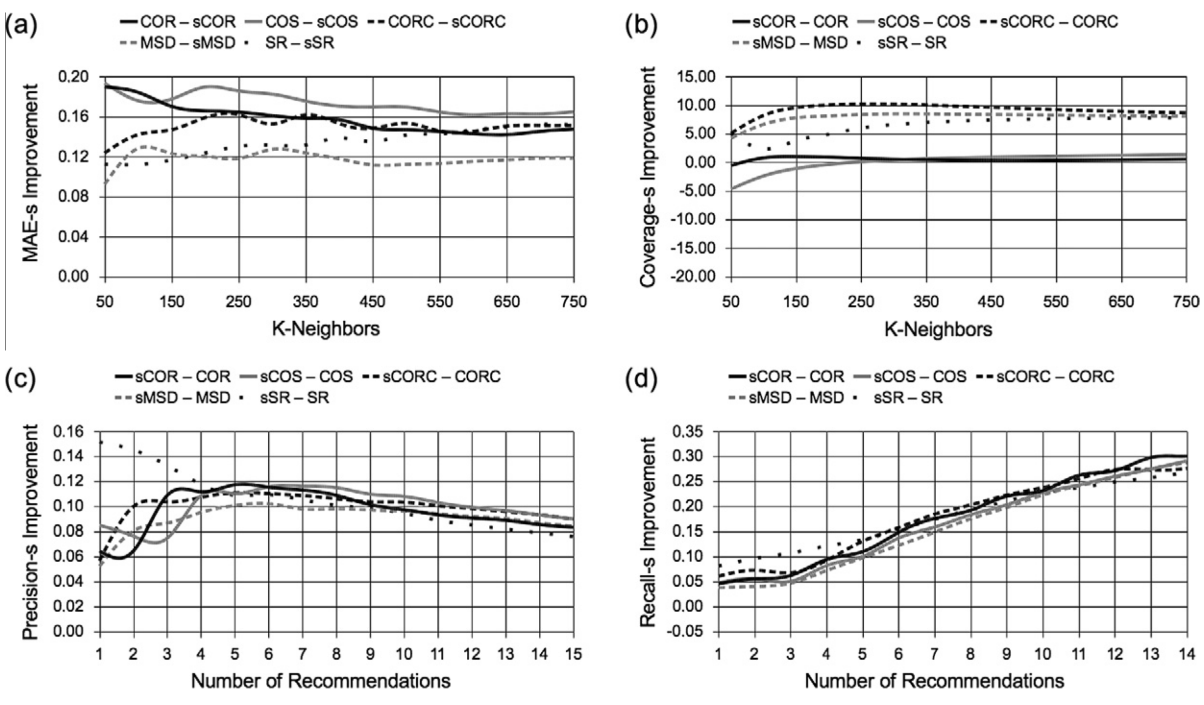

Fig. 4. Significance quality results using Movielens 1 M, (a) MAE improvements, (b) coverage improvements, (c) precision improvements, (d) recall improvements, $K \in [50,\ldots,750]$ step 50, $V = \{4,5\}$, $z = 100$, 20% test users, 20% test items.

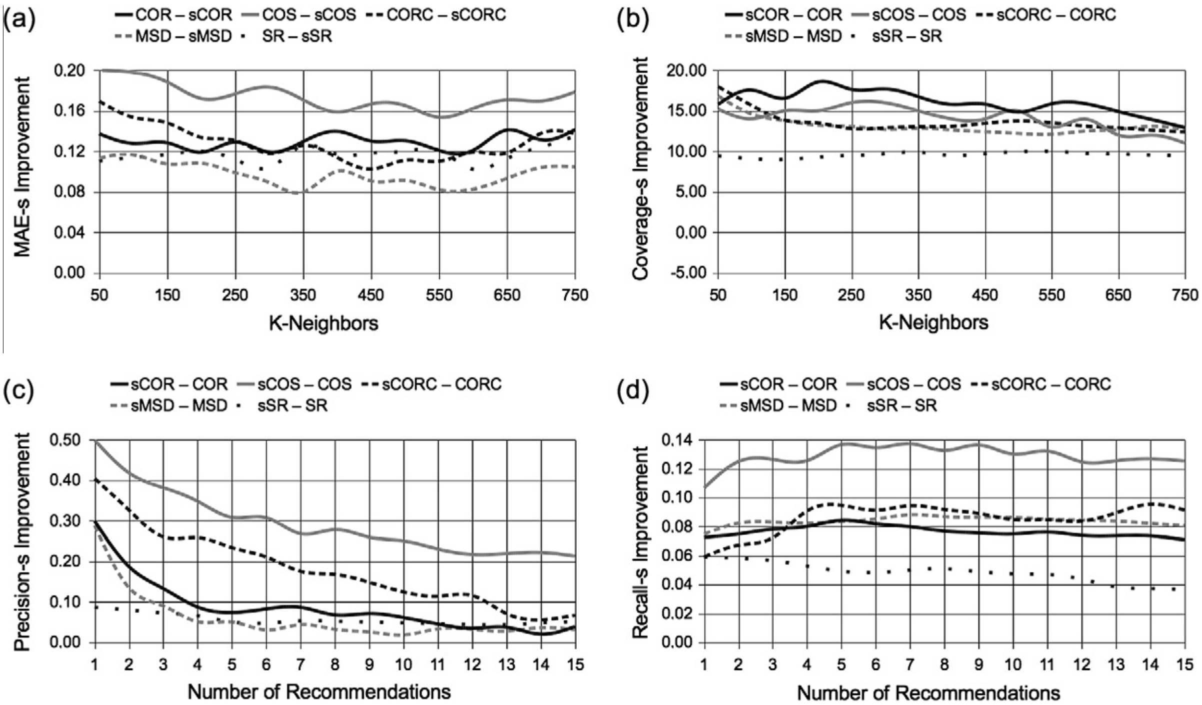

Fig. 5. Significance quality results using Netflix, (a) MAE improvements, (b) coverage improvements, (c) precision improvements, (d) recall improvements, $K \in [50, \ldots, 750]$ step 50, $V = \{4, 5\}$, $z = 100$, 5% test users, 20% test items.

4.2. Results

4.2.1. The s-metrics improvement results using traditional quality measures tested on Movielens 1 M

Fig. 2a shows improvements in accuracy with all of the metrics and for every number of neighbours (*k*-neighbours). The MAE measure obtained, commonly ranging from 0.75 to 0.85, involves an average improvement of 10 % over the traditional metrics.

Graph 2c shows an improvement extending the results in the precision of the *s*-metrics tested. The precision measure commonly ranges between 0.33 and 0.40, involving an average improvement of 10% in precision.

Graph 2d shows an outstanding improvement in the recall measure, especially when a large number of recommendations is made (*N* is high). The recall measure often ranges between 0.10 and 0.45, involving an average improvement of over 50%.

These improvements, considered in terms of the MAE, precision and recall, show that our significance-based metrics can be considered better than traditional metrics. The main drawback of significance metrics is a lower coverage, as may be seen in Graph 2b.

Fig. 2. Traditional quality results using Movielens 1 M: (a) MAE improvements, (b) coverage improvements, (c) precision improvements, (d) recall improvements, $K \in [50, \ldots, 750]$ step 50, $V = \{4, 5\}$, $z = 100$, 20% test users, 20% test items.

4.2.2. The s-metrics improvement results using traditional quality measures tested on Netflix

Because the results obtained with the database Netflix are very similar (with some variations) to those obtained with Movilens, we can be sure that our significance-based measures should also be considered as better than traditional measures.

Graph 3a shows improvements similar to the improvements described in graph 2a. The MAE calculated with Netflix is a little lower than the MAE calculated with Movielens, implying that the MAE improvements (graph 3a) could also be lower (in relation to graph 2a).

The improvements in precision obtained with Netflix (graph 3c) are better than the improvements calculated with Movielens (graph 2c). While the improvement in the recall measure (graph 3d) is worse, the improvement in the coverage measure with Netflix (graph 3b) is better.

4.2.3. The s-metrics improvement results using quality measures with significances

Figs. 4 and 5 show, respectively, the results in the significance quality measure when using Movielens and Netflix.

As may be seen in all of the graphs, the results obtained using our significance-based measures provide an improvement over those obtained from traditional quality measures (Figs. 2 and 3). This shows the ability of the significance quality measures to detect the best similarity metrics.

Thanks to the capacity of the *s*-metrics for improving the significance-based quality measures (they take into account the importance of the items), the *s*-metrics improve the results in these quality metrics (using significances).

5. Conclusions

In this paper, we have proposed some similarity functions, called *s*-metrics, that take into account the importance of items and the importance of users to make recommendations. We have seen that these *s*-metrics involve an improvement in the results over those obtained using traditional metrics.

We have also proposed significance-based quality measures. These measures evaluate the *s*-metrics while considering the importance of items and of ratings in the RS. Our experiments show that our significance-based metrics improve the recommendation results, particularly when dealing with the most significant items of the system.

Likewise, the results of the experiments demonstrate that if we apply the concept of significances to RS, we can improve the quality of the predictions and the recommendations as compared with the qualities obtained in traditional RS. This fact is made clear by comparing traditional metrics and significance-based metrics on the grounds of the traditional prediction and recommendation quality measures.

The significance quality and significance similarity measures proposed provide a generalisation of the traditional CF. With respect to this generalisation, it is sufficient to assign a value of 1.0 to each of the significances in the CF method presented here to obtain the results of the CF without significances.

As for future work using significance-based CF methodology, we will attempt to identify and test different RSs in which the relative importance of the items can be determined: most widely sold products in e-commerce RS, most widely commented news items in news RS, most widely watched programs in TV RS, and the latest champions in sports RS are among the variety of applications available for performing studies with similar recommender systems.

Acknowledgements

Our acknowledgement goes to the *GroupLens Research Group, to* the *FilmAffinity.com*& *NetFlix* companies and to Elsevier Knowledge Based Systems Journal.

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. Knowledge Data Eng. 17 (6) (2005) 734–749.
[2] N. Antonopoulus, J. Salter, Cinema screen recommender agent: combining collaborative and content-based filtering, IEEE Intell. Syst. (2006) 35–41.
[3] L. Baltrunas, F. Ricci, Dynamic item weighting and selection for collaborative filtering, Web Mining 2.0 (2007).
[4] N. Barbieri, M. Guarascio, E. Ritacco, An empirical comparison of collaborative filtering approaches on Netflix data, Italian Information Retrieval (IIR) (2010) 23–27.
[5] A.B. Barragáns, E. Costa, J.C. Burguillo, M. Rey, F.A. Mikic, A. Peleteiro, A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition, Inform. Sci. 180 (22) (2010) 4290–4311.
[6] J. Bobadilla, F. Serradilla, A. Hernando, Collaborative filtering adapted to recommender systems of e-learning, Knowledge Based Syst. 22 (2009) 261–265, doi:10.1016/j.knosys.2009.01.008.
[7] J. Bobadilla, F. Serradilla, The incidence of sparsity on collaborative filtering metrics, Australian DataBase Conference (ADC) 92 (2009) 9-18.
[8] J. Bobadilla, F. Serradilla, J. Bernal, A new collaborative filtering metric that improves the behavior of recommender Systems, Knowledge-Based Syst. 23 (2010) 520–528, doi:10.1016/j.knosys.2010.03.009.
[9] J. Bobadilla, F. Ortega, A. Hernando, A collaborative filtering similarity measure based on singularities, Information Processing and Management, doi:10.1016/j.ipm.2011.03.007, (2011) in press.
[10] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, 14th Conf. on Uncertain. in Artif. Intell., Morgan Kaufmann (1998) 43-52.
[11] L. Candillier, F. Meyer, M. Boullé, Comparing state-of-the-art collaborative filtering systems, LNAI 4571 (2007) 548–562.
[12] P. Ciaccia, M. Patella, Metric information filtering, Information Sci. 36 (4) (2011) 708–720.
[13] H. Denis, Managing collaborative learning processes, e-learning applications, 29th International Conference on Information Technology Interfaces (2007) 345–350.
[14] Y. Ding, X. Li, Time weight collaborative filtering, CIKM (2005) 485–492.
[15] D.R. Liu, Ch.H. Lai, W.J. Lee, A hybrid of sequential rules and collaborative filtering for product recommendation, Information Sci. 179 (20) (2009) 3505–3519.
[16] Fuyuki, T.K. Quan, H. Shinichi, Improving accuracy of recommender systems by clustering items based on stability of user similarity, IEEE International Conference on Intelligent Agents, Web Technologies and Internet Commerce (2006) 61-61, doi: 10.1109/CIMCA.2006.123.
[17] L.Q. Gao, C. Li, Hybrid personalized recommended model based on genetic algorithm, Int. Conf. on Wireless Commun. Netw. and Mob. Computing (2008) 9215-9218.
[18] G.M. Giaglis, Lekakos, Improving the prediction accuracy of recommendation algorithms: approaches anchored on human factors, Interact. Comput. 18 (3) (2006) 410–431.
[19] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, Inf. Retr. 4 (2) (2001) 133–151.
[20] J.L. Herlocker, J.A. Konstan, A. Borchers, J.T. Riedl, An algorithmic framework for performing collaborative filtering, SIGIR (1999) 230–237.
[21] J.L. Herlocker, J.A. Konstan, J.T. Riedl, An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms, Inform. Retriev. 5 (2002) 287–310.
[22] J.L. Herlocker, J.A. Konstan, J.T. Riedl, L.G. Terveen, Evaluating collaborative filtering recommender Systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53.
[23] F. Hernández, E. Gaudioso, Evaluation of recommender systems: a new approach, Expert Syst. Appl. 35 (3) (2008) 790–804.

[24] M.H. Hsu, Proposing an ESL recommender teaching and learning system, Expert Syst. Appl. 34 (3) (2008) 2102–2110.
[25] B. Jeong, J. Lee, H. Cho, User credit-based collaborative filtering, Expert Syst. Appl. 36 (3) (2009) 7309–7312.
[26] B. Jeong, J. Lee, H. Cho, Improving memory-based collaborative filtering via similarity updating and prediction modulation, Inform. Sci. 180 (5) (2010) 602–612.
[27] R. Jin, J. Chai, L. Si, An automatic weighting scheme for collaborative filtering, SIGIR (2004) 337–344.
[28] Y. Koren, Collaborative filtering with temporal dynamics, 15th ACM SIGKDD (2009) 447-456.
[29] G. Koutrika, B. Bercovitz, H. Garcia, FlexRecs: expressing and combining flexible recommendations, SIGMOD (2009) 745–757.
[30] B. Krulwich, Lifestyle finder: intelligent user profiling using large-scale demographic data, J.B. Artificial Intell. Mag. 18 (2) (1997) 37–45.
[31] C.W. Leung, S. Chi-fai, F.L. Chung, An empirical study of a cross-level association rule mining approach to cold-start recommendations, Knowledge Based Syst. 21 (7) (2008) 515–529.
[32] S.K. Lee, Y.H. Cho, S.H. Kim, Collaborative filtering with ordinal scale-based implicit ratings for movile music recommendation, Inform. Sci. 180 (11) (2010) 2142–2155.
[33] Y.M. Li, Ch.P. Kao, A trust-based recommender system for peer production services, Expert Syst. Appl. 36 (2) (2009) 3263–3277.
[34] Y. Manolopoulus, A. Nanopoulus, A.N. Papadopoulus, P. Symeonidis, Collaborative recommender systems: combining effectiveness and efficiency, Expert Syst. Appl. 34 (4) (2007) 2995–3013, doi:10.1016/j.eswa.2007.05.013.
[35] M. Papagelis, D. Plexousakis, T. Kutsuras, Alleviating the sparsity problem of collaborative filtering using trust inferences, Lect. Notes Comput. Sci. 3477 (2005) 224–239.
[36] G. Pitsilis, L.F. Marshall, Modeling trust for recommender systems using similarity metrics, IFIP Adv. Inform. Commun. Technol. 263 (2008) 103–118.
[37] J.L. Sanchez, F. Serradilla, E. Martinez, & J. Bobadilla, Choice of metrics used in collaborative filtering and their impact on recommender systems, IEEE International Conference on Digital Ecosystems and Technologies (DEST) (2008) 432–436 doi:10.1109/DEST.2008.4635147.
[38] Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, The Adaptive Web, LNCS 4321 (2007) 291-324.
[39] J. Serrano, E.H. 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, Inform. Sci. 181 (8) (2011) 1503–1516.
[40] P. Victor, Ch. Cornelis, M. De Cock, P.P. da Silva, Gradual trust and distrust in recommender systems, Fuzzy Sets Syst. 160 (10) (2009) 1367–1382.
[41] W. Yuan, D. Guan, Y.K. Lee, S. Lee, S.J. Hur, Improved trust-aware recommender system using small-worldness of trust networks, Knowledge-Based Syst. 23 (3) (2010) 232–238.
[42] S. Zhang, A. Chakrabarti, J. Ford, F. Makedon, Attack detection in time series for recommender systems, 12th ACM SIGKDD (2006) 809-814.