A collaborative filtering similarity measure based on singularities

Jesús Bobadilla , Fernando Ortega, Antonio Hernando

ABSTRACT

Recommender systems play an important role in reducing the negative impact of information overload on those websites where users have the possibility of voting for their preferences on items. The most normal technique for dealing with the recommendation mechanism is to use collaborative filtering, in which it is essential to discover the most similar users to whom you desire to make recommendations. The hypothesis of this paper is that the results obtained by applying traditional similarities measures can be improved by taking contextual information, drawn from the entire body of users, and using it to calculate the singularity which exists, for each item, in the votes cast by each pair of users that you wish to compare. As such, the greater the measure of singularity result between the votes cast by two given users, the greater the impact this will have on the similarity. The results, tested on the Movielens, Netflix and FilmAffinity databases, corroborate the excellent behaviour of the singularity measure proposed.

1. Introduction

In recent years, recommender systems (RS) have played an important role in reducing the negative impact of information overload on those websites where users have the possibility of voting for their preferences on a series of articles or services. Movie recommendation websites are probably the most well-known cases to users and are without a doubt the most well studied by researchers (Antonopoulus & Salter, 2006; Konstan, Miller, & Riedl, 2004; Li & Yamada, 2004), although there are many other fields in which RS have great and increasing importance, such as e-commerce (Jinghua, Kangning, & Shaohong, 2007) and e-learning (Bobadilla, Serradilla, & Hernando, 2009; Denis, 2007).

Currently, the fast increase of Web 2.0 (Knights, 2007; Lin, 2007) has led to the proliferation of collaborative websites. The number of elements that can be recommended (e.g. blogs, music, videogames, videos, photographs) can increase significantly when introduced (and not only voted) by the users. This new Web 2.0 approach generates new challenges for researchers in the field of RS, at the same time as it increases the possibilities and importance of these information retrieval techniques.

RS makes it possible for each user to obtain the most relevant information in a personalised way. Conceptually, the way they work is very simple; a filtration process is performed for items using one of the following models:

- Content-based filtering (Antonopoulus & Salter, 2006; Lang, 1995) makes recommendation based on user choices which has made in the past (e.g. in a web-based e-commerce RS, if the user purchased some fiction films in the past, the RS will probably recommend a recent fiction film that he has not yet purchased on this website).

- Demographic filtering (Krulwich, 1997) based RS are justified on the principle that individuals with certain common personal attributes (sex, age, country, etc.) will also have common preferences.
- Collaborative filtering (CF) based RS (Adomavicius & Tuzhilin, 2005; Herlocker, Konstan, Riedl, & Terveen, 2004; Herlocker, Konstan, Borchers, & Riedl, 1999; Nayak, Weng, & Xu, 2005; Candillier, Meyer, & Boullé, 2007) allow users to give ratings about a set of items (e.g. videos, songs, films, etc. in a CF based website), in such a way that when enough information is stored on the system we can make recommendations to each user based on information provided by those users we consider to have the most in common with them. CF is an interesting open research field (Chen, Wang, & Zhang, 2009; Leung, Chan, & Chung, 2008; Qing, Sung, Myaeng, & Man, 2007; Ryan & Bridge, 2006).
- The RS hybrid user models (Al-Shamri & Bharadwaj, 2008; Gao & Li, 2008) commonly use a combination of CF with demographic filtering or CF with content based filtering, to exploit merits of each one of these techniques.

Regardless of the method used in the CF stage, the technical objective generally pursued is to minimize the prediction errors, by making the accuracy (Fuyuki, Quan, & Shinichi, 2006; Giaglis & Lekakos, 2006; Manolopoulus, Nanopoulus, Papadopoulus, & Symeonidis, 2007) of the RS as high as possible.

A key factor in the quality of the recommendations obtained in a CF based RS lies in its capacity to determine which users have the most in common (are the most similar) to a given user. A series of algorithms and metrics (Herlocker, Konstan, & Riedl, 2002; Sanchez, Serradilla, Martinez, & Bobadilla, 2008) of similarity between users are currently available which enable this important function to be performed in the CF core of this type of RS.

Let: U set of users, I set of items. The prediction of a non-rated item i for a user u is computed as an aggregate of the ratings of the K most similar users (k-neighbors) for the same item i, where K_u denotes the set of k-neighbors of u and $r_{n,i}$ denotes de value of the user n rating on the item i (if there is not rating value).

Once the set of K users (neighbors) similar to active u has been calculated, in order to obtain the prediction of item i on user u, one of the following aggregation approaches is often used: the average (2), the weighted sum (3) and the adjusted weighted aggregation (deviation-from-mean) (4).

$$G_{u,i} = \{n \in K_u | \exists r_{n,i} \neq \bullet\}, \quad \text{set of } k\text{-neighbors which have rated the item '}i\text{'}. \tag{1}$$

$$p_{u,i} = \frac{1}{\# G_{u,i}} \sum_{n \in G_{u,i}} r_{n,i} \iff G_{u,i} \neq \emptyset \tag{2}$$

$$p_{u,i} = \mu_{u,i} \sum_{n \in G_{u,i}} sim(u,n) r_{n,i} \iff G_{u,i} \neq \emptyset \tag{3}$$

$$p_{u,i} = \bar{r}_u + \mu_{u,i} \sum_{n \in G_{u,i}} sim(u,n)(r_{n,i} - \bar{r}_n) \iff G_{u,i} \neq \emptyset \tag{4}$$

where μ serves as a normalizing factor, usually computed:

$$\mu_{u,i} = 1 \Big/ \sum_{n \in G_{u,i}} sim(u,n) \iff G_{u,i} \neq \emptyset \tag{5}$$

The most popular similarity metrics are Pearson correlation (6), cosine (7), constrained Pearson's correlation (8) and Spearman rank correlation (9).

$$sim(x,y) = \frac{\sum_i (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_i (r_{x,i} - \bar{r}_x)^2 \sum_i (r_{y,i} - \bar{r}_y)^2}} \tag{6}$$

$$sim(x,y) = \frac{\sum_i r_{x,i} r_{y,i}}{\sqrt{\sum_i r_{x,i}^2} \sqrt{\sum_i r_{y,i}^2}} \tag{7}$$

$$sim(x,y) = \frac{\sum_i (r_{x,i} - r_{med})(r_{y,i} - r_{med})}{\sqrt{\sum_i (r_{x,i} - r_{med})^2} \sqrt{\sum_i (r_{y,i} - r_{med})^2}}, \quad r_{med}: \text{ median value in the rating scale} \tag{8}$$

$$sim(x,y) = \frac{\sum_i (rank_{x,i} - \overline{rank}_x)(rank_{y,i} - \overline{rank}_y)}{\sqrt{\sum_i (rank_{x,i} - \overline{rank}_x)^2 \sum_i (rank_{y,i} - \overline{rank}_y)^2}} \tag{9}$$

The work on this field can be regarded to have been started with the definition of context-aware computing stated in Dey (2001): "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task". In our case, the context-aware information is related to the ratings made by the users in the recommender system.

Most papers dealing with contextual information in RS make use of information additional to the own ratings, such as descriptive features of the items involved (Baltrunas, 2008) and physical data sources: noise, temperature, humidity (Park, Yoo, & Cho, 2006), audio features (Li, Myaeng, & Kim, 2007). We are instead concerned with the more general case (and therefore intended for a wider use) of obtaining the contextual information through of the hidden attributes of the users' ratings.

Distributional clustering (Pereira, Tishby, & Lee, 1993) and statistical techniques, such as probabilistic Latent Semantic analysis (pLSA) (Hofmann, 2004) are some of the most usual approaches for managing hidden attributes. In these approaches, a RS model is developed before the prediction and recommendation phase is started.

The main contribution of own work is related to the use of hidden attributes in the collaborative filtering process. By including this information in the similarity metric between users, not only a more simplified approach is obtained, but also in a wider range of applications, and resulting in a better RS performance as compared to that of the previous approaches.

The rest of the paper is divided into sections ordered according to the level of abstraction of the concepts they cover. Each numerical value on the list indicates its corresponding section number.

2. Motivation: why it is important the proposed singularities measure?
3. Singularities concept details.
4. Mathematical formalism defining the proposed singularities measure: Basic and extended models.
5. Designing suitable experiments for testing the proposed metric.
6. General discussions of the results.
7. Most relevant conclusions obtained and future work.

2. Motivation

The core of the CF stage in a RS is the choice of *k*-neighbors to the user to whom you wish to make recommendations. This process is often achieved by using similarity measures which do not make use of all of the available information, and what is more, the information that is used (votes in common for each pair of users analysed) ends up being much less relevant than expected. We will try to improve results using contextual information (drawn from the entire body of users) as well as categorizing the rating values.

2.1. Using contextual information

The traditional similarity measures between two users (Pearson correlation, cosine, constrained Pearson correlation, Spearman rank, mean squared differences, etc.) are calculated taking into account only the ratings made by these two users. Although they are very efficiently calculated, they ignore the context for the ratings made by the users. In this paper, we will consider this context by analysing the ratings of each item made by all users.

We worth more if the votes cast by two users are similar for one item if, in turn, they are different for this item to votes made by the majority of the other users, as that indicates a similarity which is difficult to find between the user in question and the rest. On the other hand, if the majority of users have voted for an item in a similar way, the similarity (with regards to this item) between two users who have voted for this item in the same way as the rest should be considered not very relevant. This concept is illustrated by the following example: imagine that in a RS for electronic equipment, the vast majority of users who have voted for a specific item (for example a state-of-the-art mobile phone) have done so in a positive or very positive way; in this context, two users who have voted negatively for this item present a relative similarity (with regards to this item) which is much greater than two users who have voted positively for it.

2.2. Categorizing rating values

The majority of RS in current use offer a moderate range of possible values for casting a vote (normally in the range of $1,\ldots,5$). In these RS, users tend to divide their evaluations conceptually into "positive" or "non-positive", where a 4 will generally represent a positive rating, which in some cases will be reinforced with the rating 5. Similarly, a 3 will represent a non-positive rating, which in some cases will be reinforced with the rating 2 or 1. In order to test this hypothesis, we have designed an experiment (Dey, 2001) on the MovieLens 1M database: we transformed all 4 and 5 votes into *P* votes (Positive) and all of 1, 2 and 3 votes into *N* votes (non-positive), in such a way that we aim to measure the impact made on the recommendations by doing without the detailed information provided by the numerical values of the votes.

In the experiment we compare the precision/recall obtained in a regular way (using the numerical values of the votes) with that obtained using only the discretized values *P* and *N*; for this purpose, we establish the relevance threshold at value 4 ($\theta = 4$), assimilating "relevant" with "positive"; we use Pearson correlation, deviation from mean aggregation approach, 20% of test users, 20% of test items, number of recommendations from 2 to 20, $K = 150$. The experiment has been repeated for values between $K = 100$ and $K = 200$, obtaining equivalent results.

Fig. 1 displays the results, which show how the "positive/non-positive" discretization not only does not worsen the precision/recall measurements, but rather it improves them both, particularly the precision when the number of recommendations (N) is high. The numerical key to this improvement lies in the improved capacity of the discrete calculation to

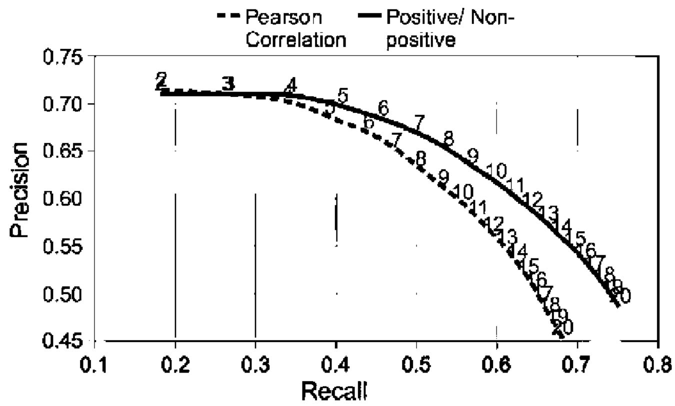

Fig. 1. Precision/recall obtained by transforming all 4 and 5 votes into *P* votes (positive) and all 1, 2 and 3 votes into *N* votes (non-positive), compared to the results obtained using the numerical values. 20% of test users, 20% of test items, $K = 150$, Pearson correlation, $\theta = 4$, database: MovieLens 1M, (Bobadilla, Serradilla, & Bernal, 2010).

determine whether an item is recommended (based on the number of *k*-neighbors with value *P* in that item), as regards the calculation with numerical values (prediction obtained by applying the selected aggregation approach on the numerical values of the votes and their subsequent thresholding).

If we consider the results above, a simple way of focusing the question of whether two users have cast “similar” votes for an item and if, in turn, those votes are “similar” to those made by the rest of the users for this item, consists of dividing the entire set of possible values of votes into “positives/non-positives”, as with the preceding experiment; in this respect, we could consider those votes which belong to the same category (“positive” or “non-positive”) as “similar”.

3. Incorporation of the concept of singularity

The main idea of this paper lies in the fact that the contribution of an item to the similarity assigned to two users ought not to be considered as absolute (which is what happens with traditional metrics), but rather as relative to the vote awarded to this item by the rest of the users in the system.

The basis of our hypothesis is that the value of the similarity must be modulated by the value of the singularity, in such a way that a very singular similarity should be awarded a higher value than a very normal similarity. Going back to the previous example, if 95% of users voted positively for the item, the similarity derived (for this item) between two users who belong to the 5% (very singular) must be greater than the similarity derived between two users who belong to the 95% (not very singular).

Following the reasoning expressed through to its extremes, if all of the users who have voted for an item have cast the same vote, it is difficult to consider this item as a factor of similarity between two users. In the opposite case scenario, if only two users have voted differently to the rest for one item, this represents a very great singularity which should be translated into a very great similarity for this item.

Modulating the values for similarity with those for singularity among the whole group of items in a RS, we will be able to incorporate a very valuable source of information which complements the isolated values which are used by traditional similarities measures.

In Table 1 you will see five examples (one for each item) of different situations which might arise in applying the concept of singularity. Table 1 represents a hypothetical portion of a RS, where possible votes are in the range of $1, \ldots, 5$. The votes in bold symbolise positive evaluations, while the votes in italic symbolise non-positive evaluations; we are trying to find the similarity between users 1 and 6.

Considering item 1, we can deduce that the similarity between users 1 and 6 should be established as very high as they are the only users to have voted positively for this item (their singularity is high). It can be highlighted that, although their

Table 1
Different singularity cases for items. Votes in bold: positive values; votes in italic: non-positive values.

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	**4**	**5**	**5**	*2*	*2*
User 2	*2*	**4**	**4**	**4**	*1*
User 3	*1*	**5**	**4**	**4**	*2*
User 4	*3*	**5**	**5**	**5**	*2*
User 5	*2*	**5**	**4**	**4**	**5**
User 6	**5**	**5**	*2*	**4**	**4**
User 7	*2*	**4**	**5**	**5**	**5**
User 8	*1*	**5**	**4**	**4**	**5**

votes do not coincide, their similarity should be established as very high regarding the contribution of item 1 to the total measure of similarity between these two users.

Concerning item 2, we can deduce that the similarity between users 1 and 6 should be established as very low, or even non-existent, as there is no singularity whatsoever (all users have voted "positive"). This is the case even though both users have cast the same vote.

Items 3 and 4 demonstrate the situation in which one user presents a very singular vote and the other user presents a vote with virtually no singularity value. In this case, the similarity between the users should be established as very low, as the similarity between the user with a high level of singularity for the item and any other user is the same.

Item 5 shows the scenario in which the orientation (positive/non-positive) of the votes by users is evenly distributed; these average singularities ought to lead to moderate similarity levels.

The additional singularity information that we propose to incorporate at the heart of the CF stage of RS (the identification of neighborhoods) will bring about improvements in the results, but with a potential theoretical problem in performance, as the singularity values would vary as users continue to cast their votes. The most obvious way of dealing with this situation is to update singularity values periodically, so that in a short interval of time (for example, a period of 15 min), it can be considered that any new votes cast would not significantly affect the millions of votes accumulated over many years in an established RS.

4. Formulation

In this section the mathematical equations which support the proposed similarity measure are presented, which we will call singularity measure (SM). In the first sub-section the parameters, measures and set descriptions used in the equations are specified. In the second sub-section a simple model is defined which is valid for RS based on groups of values for votes of up to five possibilities (for example from one to five "stars"), which make up the majority of existing RS. In the third subsection a generalisation of the model for RS is set out, where you can vote according to any range of votes (for example in the interval $1, \ldots, 10$).

4.1. Parameters, measures and sets descriptions

Below we present the tables of parameters, measures and sets used in the formalizations made in the paper (see Tables 2 and 3).

4.2. Basic model

We define U as the set of the RS users and I as the set of the RS items. We start with all of the votes cast by two generic users x and y, in a RS:

Let $r_{u,i}$ be the rating made by the user u on the item i. In case that the user u has not rated the item i yet, then $r_{u,i} = \bullet$. $r_x = (r_{x,1}, r_{x,2}, r_{x,3}, \ldots.., r_{x,\#I})$, $r_y = (r_{y,1}, r_{y,2}, r_{y,3}, \ldots.., r_{y,\#I})$, where the possible values of votes are defined through the set $V = \{\{m, \ldots, M\}\} \cup \{\bullet\}$, where m is the lowest possible value (usually 1), M is the highest possible (usually 5 or 10) and the symbol represents the absence of a vote.

We define R as the set of relevant values (positives) in a RS, and therefore R^c as the set of non-positive values. For example, in a RS where $m = 1$, $M = 5$, we can define

$$R = \{4,5\}, \quad R^c = \{1,2,3\} \tag{10}$$

Table 2
Parameters and measures.

Name	Parameters and measures descriptions
m	#Min rating value
M	#Max rating value
K	#Neighborhoods
N	#Recommendations
d	Max. relevancy value
$r_{u,i}$	Rating of the user u on the item i
S_P^i	Singularity of the relevant rating on item i
S_N^i	Singularity of the non-relevant rating on item i
s_i^j	Singularity of the rating of relevance j on item i
e_{jq}	Element of the SM matrix (position j, q)
r	Relevant rating
n	Non-relevant-rating
$p_{u,i}$	Prediction to user u on item i
$sim(x, y)$	Users similarity

Table 3
Sets.

Name	Sets descriptions
U	Users
I	Items
V	Rating values
R	Relevant values
R^c	Non-relevant values
P_i	Users who have rated item i as relevant
N_i	Users who have rated item i as non-relevant
A	Items simultaneously rated as relevant by the two compared users
B	Items simultaneously rated as non-relevant by the two compared users
C	Items simultaneously rated as relevant by one user and non-relevant by the other user of the two compared users
$D_{x,y}$	Items simultaneously rated by users x and y
R^j	Set of values of relevance j
P_i^j	Set of users who have rated item i with relevance j
$A_{j,q}$	Items simultaneously rated with relevance j by one user and with relevance q by the other user of the two compared users
K_u	K neighborhoods of the user u
$G_{u,i}$	Users belonging to K_u which have voted item i

We define P_i as the set of users who have assigned to the item i a relevant value (positive).

$$P_i = \{u \in U | r_{u,i} \in R\} \tag{11}$$

Using the example in Table 1:

$$P_1 = \{1,6\}, \quad P_2 = \{1,2,3,4,5,6,7,8\}, \quad P_3 = \{1,2,3,4,5,7,8\}, \quad P_4 = \{2,3,4,5,6,7,8\}, \quad P_5 = \{5,6,7,8\}$$

We define N_i as the set of users who have assigned to the item i a non-relevant value (non-positive).

$$N_i = \{u \in U | r_{u,i} \in R^c\} \tag{12}$$

Using the example in Table 1:

$$N_1 = \{2,3,4,5,7,8\}, \quad N_2 = \emptyset, \quad N_3 = \{6\}, \quad N_4 = \{1\}, \quad N_5 = \{1,2,3,4\}$$

S_P^i is defined as the singularity of the relevant vote concerning the item i; the more users have rated the item i with positive ratings, the lower the singularity S_p^i related to this item will be, and vice versa. Using the example in Table 1: $S_P^1 = 1 - 2/8 = 0.75, S_P^2 = 1 - 8/8 = 0, S_P^3 = 1 - 7/8 = 0.125, S_P^4 = 1 - 7/8 = 0.125, S_P^5 = 1 - 4/8 = 0.5$.

$$s_P^i = 1 - \frac{\#P_i}{\#U} \tag{13}$$

S_N^i is defined as the singularity of the non-relevant vote concerning the item i; the more users have rated the item i with negative ratings, the lower the singularity S_p^i related to this item will be, and vice versa. Using the example in Table 1: $S_N^1 = 1 - 6/8 = 0.25, S_N^2 = 1 - 0/8 = 1, S_N^3 = 1 - 1/8 = 0.875, S_N^4 = 1 - 1/8 = 0.875, S_N^5 = 1 - 4/8 = 0.5$.

$$s_N^i = 1 - \frac{\#N_i}{\#U} \tag{14}$$

$$s_P^i \in [0, \ldots, 1], s_N^i \in [0, \ldots, 1], R \cup R^c = V - \{\bullet\} \Rightarrow s_P^i + s_N^i \Leftarrow 1$$

1 – not very common (very singular) and 0 – very common (not very singular)

Respecting the principle of maintaining symmetry in the metrics used, *sim*(u1, u2) = *sim*(u2, u1), the pairs (user x vote, user y vote) of possible cases of votes by two users concerning an item are shown in the second column of Table 4. The singularity of each case, regarding votes by two users x and y, is shown in the third column of Table 4.

In terms of the example considered in Table 1, the singularities between the users 1 and 6 can be obtained as follows: A = 1, 2, B = , C = 3, 4, 5; item 1: $S_P^1 S_P^1 = 0.75^2 = 0.562$; item 2: $S_P^2 S_P^2 = 0^2 = 0$; item 3: $S_P^3 S_N^3 = 0.125 \cdot 0.875 = 0.109$; item 4: $S_N^4 S_P^4 = 0.875 \cdot 0.125 = 0.109$; item 5: $S_N^5 S_P^5 = 0.5^2 = 0.25$.

As can be observed, a combined measure of singularity has been chosen in which the overall singularity is obtained as the combination of the independent probabilities of each of the singularities provided by the two users compared. Graphically, the singularity obtained is shown in Fig. 2, where the x and z axes indicate the singularities of the votes by the two users compared and the y axis represents the combined singularity obtained. It can be noted that in order to achieve good results of singularity it is necessary to have high values of singularity for the two users compared.

The singularity factors obtained provide us with contextual information with which we can improve similarity results based on the numerical values of votes in common by the users compared. As a numerical similarity measure we use a variation of Mean Absolute Differences: Mean Squared Differences (MSD):

Table 4
Possible combinations of votes by two users x and y and their associated singularity values, where $r \in R$ and $n \in R^c$; the sets A, B and C respectively stand for: those items which have been assigned relevant ratings (A) by both users, those items which have been assigned non-relevant ratings (B) by both users, and those items which have been assigned a relevant rating by a user and a non-relevant rating by the other (C).

Case	Combinations	Singularity
a	(r, r)	$A = \{i \in I \mid r_{x,i} \in R \wedge r_{y,i} \in R\} s_P^i s_P^i$
b	(n, n)	$B = \{i \in I \mid r_{x,i} \in R^c \wedge r_{y,i} \in R^c\} s_N^i s_N^i$
c	(r, n), (n, r)	$C = \{i \in I \mid (r_{x,i} \in R \wedge r_{y,i} \in R^c) \vee (r_{x,i} \in R^c \wedge r_{y,i} \in R)\} s_P^i s_N^i$

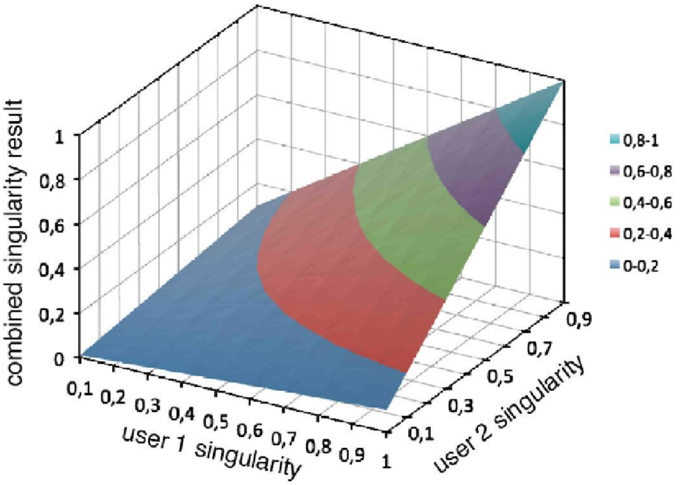

Fig. 2. Singularity results (y axis) obtained from singularity values for each of the two users compared (x and z axis).

First, we perform the normalization on the ratings: $r_{u,i} \in [0,1] \forall u \in U \wedge \forall i \in I | r_{u,i} \neq \bullet$

$$MSD(x,y) = \frac{1}{\#D_{x,y}} \sum_{i \in D_{x,y}} \left[1 - (r_{x,i} - r_{y,i})^2\right], \quad MSD(x,y) \in [0,1] \tag{15}$$

$D_{x,y} = \{i \in I | r_{x,i} \neq \bullet \wedge r_{y,i} \neq \bullet\}$, items rated by both user x and user y.

By combining the equations described, we obtain the proposed *singularities measure* (*SM*):

$$\begin{aligned} SM(x,y) &= \frac{1}{3}\left[\frac{1}{\#A}\sum_{i\in A}[1-(r_{x,i}-r_{y,i})^2](s_P^i)^2 + \frac{1}{\#B}\sum_{i\in B}[1-(r_{x,i}-r_{y,i})^2](s_N^i)^2 + \frac{1}{\#C}\sum_{i\in C}[1-(r_{x,i}-r_{y,i})^2]s_P^i s_N^i\right] \\ &\iff A \neq \emptyset \vee B \neq \emptyset \vee C \neq \emptyset \\ SM(x,y) &= \bullet \iff A = \emptyset \wedge B = \emptyset \wedge C = \emptyset \end{aligned} \tag{16}$$

$$r_{u,i} \in [0,1] \Rightarrow SM(x,y) \in [0,1] \tag{17}$$

The first term in Eq. (16) stands for the similarity between the users x and y on behalf of the items which both of them have rated with relevant values (these items belong to the set A). The second term in Eq. (16) stands for the similarity between the users x and y on behalf of the items which both of them have rated with non-relevant values (these items belong to the set B). As for the third term in the equation, it stands for the similarity between the users x and y on behalf of those items which a user has rated as relevant while the other has rated as non-relevant (these items belong to the set C).

Specifically, the Eq. (16) stands for the similarity between the users x and y on behalf of these three terms:

- Term 1. *Positive agreement*: similarity between the users x and y on behalf of the items which both of them have rated with relevant values, weighting them by the singularity of relevant ratings for each of the items.
- Term 2. *Negative agreement*: similarity between the users x and y on behalf of the items which both of them have rated with non-relevant values, weighting them by the singularity of non-relevant ratings for each of the items.
- Term 3. *Disagreement*: similarity between the users x and y on behalf of those items which a user has rated as relevant while the other has rated as non-relevant, weighting them by the singularity of relevant ratings and non-relevant ratings for each of the items.

The most important difference between the proposed metric *SM*(*x*, *y*) and the traditional ones (*Pearson*, *MSD*) is related to introducing the already explained singularity factors (S_P^i, S_N^i), which makes use of the context information derived from the whole set of ratings in RS, while the traditional similarity measures only use the information related to the set of ratings made by the users *x* and *y*.

4.3. Extended model

When the value of *M* is high in the set of possible votes ($V = \{\{m, \ldots, M\}\} \cup \{\bullet\}$), the division of the votes into the sets "relevant/non-relevant" (R, R^c) is not sufficient. In this case, the votes can be divided into an arbitrary number of sets, for example: "irrelevant at all, not very relevant, slightly relevant, relevant, very relevant". We designate these sets as follows:

$$\{R^j\}, j \in \{1, \ldots, d\}, d \in N, \quad N : \text{natural numbers} \tag{18}$$

where *j* indicates the relevance value (1 – minimum relevance, *d* – maximum relevance), and R^j denotes the set of values for *V* with relevance *j*. In the case of the value for *M* not being high (for example 5), we can use a value of *d* = 2 and obtain the sets: R^1 – set of non-relevant values, R^2 – set of relevant values, which situation corresponds to the simplified model described in the previous sub-section.

P_i^j is defined as the whole set of users who have awarded a relevance value *j* to the item *i*

$$P_i^j = \{u \in U | r_{u,i} \in R^j\} \tag{19}$$

$s_i^j$is defined as the singularity of the vote with relevance *j* for the item *i.*

$$s_i^j = 1 - \frac{\#P_i^j}{\#U} \tag{20}$$

The votes cast for an item *i* by users *x* and *y* can be represented as:

$$(r_{x,i} \in R^j, r_{y,i} \in R^q) | j, q \in \{1, \ldots, d\} \tag{21}$$

The set of items in which the pairs of votes cast is:

$$A_{j,q} = \{i \in I | r_{x,i} \in R^j \wedge r_{y,i} \in R^q\} \tag{22}$$

The extended equation equivalent to (16) is

$$SM(x,y) = \frac{1}{\#G} \sum_{j=1}^{d} \sum_{q=1}^{d} \frac{1}{\#A_{j,q}} \sum_{i \in A_{j,q}} \left[1 - (r_{x,i} - r_{y,i})^2\right] s_i^j s_i^q \quad \forall A_{j,q} \neq \emptyset \tag{23}$$

where

$$G = \{A_{j,q} | j, q \in \{1, \ldots, d\} \wedge A_{j,q} \neq \emptyset\} \tag{24}$$

SM can be represented as a matrix, whereby each element of the matrix is assigned the value of the term affected by the double summation, which we will call e_{jq}:

$$\begin{pmatrix} e_{11} & \cdots & e_{1q} \\ \cdots & \cdots & \cdots \\ e_{j1} & \cdots & e_{jq} \end{pmatrix} \Bigg| e_{jq} = \frac{1}{\#A_{j,q}} \sum_{i \in A_{j,q}} [1 - (r_{x,i} - r_{y,i})^2] s_i^j s_i^q \quad \forall A_{j,q} \neq \emptyset \tag{25}$$

Taking into account that

$$s_i^j s_i^q = s_i^q s_i^j \forall i \in I \wedge \forall j, \quad q \in \{1, \ldots, d\} \tag{26}$$

The number of terms in the matrix (25) can be reduced according to the specifications of Eq. (27):

$$\begin{pmatrix} e_{11} & 0 & 0 \\ \cdots & \cdots & 0 \\ e_{j1} & \cdots & e_{jq} \end{pmatrix} \Bigg| e_{jq} = \frac{1}{\#A_{j,q} + \#A_{q,j}} \sum_{i \in A_{j,q} \cup A_{q,j}} [1 - (r_{x,i} - r_{y,i})^2] s_i^j s_i^q \quad \forall A_{j,q} \cup A_{q,j} \neq \emptyset \tag{27}$$

Finally, Eq. (23) can alternatively be defined as:

$$\begin{aligned} SM(x,y) &= \frac{1}{\#H} \sum_{j=1}^{d} \sum_{q=1, j \geqslant q}^{d} e_{jq} | \\ e_{jq} &= \frac{1}{\#A_{j,q} + \#A_{q,j}} \sum_{i \in A_{j,q} \cup A_{q,j}} [1 - (r_{x,i} - r_{y,i})^2] s_i^j s_i^q \quad \forall A_{j,q} \cup A_{q,j} \neq \emptyset \end{aligned} \tag{28}$$

where

$$H = \{A_{j,q} | j, q \in \{1, \dots, d\} \wedge j > q \wedge (A_{j,q} \cup A_{q,j} \neq \emptyset)\} \tag{29}$$

As an example, using the FilmAffinity database, where $M = 10$, we can establish a value of $d = 3$, and the sets:

$$R^1 = \{1,2,3\}, \quad R^2 = \{4,5,6,7\}, \quad R^3 = \{8,9,10\}$$

By applying Eq. (23), we obtain the following:

$$\begin{aligned} SM(x,y) = \frac{1}{9}\Bigg(& \frac{1}{\#A_{1,1}} \sum_{i \in A_{1,1}} [1 - (r_{x,i} - r_{y,i})^2] s_i^1 s_i^1 + \frac{1}{\#A_{1,2}} \sum_{i \in A_{1,2}} [1 - (r_{x,i} - r_{y,i})^2] s_i^1 s_i^2 + \frac{1}{\#A_{1,3}} \sum_{i \in A_{1,3}} [1 - (r_{x,i} - r_{y,i})^2] s_i^1 s_i^3 \\ & + \frac{1}{\#A_{2,1}} \sum_{i \in A_{2,1}} [1 - (r_{x,i} - r_{y,i})^2] s_i^2 s_i^1 + \frac{1}{\#A_{2,2}} \sum_{i \in A_{2,2}} [1 - (r_{x,i} - r_{y,i})^2] s_i^2 s_i^2 + \frac{1}{\#A_{2,3}} \sum_{i \in A_{2,3}} [1 - (r_{x,i} - r_{y,i})^2] s_i^2 s_i^3 \\ & + \frac{1}{\#A_{3,1}} \sum_{i \in A_{3,1}} [1 - (r_{x,i} - r_{y,i})^2] s_i^3 s_i^1 + \frac{1}{\#A_{3,2}} \sum_{i \in A_{3,2}} [1 - (r_{x,i} - r_{y,i})^2] s_i^3 s_i^2 + \frac{1}{\#A_{3,3}} \sum_{i \in A_{3,3}} [1 - (r_{x,i} - r_{y,i})^2] s_i^3 s_i^3 \Bigg) \end{aligned}$$

By applying Eq. (28), we obtain the following:

$$\begin{aligned} SM(x,y) = \frac{1}{6}\Bigg(& \frac{1}{\#A_{1,1}} \sum_{i \in A_{1,1}} [1 - (r_{x,i} - r_{y,i})^2] s_i^1 s_i^1 + \frac{1}{\#A_{2,1} + \#A_{1,2}} \sum_{i \in A_{2,1} \cup A_{1,2}} [1 - (r_{x,i} - r_{y,i})^2] s_i^2 s_i^1 \\ & + \frac{1}{\#A_{2,2}} \sum_{i \in A_{2,2}} [1 - (r_{x,i} - r_{y,i})^2] s_i^2 s_i^2 + \frac{1}{\#A_{3,1} + \#A_{1,3}} \sum_{i \in A_{3,1} \cup A_{1,3}} [1 - (r_{x,i} - r_{y,i})^2] s_i^3 s_i^1 \\ & + \frac{1}{\#A_{3,2} + \#A_{2,3}} \sum_{i \in A_{3,2} \cup A_{2,3}} [1 - (r_{x,i} - r_{y,i})^2] s_i^3 s_i^2 + \frac{1}{\#A_{3,3}} \sum_{i \in A_{3,3}} [1 - (r_{x,i} - r_{y,i})^2] s_i^3 s_i^3 \Bigg) \end{aligned}$$

5. Experiments

In order to compare our proposed singularity measure (SM) to traditional similarity measure, thus avoiding lengthy comparisons with large groups of traditional similarities measures. As such, the intention has been to show the improvements achieved both in applying prediction qualities measures and in testing with recommendation qualities measures.

5.1. Reference similarity measure

According to the general bibliography in the field of CF RS, Pearson correlation presents the best traditional similarity measure prediction and recommendation results; furthermore, it is the most commonly used, and therefore, any alternative metric proposed must improve its results.

In order to corroborate the proper functioning of Pearson correlation and to discover the aggregation approach and the standardization process which yield the best results, the following experiment was performed on the MovieLens 1 M database, combining:

- *Similarity measures*: metrics set out in Eqs. (6)–(9); therefore: (PC, COS, CPC, SRC).
- *Aggregation approaches*: aggregation approaches set out in Eqs. (3) and (4), which we will call WS and DFM, respectively.

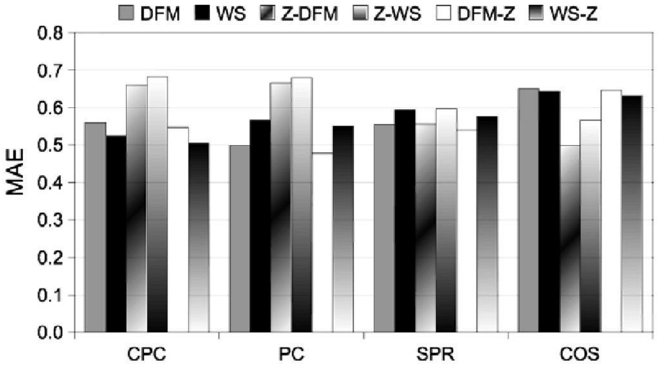

Fig. 3. Comparative MAE results using CPC, PC, SPR and COS similarity measures, WS and DFM aggregation approaches and z-scores applied to input data (Z-) or applied to the similarity values (-Z).

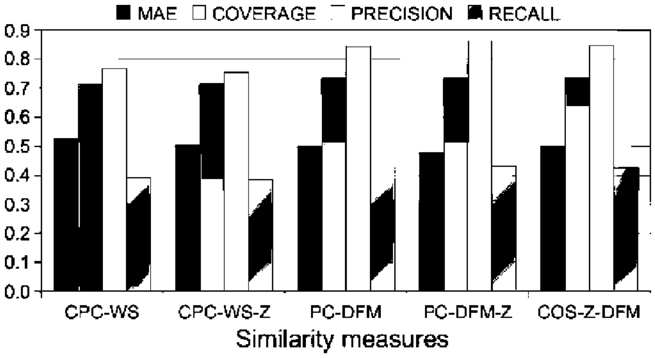

Fig. 4. Comparative MAE, coverage, precision and recall results using de best combinations obtained on Fig. 3: CPC–WS, CPC–WS–Z, PC–DFM, PC–DFM–Z and COS–Z–DFM.

Table 5
Main parameters of the databases used in the experiments.

	Movielens	FilmAffinity	Netflix
#Users	6040	26,447	480,189
#Movies	3706	21,128	17,770
#Ratings	1,000,209	19,126,278	100,480,507
Min and max values	1–5	1–10	1–5

- *Standardization process*: *z*-scores applied to the input data (Z-), *z*-scores applied to the similarity values "*sim*(*u*, *n*)" used in Eqs. (3) and (4) (-Z).
- *Results calculated*: Mean Absolute Error (MAE), coverage, precision and recall.

Fig. 3 contains a graph for each study group showing the results obtained by only calculating the MAE. The combinations that present higher accuracy (lower MAE) are: CPC–WS, CPC–WS–Z, PC–DFM, PC–DFM–Z and COS–Z–DFM.

In order to decide on the best combination, the five combinations selected have been compared by listing all the measures studied (MAE, coverage, precision and recall). Fig. 4 shows the results; we can see how the combinations which present the best values overall are: PC–DFM, PC–DFM–Z and COS–Z–DFM. In turn, of these combinations, the one that offers the best result is PC–DFM–Z, which will be the combination (measure, aggregation approach, standardization process) selected as the reference similarity measure.

5.2. Experiments using the proposed singularity measure (SM)

The experiments have been grouped in such a way that the following can be determined: accuracy, coverage, number of perfect predictions, precision/recall. We consider a perfect prediction to be each situation in which the prediction of the rating recommended to one user in one film matches the value (the stars number) rated by that user for that film.

The RS databases that we use in our experiments present the general characteristics summarized in Table 5.

The experiments were carried out, depending on the size of the database, for each of the following *k*-neighbors values: Movielens 2, ..., 1500 step 50, FilmAffinity 2, ..., 2000 step 100, Netflix 2, ..., 10000 step100, due to the fact that depending on the size of each particular RS database, it is necessary to use a different number of *k*-neighbors in order to display tendencies in the graphs that show their results. The precision/recall recommendation quality results have been obtained using a range 2, ..., 20 of recommendations (*N*) and relevance thresholds $\theta = 5$ using Movielens and Netflix and $\theta = 9$ using FilmAffinity.

When we use MovieLens and FilmAffinity we use 20% of test users taken at random from all the users of the database; with the remaining 80% we carry out the training. When we use Netflix, given the huge number of users in the database, we only use 5% of its users as test users. In all cases we use 20% of test items. Table 6 shows the numerical parameters used in the experiments.

The presentation of the results is divided into two sub-sections: the first uses the basic formulation on the Movielens 1M and Netflix databases, both of which have voting ranges in the interval of 1, ..., 5; the second sub-section shows the results obtained by using the FilmAffinity database (which has a voting range in the order of 1, ..., 10) using both the basic and the extended formulation model.

Overall, four figures are presented (results with Movielens 1M and Netflix using the basic formulation, and with FilmAffinity using both the basic and the extended formulation). Each figure is composed of four graphs: (a) Mean Absolute Error (MAE), (b) coverage, (c) percentage of perfect predictions, d) Precision versus Recall. Each graph displays the results obtained by the proposed metric (SM, as a continuous line) and the reference metric (PC–DFM–Z, as a broken line).

Table 6
Main parameters used in the experiments.

	K (MAE, coverage, perfect predictions)		Precision/ recall			Test users (%)	Test items (%)
	Range	Step	K	N	θ		
Movielens 1M	{2, ..., 1500}	50	150	{2, ..., 20}	5	20	20
FilmAffinity	{2, ..., 2000}	100	180	{2, ..., 20}	5	20	20
Netflix	{2, ..., 10,000}	100	500	{2, ..., 20}	9	5	20

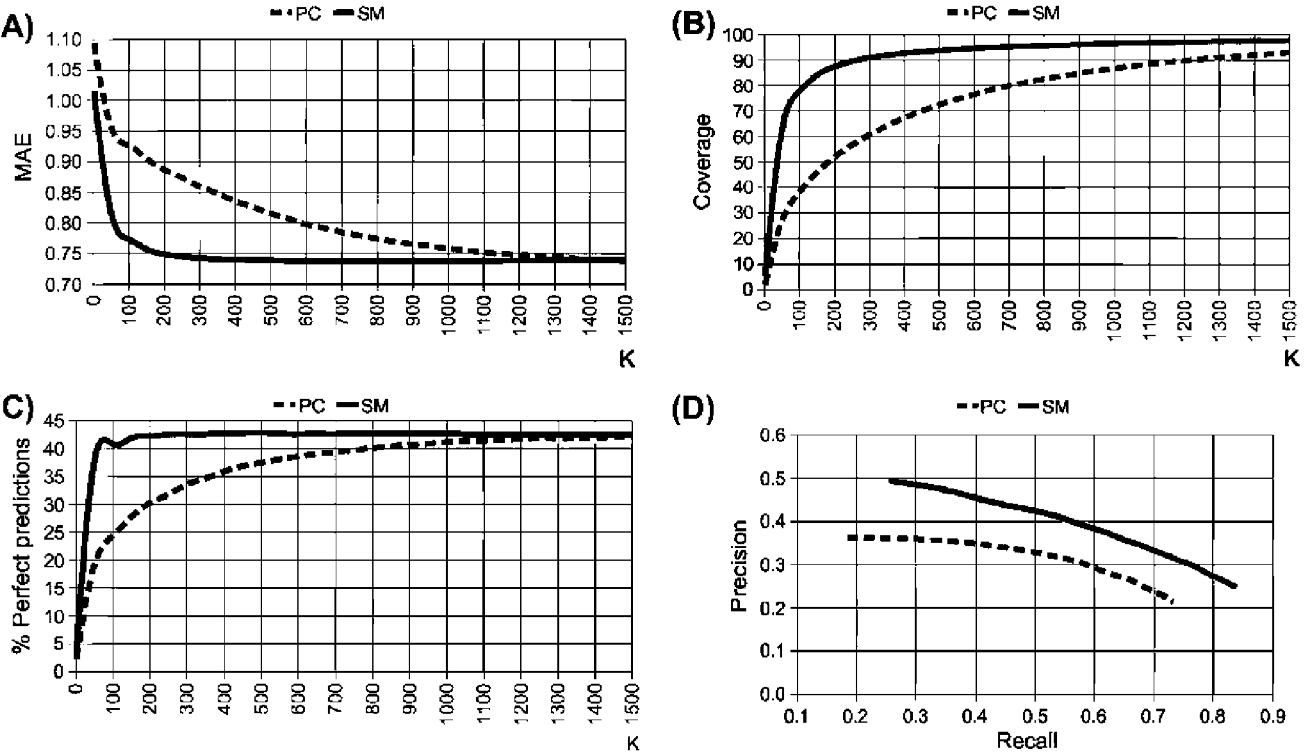

Fig. 5. Movielens 1 M quality results comparing the proposed singularity metric (SM, continuous line) with the reference one (Pearson correlation, broken line). The basic formulation was used and the values of the parameters are specified in Table 6.

5.3. Results obtained using databases with limited voting ranges

Fig. 5 shows the results with MovieLens 1M using the basic formulation: the proposed SM significantly improves all of the quality measures in comparison to the results provided by Pearson Correlation. For values of K = 200–300, where the SM trend stabilises, the improvements fluctuate between approximate values of 15% for MAE and 60% for coverage. The recommendation quality improves, on average, by 25% for recall and by 30% for precision.

Fig. 6 presents the results for Netflix using the basic formulation: in this case also, the proposed SM significantly improves all of the quality measures in comparison to the results provided by Pearson Correlation. For values of K = 1000–2000, where the SM trend stabilises, the improvements range between approximate values of 11% for MAE and 60% for coverage. The recommendation quality improves, on average, by 20% for recall and by 60% for precision.

5.4. Results obtained using a database with a wide voting range

Fig. 7 shows the results with FilmAffinity using the basic formulation: in spite of the fact that this formulation is not adapted to the voting range used by the RS, the results are reasonably similar to those obtained using Pearson Correlation, and even the coverage improves spectacularly (for example, by 250% when K = 100, by 100% when K = 200, etc.). This situation provides very good prospects for the successful use of the extended formulation model.

Fig. 8 shows the results for FilmAffinity using the extended formulation. As it can be seen, the obtained values and the improvements percentages are very similar to the Netflix and the MovieLens ones, showing that the extended formulation works fine using real databases and extending properly the basic equations.

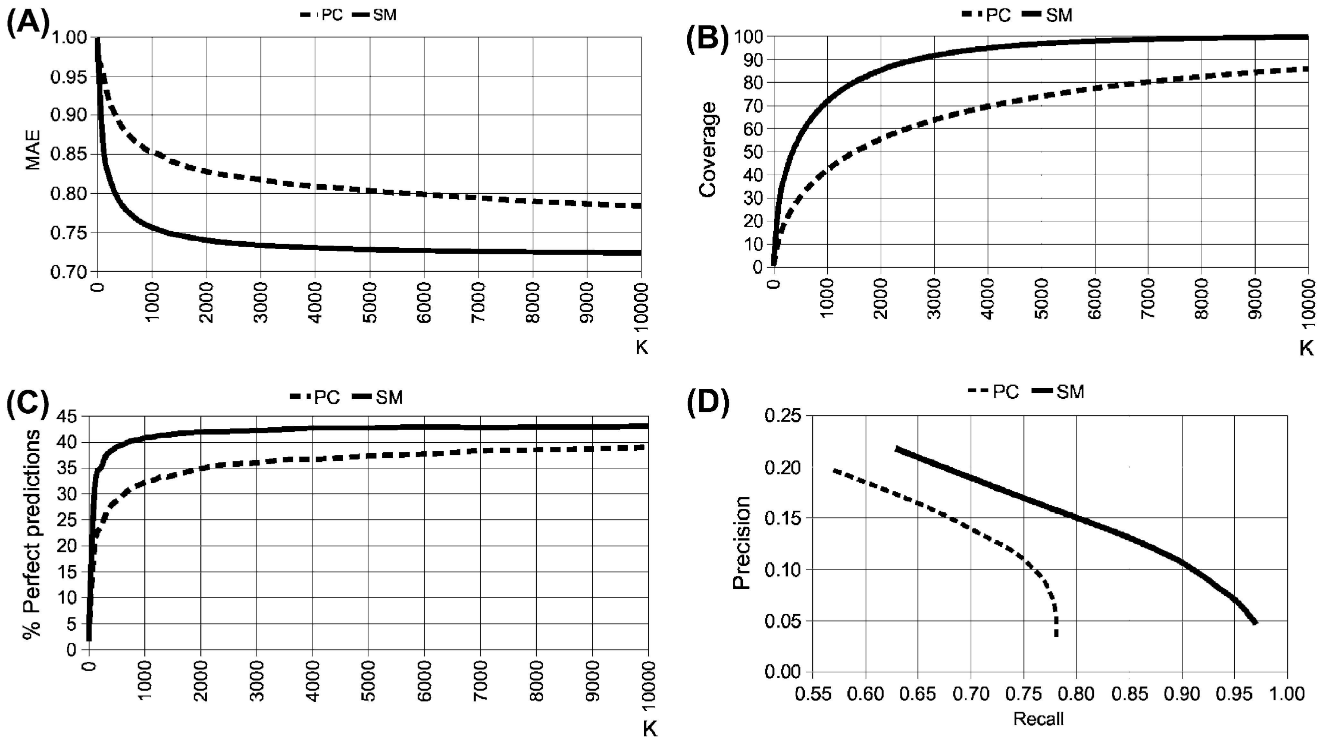

Fig. 6. Netflix quality comparing the proposed singularity metric (SM, continuous line) with the reference (Pearson correlation, broken line). The basic formulation was used and the values of the parameters are specified in Table 6.

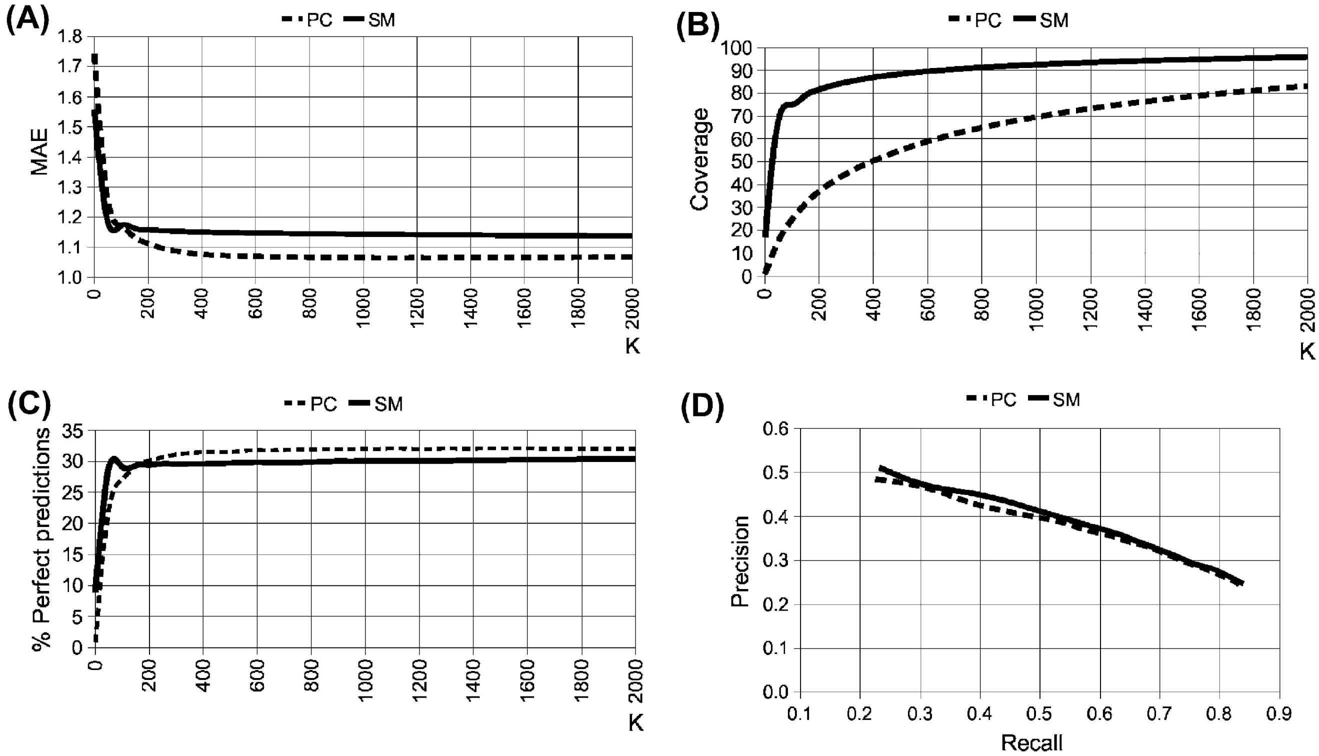

Fig. 7. FilmAffinity quality results comparing the proposed singularity metric (SM, continuous line) with the reference (Pearson correlation, broken line). The basic formulation was used and the values of the parameters are specified in Table 6.

6. Discussion

All of the results obtained using the SM proposed improve vastly in comparison to those obtained using the best available traditional metrics. In terms of prediction quality, in addition to reductions in the MAE margin, the improvements are

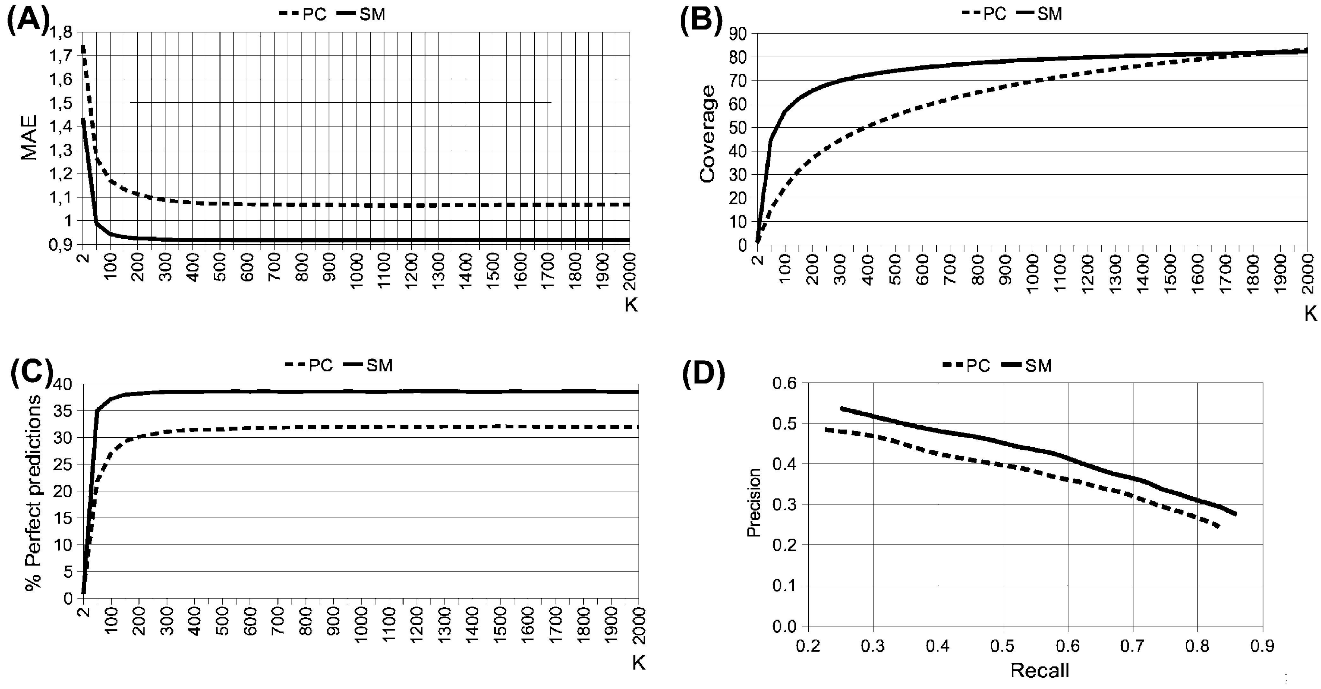

Fig. 8. FilmAffinity quality results comparing the proposed singularity metric (SM, continuous line) with the reference (Pearson correlation, broken line). The extended formulation was used and the values of the parameters are specified in Table 6.

especially noticeable in the percentage of perfect predictions achieved, which translates into large increases in recommendation quality measure precision versus recall, which is a measure which displays much smaller improvement variations when the various traditional metrics in CF are compared.

The coverage also displays drastic improvements when the proposed SM is used, and this contributes to an improvement in the measures of recommendation quality, as it expands the group of items liable to be chosen as recommended-relevants.

From a practical perspective, where the improvements are is important; as in this case the improvements from using the SM proposed are achieved from the very beginning of the graphs, which facilitates the choice of low values for *K*, thus maintaining an adequate quality of results and performance in the RS.

Although the basic formulation model achieves very good results in RS with small ranges of values for votes, in order to tackle CF processes in RS with ranges greater than five possible values of votes, it becomes necessary to use the extended formulation model.

7. Conclusions and future work

Recommender systems contain information which is not made use of by traditional metrics, the use of which is simple and yields excellent results for prediction and recommendation. This information is based on the analysis of the singularity of user votes in comparison to votes cast by the rest of the users; as such, it makes use of contextual information which is ignored in current RS.

The singularity information obtained can be combined with the values of the votes by the users to be compared, thus giving rise to the achievement of singularity measures such as the one proposed in this paper.

Using the singularity measure suggested, a vast improvement is achieved comparing to the results obtained using traditional similarity measures, both in terms of prediction quality and recommendation quality.

Recommender systems which have voting ranges with more than five options (for example, in the order of 1, ... , 10) require the use of the extended formulation model provided, whereas any other recommender systems can be processed using the basic formulation model suggested.

As future works, this paper leaves open the possibility of discovering a method for determining the most appropriate number and composition of the set of groups into which to divide possible votes in RS. This method would serve to complement and simplify the use of the extended formulation model set out in Section 4.3. Another possible extension would consist of studying the impact resulting from incorporating the pairs of votes ("non-positive, "not voted") and ("not voted", "not voted") into the model, taking as the hypothesis that, in many cases, an item which has not been voted for implies that the user has preferred not to give, in an explicit way, a negative evaluation.

Acknowledgements

Our acknowledgement to the *Grouplens* Research Group, to the *Netflix* and *FilmAffinity* companies, and to the *Knowledge Based Systems* journal.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734–749.

Al-Shamri, M. Y., & Bharadwaj, K. K. (2008). Fuzzy-genetic approach to recommender systems based on a novel hybrid user model. *Expert Systems with Applications, 35*, 1386–1399.

Antonopoulus, N., & Salter, J. (2006). Cinema screen recommender agent: Combining collaborative and content-based filtering. *IEEE Intelligent Systems*, 35–41.

Baltrunas, L. (2008). Exploiting contextual information in recommender systems. In: *ACM conference on recommender systems: RecSys* (pp.295–298).

Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge Based Systems, 23*(6), 520–528. doi:10.1016/j.knosys.2010.03.009.

Bobadilla, J., Serradilla, F., & Hernando, A. (2009). Collaborative filtering adapted to recommender systems of e-learning. *Knowledge Based Systems, 22*, 261–265. doi:10.1016/j.knosys.2009.01.008.

Candillier, L., Meyer, F., Boullé, M. (2007). Comparing state-of-the-art collaborative filtering systems. In *LNAI 4571* (pp. 548–562).

Chen, G., Wang, F., & Zhang, Ch. (2009). Collaborative filtering using orthogonal nonnegative matrix tri-factorization. *Information Processing Management, 45*(3), 368–379.

Denis, H. (2007). Managing collaborative learning processes, e-learning applications. In *29th int. conf. on inf. technol. interfaces* (pp. 345–350).

Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing, 5*(1), 4–7.

Fuyuki, I., Quan, T. K., Shinichi, H. (2006). Improving accuracy of recommender systems by clustering items based on stability of user similarity. In *IEEE int. conf. on intell. agents, web tech. and internet commer.* (p. 61). doi:10.1109/CIMCA.2006.123.

Gao, L. Q., Li, C. (2008). Hybrid personalizad recommended model based on genetic algorithm. In *Int. conf. on wireless commun. netw. and mob. computing* (pp. 9215–9218).

Giaglis, G. M., & Lekakos (2006). Improving the prediction accuracy of recommendation algorithms: Approaches anchored on human factors. *Interacting with Computers, 18*(3), 410–431.

Herlocker, J. L., Konstan, J. A., Borchers, Al., Riedl, J. T. (1999). An algorithmic framework for performing collaborative filtering. In *SIGIR* (pp. 230–237).

Herlocker, J. L., Konstan, J. A., & Riedl, J. T. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval, 5*, 287–310.

Herlocker, J. L., Konstan, J. A., Riedl, J. T., & Terveen, L. G. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*(1), 5–53.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS), 22*(1), 89–115. doi:10.1145/963770.963774.

Jinghua, H., Kangning, W., & Shaohong, F. (2007). A survey of e-commerce recommender systems. *International Conference on Service System and Service Management*, 11–125. doi:10.1109/ICSSSM.2007.4280214.

Knights, M. (2007). Web 2.0. *IET Communications Engineer*, 30–35.

Konstan, J. A., Miller, B. N., & Riedl, J. (2004). PocketLens: Toward a personal recommender system. *ACM Transactions on Information Systems, 22*(3), 437–476.

Krulwich, B. (1997). Lifestyle finder: Intelligent user profiling using large-scale demographic data. *Artificial Intelligence Magazine, 18*(2), 37–45.

Lang, K. (1995). NewsWeeder: Learning to filter netnews. In *12th int. conf. on machine learning*, Tahoe City, CA.

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

Li, Q., Myaeng, S. H., & Kim, B. M. (2007). A probabilistic music recommender considering user opinions and audio features. *Information Processing and Management, 43*(2), 473–487. doi:10.1016/j.ipm.2006.07.005.

Li, P., & Yamada, S. (2004). A movie recommender system based on inductive learning. *IEEE Conference on Cybernetics and Intelligent Systems, 1*, 318–323. doi:10.1109/ICCIS.2004.1460433.

Lin, K. J. (2007). Building Web 2.0. *Computer*, 101–102.

Manolopoulus, Y., Nanopoulus, A., Papadopoulus, A. N., & Symeonidis, P. (2007). Collaborative recommender systems: Combining effectiveness and efficiency. *Expert Systems with Applications* (34), 2995–3013. doi:10.1016/j.eswa.2007.05.013.

Nayak, R., Weng, L. T., & Xu, Y. (2005). An improvement to collaborative filtering for recommender systems. *IEEE International Conference on Computational Intelligence for Modeling, Control and Automatitation, 1*, 792–795. doi:10.1109/CIMCA.2005.1631361.

Park, Han-Saem, Yoo, Ji-Oh, & Cho, Sung-Bae (2006). A context-aware music recommendation system using fuzzy bayesian networks with utility theory. *Fuzzy Systems and Knowledge Discovery, 4223*, 970–979. doi:10.1007/11881599_121.

Pereira, F. C. N., Tishby, N., Lee, L. (1993). Distributional clustering of English words. *Meeting of the Association for Computational Linguistics* (pp. 183–190).

Qing, L., Sung, H., Myaeng, B., & Man, K. (2007). A probabilistic music recommender considering user opinions and audio features. *Information Processing Management, 43*(2), 473–487.

Ryan, P. B., & Bridge, D. (2006). Collaborative recommending using formal concept analysis. *Knowledge Based Systems, 19*(5), 309–315.

Sanchez, J. L., Serradilla, F., Martinez, E., Bobadilla, J. (2008). Choice of metrics used in collaborative filtering and their impact on recommender systems. In *IEEE DEST* (pp. 432–436). doi:10.1109/DEST.2008.4635147.