A framework for collaborative filtering recommender systems

Jesus Bobadilla *, Antonio Hernando, Fernando Ortega, Jesus Bernal

ABSTRACT

As the use of recommender systems becomes more consolidated on the Net, an increasing need arises to develop some kind of evaluation framework for collaborative filtering measures and methods which is capable of not only testing the prediction and recommendation results, but also of other purposes which until now were considered secondary, such as novelty in the recommendations and the users' trust in these. This paper provides: (a) measures to evaluate the novelty of the users' recommendations and trust in their neighborhoods, (b) equations that formalize and unify the collaborative filtering process and its evaluation, (c) a framework based on the above-mentioned elements that enables the evaluation of the quality results of any collaborative filtering applied to the desired recommender systems, using four graphs: quality of the predictions, the recommendations, the novelty and the trust.

1. Introduction

Recommender systems (RS) are developed to attempt to reduce part of the information overload problem produced on the Net. As opposed to other traditional help systems, such as search engines (Google, Yahoo, etc.), RS generally base their operation on a Collaborative Filtering (CF) process, which provides personalized recommendations to active users of websites where different elements (products, films, holidays, etc.) can be rated.

RS are inspired by human social behavior, where it is common to take into account the tastes, opinions and experiences of our acquaintances when making all kinds of decisions (choosing films to watch, selecting schools for our children, choosing products to buy, etc.). Obviously, our decisions are modulated according to our interpretation of the similarity that exists between us and our group of acquaintances, in such a way that we rate the opinions and experiences of some more highly than others.

By emulating each step of our own behavior insofar as is possible, the CF process of RS firstly selects the group of users from the RS website that is most similar to us, and then provides us with a group of recommendations of elements that we have not rated yet (assuming in this way that they are new to us) and which have been rated the best by the group of users with similar tastes to us. This way, a trip to the Canary Islands could be recommended to an individual who has rated different destinations in the Caribbean very highly, based on the positive ratings about the holiday destination of "Canary Islands" of an important number of individuals who also rated destinations in the Caribbean very highly. This suggestion (recommendation) will often provide the user of the service with inspiring information from the collective knowledge of all other users of the service.

RS cover a wide variety of applications (Baraglia & Silvestri, 2004; Bobadilla, Serradilla, & Hernando, 2009; Fesenmaier et al., 2002; Jinghua, Kangning, & Shaohong, 2007; Serrano, Viedma, Olivas, Cerezo, & Romero, 2011), although those related to movie recommendations are by far the best and most widely-used in the research field (Antonopoulus & Salter, 2006; Konstan, Miller, & Riedl, 2004).

A substantial part of the research in the area of CF focuses on how to determine which users are similar to the given one; in order to tackle this task, there are fundamentally three approaches: memory-based methods, model-based methods and hybrid approaches.

Memory-based methods (Bobadilla, Ortega, & Hernando, in press; Bobadilla, Serradilla, & Bernal, 2010; Kong, Sun, & Ye, 2005; Sanchez, Serradilla, Martinez, & Bobadilla, 2008; Symeonidis, Nanopoulos, & Manolopoulos, 2008) use similarity metrics and act directly on the ratio matrix that contains the ratings of all users who have expressed their preferences on the collaborative service; these metrics mathematically express a distance between two users based on each of their ratios. Model-based methods (Adomavicius & Tuzhilin, 2005) use the ratio matrix to create a model from which the sets of similar users will be established. Among the most widely used models we have: Bayesian classifiers (Cho, Hong, & Park, 2007), neural networks (Ingoo, Kyong, & Tae, 2003) and fuzzy systems (Yager, 2003). Generally, commercial RS use memory-based methods (Giaglis & Lekakos, 2006), whilst model-based methods are usually associated with research RS.

Regardless of the method used in the CF stage, the technical aim generally pursued is to minimize the prediction errors, by making the accuracy (Fuyuki, Quan, & Shinichi, 2006; Giaglis & Lekakos, 2006; Li & Yamada, 2004; Manolopoulus, Nanopoulus, Papadopoulus, & Symeonidis, 2007; Su & Khoshgoftaar, 2009) of the RS as high as possible; nevertheless, there are other purposes that need to be taken into account: avoid overspecialization phenomena, find good items, trust of recommendations, novelty, precision and recall measures, sparsity, cold start issues, etc.

The framework proposed in the paper gives special importance to the quality of the predictions and the recommendations, as well as to the novelty and trust results. Whilst the importance of the quality obtained in the predictions and recommendations has been studied in detail since the start of the RS, the quality results in novelty and trust provided by the different methods and metrics used in CF have not been evaluated in depth.

Measuring the quality of the trust results in recommendations becomes even more complicated as we are entering a particularly subjective field, where each specific user can grant more or less importance to various aspects that are selected as relevant to gain their trust in the recommendations offered (recommendation of recent elements, such as film premieres, introduction of novel elements, etc.). Another additional problem is the number of nuances that can be taken into account together with the lack of consensus to define them; in this way we can find studies on trust, reputation, credibility, importance, expertise, competence, reliability, etc. which sometimes pursue the same objective and other times do not.

In Buhwan, Jaewook, and Hyunbo (2009) we can see some novel memory-based methods that incorporate the level of a user credit instead of using similarity between users. In Kwiseok, Jinhyung, and Yongtae (2009) they employ a multidimensional credibility model, source credibility from consumer psychology, and provide a credible neighbor selection method, although the equations involved require a great number of parameters of difficult or arbitrary adjustment. O'Donovan and Smyth (2005) presents two computational models of trust and show how they can be readily incorporated into CF frameworks. Kitisin and Neuman (2006) propose an approach to include the social factors e.g. user's past behaviors and reputation together as an element of trust that can be incorporated into the RS. Zhang (2008) and Hijikata et al., 2009 tackle the novelty issue: in the first paper they propose a novel topic diversity metric which explores hierarchical domain knowledge, whilst in the second paper they infer items that a user does not know by calculating the similarity of users or items based on information about what items users already know. An aspect related to the trust measures is the capacity to provide justifications for the recommendations made; in Symeonidis et al. (2008) they propose an approach that attains both accurate and justifiable recommendations, constructing a feature profile for the users to reveal their favorite features.

To date, various publications have been written which tackle the way the RS are evaluated, among the most significant we have Herlocker, Konstan, Riedl, and Terveen (2004) which reviews the key decisions in evaluating CF RS: the user tasks, the type of analysis and datasets being used, the ways in which prediction quality is measured and the user-based evaluation of the system as a whole. Hernández and Gaudioso (2008) is a current study which proposes a recommendation filtering process based on the distinction between interactive and non-interactive subsystems. General publications and reviews also exist which include the most commonly accepted metrics, aggregation approaches and evaluation measures: mean absolute error, coverage, precision, recall and derivatives of these: mean squared error, normalized mean absolute error, ROC and fallout; Goldberg, Roeder, Gupta, and Perkins (2001) focus on the aspects not related to the evaluation, Breese, Heckerman, and Kadie (1998) compare the predictive accuracy of various methods in a set of representative problem domains. Candillier, Meyer, and Boullé (2007) and Schafer, Frankowski, Herlocker, and Sen, 2007 review the main CF methods proposed in the literature.

Among the most significant papers that propose a CF framework is Herlocker, Konstan, Borchers, and Riedl (1999) which evaluates the following: similarity weight, significance weighting, variance weighting, selecting neighborhood and rating normalization; Hernández and Gaudioso (2008) propose a framework in which any RS is formed by two different subsystems, one of them to guide the user and the other to provide useful/interesting items. Koutrika, Bercovitz, and Garcia (2009) is a recent and very interesting framework which introduces levels of abstraction in CF process, making the modifications in the RS more flexible.

The RS frameworks proposed until now present two deficiencies which we aim to tackle in this paper. The first of these is the lack of formalization in the evaluation methods; although the quality metrics are well defined, there are a variety of details in the implementation of the methods which, in the event they are not specified, can lead to the generation of different results in similar experiments. The second deficiency is the absence of quality measures of the results in aspects such as novelty and trust of the recommendations.

The following section of this paper develops a complete series of mathematical formalizations based on sets theory, backed by a running example which aids understanding and by cases of studies which show clarifying results of the aspects and alternatives shown; in this section, we also obtain the combination of metric, aggregation approach and standardization method which provides the best results, enabling it to be used as a reference to evaluate metrics designed by the scientific community. In Section 3 we specify the evaluation measures proposed in the framework, which include the quality analysis of the following aspects: predictions (estimations), recommendations, novelty and trust; this same section shows the results obtained by using MovieLens 1M and NetFlix. Finally, we set our most relevant conclusions.

2. Framework specifications

This section provides both the equations on which the prediction/recommendation process in the CF stage is based and the equations that support the quality evaluation process offered in the proposed framework; between these last two we have the traditional MAE, coverage, precision, recall and those developed specifically to complete the framework: novelty-precision, novelty-recall, trust-precision, trust-recall.

The objective of formalizing the prediction, recommendation and evaluation processes is to ensure that the experiments carried out by different researchers can be reproduced and are not altered by different decisions made on behalf of different implementation details: e.g. deciding how to act when no *k*-neighborhoods have voted for a specific item (we could say not predict, or predict with the average votes of all users on that item), whether we apply a standardization process to the input data or to the weightings of the aggregation approach, whether on finding an error in a prediction we take the decimal values of the prediction or round them off to the nearest whole value, etc.

The formalization presented here is fundamental when specifying a framework, where the same experiments carried out by different researchers must give the same results, in order to be able to compare the metrics and methods developed over time at different research centers.

Throughout the section, a running example is provided to help to understand and follow the underlying ideas in each group of directly interrelated equations. In the same way, various real results are provided (obtained with MovieLens) grouped into "case of study" subsections where the integrities and defects of each of the alternatives mentioned can be compared.

The main subsections in which this section is structured are: preliminary definitions, similarity measures, obtaining a user's k-neighborhoods, prediction of the value of an item, obtaining the accuracy, standardization process, obtaining the coverage, top N recommendations, quality of the recommendation: precision and recall, quality of the novelty: novelty-precision and novelty-recall and quality of the trust: trust-precision and trust-recall.

2.1. Preliminary definitions

In this subsection we specify the definitions, parameters, measures and sets used in the equations, as well as the values of the running example. In order to simplify the equations of the other subsections, we do not specify here the different learning and test groups which must be used in the framework operation.

2.1.1. Formalization

Given an RS with a database of L users and M items rated in the range $\{min,\ldots,max\}$, where the absence of ratings will be represented by the symbol $\bullet$.

We define the basic sets: (N: natural numbers, R: real numbers)

$$U = \{u \in N | 1 \leqslant u \leqslant L\}, \quad \text{set of users} \tag{1}$$

$$I = \{i \in N | 1 \leqslant i \leqslant M\}, \quad \text{set of items} \tag{2}$$

$$V = \{v \in N | \min \leqslant v \leqslant \max\} \cup \{\bullet\}, \quad \text{set of possible votes} \tag{3}$$

$$R_u = \{(i, v) | i \in I, v \in V\}, \quad \text{ratings of user } u \tag{4}$$

$$\text{We define vote } v \text{ of user } u \text{ on item } i \text{ as } r_{u,i} = v \tag{5}$$

$$\text{We define the average of the valid votes of user } u \text{ as } \bar{r}_u \tag{6}$$

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

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

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

Below we present the tables of parameters (Table 1), measures (Table 2) and sets (Table 3) used in the formalizations made in the paper.

2.1.2. Running example

$$U = \{u \in N | 1 \leqslant u \leqslant 5\}, \quad I = \{i \in N | 1 \leqslant i \leqslant 14\},$$
$$V = \{v \in N | 1 \leqslant v \leqslant 5 \vee v = \bullet\}$$

$$R_1 = \left\{ \begin{array}{c} \langle 1,5\rangle, \langle 2,\bullet\rangle, \langle 3,\bullet\rangle, \langle 4,3\rangle, \langle 5,\bullet\rangle, \langle 6,4\rangle, \langle 7,1\rangle, \\ \langle 8,\bullet\rangle, \langle 9,\bullet\rangle, \langle 10,4\rangle, \langle 11,\bullet\rangle, \langle 12,2\rangle, \langle 13,4\rangle, \langle 14,\bullet\rangle \end{array} \right\}$$

Table 1
Parameters.

Name	Parameters descriptions
L	#Users
M	#Items
min	#min rating value
max	#max rating value
K	#Neighborhoods
N	#Recommendations
β	#Range to the current time
γ	#Ratings received
θ	Relevant item threshold
q	#Trust users
h	#Trust items

Table 2
Measures.

Name	Measures descriptions	Usage
r_u, i	Rating of the user on the item	General use
t_u, i	Rating time of the user on the item	
p_u, i	Prediction to the user on the item	
m_u	Mean absolute error on the user	Prediction quality
m	Mean absolute error of the RS	
c_u	Coverage on the user	
c	Coverage of the RS	
t_u	Recommendation precision on the user	Recommendation quality
t	Recommendation precision of the RS	
x_u	Recommendation recall on the user	
x	Recommendation recall of the RS	
n_u	Novelty precision on the user	Novelty quality
n	Novelty precision of the RS	
l_u	Novelty recall on the user	
l	Novelty recall of the RS	
w_u	Trust precision on the user	Trust quality
w	Trust precision of the RS	
z_u	Trust recall on the user	
z	Trust recall of the RS	

$$R_2 = \left\{ \begin{array}{c} \langle 1,1\rangle, \langle 2,\bullet\rangle, \langle 3,\bullet\rangle, \langle 4,2\rangle, \langle 5,4\rangle, \langle 6,1\rangle, \langle 7,\bullet\rangle \\ \langle 8,\bullet\rangle, \langle 9,\bullet\rangle, \langle 10,\bullet\rangle, \langle 11,\bullet\rangle, \langle 12,\bullet\rangle, \langle 13,4\rangle, \langle 14,1\rangle \end{array} \right\}$$

$$R_3 = \left\{ \begin{array}{c} \langle 1,5\rangle, \langle 2,2\rangle, \langle 3,\bullet\rangle, \langle 4,4\rangle, \langle 5,\bullet\rangle, \langle 6,\bullet\rangle, \langle 7,\bullet\rangle, \\ \langle 8,3\rangle, \langle 9,5\rangle, \langle 10,4\rangle, \langle 11,\bullet\rangle, \langle 12,\bullet\rangle, \langle 13,4\rangle, \langle 14,\bullet\rangle \end{array} \right\},$$

$$R_4 = \left\{ \begin{array}{c} \langle 1,4\rangle, \langle 2,\bullet\rangle, \langle 3,\bullet\rangle, \langle 4,3\rangle, \langle 5,\bullet\rangle, \langle 6,\bullet\rangle, \langle 7,\bullet\rangle, \\ \langle 8,\bullet\rangle, \langle 9,5\rangle, \langle 10,4\rangle\langle 11,\bullet\rangle, \langle 12,\bullet\rangle, \langle 13,\bullet\rangle, \langle 14,\bullet\rangle \end{array} \right\}$$

$$R_5 = \left\{ \begin{array}{l} \langle 1,\bullet\rangle, \langle 2,\bullet\rangle, \langle 3,\bullet\rangle, \langle 4,\bullet\rangle, \langle 5,\bullet\rangle, \langle 6,\bullet\rangle, \langle 7,3\rangle, \\ \langle 8,3\rangle, \langle 9,4\rangle, \langle 10,5\rangle, \langle 11,\bullet\rangle, \langle 12,\bullet\rangle, \langle 13,5\rangle, \langle 14,\bullet\rangle \end{array} \right\}$$

2.2. Similarity measures

2.2.1. Introduction

The proposed framework will allow to compare future similarity measures and methods, in the meantime, it is advisable to substantiate the behavior of well-known similarity measures and propose the one that gives the best results, so that it can act as a reference for future comparisons. The user to user similarity measures most commonly-used in RS are: Pearson Correlation, cosine, Constrained Pearson's Correlation and Spearman rank correlation.

The similarity approaches usually compute the similarity between two users x and y: $sim(x,y)$ based on their ratings of items that both users have rated (9).

$$A_{x,y} = \{i \in I | r_{x,i} \neq \bullet \wedge r_{y,i} \neq \bullet\}. \tag{9}$$

2.2.2. Running example

In order to make the example easier to follow we will use a similarity measure that is very easy to calculate manually: the Mean Square Difference (MSD) of two users x and y.

$$MSD(x,y) = \frac{1}{\#A_{x,y}} \sum_{i \in A_{x,y}} (r_{x,i} - r_{y,i})^2.$$

We represent the votes issued in table format (Table 4):

We obtain the table of similarities between users (Table 5), taking into account that $MSD(x,y) = MSD(y,x)$. The maximum similarity is reached at value 0.

Calculation example: $MSD(U_1, U_2) = \frac{1}{4}[(5-1)^2 + (3-2)^2 + (4-1)^2 + (4-4)^2] = 6.5$.

Table 3
Sets.

Name	Sets descriptions	Parameters
U	Users	L
I	Items	M
V	Rating values	*min, max*
R_u	User ratings	*user*
K_u	Neighborhoods of the user	*user, k*
P_u	Predictions to the user	*user, k*
X_u	Top recommended items to the user	*user, k, θ*
Z_u	Top N recommended items to the user	*user, k, N, θ*
Y	Items voted of the most by γ users	γ
T_u	User's neighborhoods taking into account β	*user, k, β, q*
Q_u	Trust users	*user, k, β, h*
H_u	Trust pairs (user, item)	*user, k, β*
$A_{x,y}$	Items rated simultaneously by users x and y	*user*1, *user*2
$G_{u,i}$	User's neighborhoods which have rated item i	*user, k*
$B_{u,i}$	Users who have voted for item i, except*user*	*user, item*
O_u	Items that the user has voted for and on which predictions exist	*user, k*
O	Users from whom a MAE can be obtained	k
C_u	Items that the user has not voted for and on which predictions exist	*user, k*
D_u	Items that the user has not voted for	*user, k*
C	Pairs (user, item) that have not been voted for and accept predictions	k
D	Pairs (user, item) that have not been voted for	
$E_{x,y}$	Items that have recently been voted for by both user x and user y	*β, user*1, *user*2
S_u	User's recent votes	*user, β*

Table 4
Running example: RS database.

$r_{u,i}$	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}
U_1	5	•	•	3	•	4	1	•	•	4	•	2	4	•
U_2	1	•	•	2	4	1	•	•	•	•	•	•	4	1
U_3	5	2	•	4	•	•	•	3	5	4	•	•	4	•
U_4	4	•	•	3	•	•	•	•	5	4	•	•	•	•
U_5	•	•	•	•	•	•	3	3	4	5	•	•	5	•

2.3. Obtaining a user's K-neighbors

2.3.1. Formalization

We define K_u as the set of K neighbors of the user u.

The following must be true:

$$K_u \subset U \wedge \#K_u = k \wedge u \notin K_u, \tag{10}$$

$$\forall x \in K_u, \quad \forall y \in (U - K_u), \quad sim(u,x) \geqslant sim(u,y). \tag{11}$$

2.3.2. Running example

Table 6 shows the sets of neighbors using $K = 2$ and $K = 3$:

2.4. Prediction of the value of an item

2.4.1. Formalization

Based on the information provided by the K-neighbors of a user u, the CF process enables the value of an item to be predicted as follows:

$$\text{Let } P_u = \{(i,p) | i \in I, p \in R\},$$
$$\text{set of prediction to the user } u \ (R : \text{real numbers}) \tag{12}$$

$$\text{We will assign the value of the prediction } p$$
$$\text{made to user } u \quad \text{on item } i \text{ as } p_{u,i} = p \tag{13}$$

Table 5
Running example: users similarities.

MSD	U_1	U_2	U_3	U_4	U_5
U_1	0	6.5	0.25	0.33	2
U_2	6.5	0	6.66	5	1
U_3	0.25	6.66	0	0.5	0.75
U_4	0.33	5	0.5	0	1
U_5	2	1	0.75	1	0

Table 6
Running example: 2 and 3 neighbors of each user.

K_u	U_1	U_2	U_3	U_4	U_5
$K = 2$	$\{U_3, U_4\}$	$\{U_5, U_4\}$	$\{U_1, U_4\}$	$\{U_1, U_3\}$	$\{U_3, U_2\}$
$K = 3$	$\{U_3, U_4, U_5\}$	$\{U_5, U_4, U_1\}$	$\{U_1, U_4, U_2\}$	$\{U_1, U_3, U_5\}$	$\{U_3, U_2, U_4\}$

Once the set of K users (neighbors) similar to active u has been calculated (K_u), in order to obtain the prediction of item i on user u(12), one of the following aggregation approaches is often used: the average (15), the weighted sum (16) and the adjusted weighted aggregation (Deviation-From-Mean) (17).

$$\text{Let } G_{u,i} = \{n \in K_u | \exists r_{n,i} \neq \bullet\} \tag{14}$$

$$p_{u,i} = \frac{1}{\#G_{u,i}} \sum_{n \in G_{u,i}} r_{n,i} \Longleftrightarrow G_{u,i} \neq \emptyset \tag{15}$$

$$p_{u,i} = \mu_{u,i} \sum_{n \in G_{u,i}} sim(u,n) r_{n,i} \Longleftrightarrow G_{u,i} \neq \emptyset \tag{16}$$

$$p_{u,i} = \bar{r}_u + \mu_{u,i} \sum_{n \in G_{u,i}} sim(u,n)(r_{n,i} - \bar{r}_n) \Longleftrightarrow G_{u,i} \neq \emptyset \tag{17}$$

where μ serves as a normalizing factor, usually computed:

$$\mu_{u,i} = 1 \Big/ \sum_{n \in G_{u,i}} sim(u,n) \Longleftrightarrow G_{u,i} \neq \emptyset \tag{18}$$

When it is not possible to make the prediction of an item as none of the K-neighbors has voted for this item, we can decide to make use of the average ratings given to that item by all users of the RS who have voted for it; in this case, Eqs. (14)–(18) are complemented with Eqs. (19)–(23):

$$\text{where } B_{u,i} = \{n \in U | n \neq u, r_{n,i} \neq \bullet\} \tag{19}$$

$$p_{u,i} = \frac{1}{\#B_{u,i}} \sum_{n \in B_{u,i}} r_{n,i} \Longleftrightarrow G_{u,i} = \emptyset \wedge B_{u,i} \neq \emptyset \tag{20}$$

$$p_{u,i} = \mu_{u,i} \sum_{n \in B_{u,i}} sim(u,n) r_{n,i} \Longleftrightarrow G_{u,i} = \emptyset \wedge B_{u,i} \neq \emptyset \tag{21}$$

$$p_{u,i} = \bar{r}_u + \mu_{u,i} \sum_{n \in B_{u,i}} sim(u,n)(r_{n,i} - \bar{r}_n) \Longleftrightarrow G_{u,i} = \emptyset \wedge B_{u,i} \neq \emptyset \tag{22}$$

$$\mu_{u,i} = 1 / \sum_{n \in B_{u,i}} sim(u,n) \Longleftrightarrow G_{u,i} = \emptyset \wedge B_{u,i} \neq \emptyset \tag{23}$$

Finally, in RS cases exist in which it is impossible to make predictions on some items that any other user has voted for:

$$p_{u,i} = \bullet \Longleftrightarrow G_{u,i} = \emptyset \wedge B_{u,i} = \emptyset \tag{24}$$

2.4.2. Running example

By using the simplest prediction Eq. (15) we obtain the predictions that the users can receive using $K = 3$ neighbors. Table 7 shows these predictions.

2.5. Obtaining the mean absolute error-accuracy

2.5.1. Formalization

In order to measure the accuracy of the results of an RS, it is usual to use the calculation of some of the most common error metrics, amongst which the mean absolute error (MAE) and its related metrics, mean squared error, root mean squared error, and normalized mean absolute error stand out.

$$\text{Let } O_u = \{i \in I | p_{u,i} \neq \bullet \wedge r_{u,i} \neq \bullet\} \tag{25}$$

We define the MAE of a user u as:

$$m_u = \frac{1}{\#O_u} \sum_{i \in O_u} |p_{u,i} - r_{u,i}| \iff O_u \neq \varnothing \tag{26}$$

$$m_u = \bullet \iff O_u = \varnothing \tag{27}$$

The MAE of the RS can be obtained as the average of the user's MAE:

$$\text{Let } O = \{u \in U | m_u \neq \bullet\} \tag{28}$$

We define the system's MAE as:

$$m = \frac{1}{\#O} \sum_{u \in O} m_u \iff O \neq \varnothing \tag{29}$$

$$m = \bullet \iff O = \varnothing \tag{30}$$

The accuracy is defined as the inverse of the error ($1/m$), but more specifically it can be established as: $accuracy = 1 - \frac{m}{max - min}$, $accuracy \in [0, 1]$.

2.5.2. Running example

Table 8 shows the mean absolute errors of each user (m_u) and of the system (m) using $K = 3$.

2.5.3. Case of study

Often, the system's MAE is implemented in such a way that when there are no neighbors capable of making a prediction on an item, the average for that item of all the training users (except the active user) is used as the prediction. This behavior is reflected in Eqs. (19)–(23), as opposed to Eqs. (14)–(18) which are used when there is at least one neighbor capable of making a prediction on the item considered. Fig. 1 shows the result obtained using both approaches applied to Pearson Correlation and making use of the average aggregation approaches (15), (20). Database: MovieLens 1M.

In graphs 1a (computed using Eq. (15)) and 1c (computed using Eq. (20)), a horizontal line appears at 0.797 which indicates the value of the MAE obtained using K = all the training users. Fig. 1c shows values that tend towards this limit when low values of K are selected, due to the fact that the lower the value of K the fewer the neighbors available in order to rate the items that the active user has voted for and therefore, the greater probability of having to make use of the votes of all the training users of the RS in order to make a prediction; in this case, when the MAE increases, the prediction capacity (coverage) decreases drastically (graph 1b).

Table 8
Mean absolute errors of each user (mu) and of the system (m) using K = 3.

	m_u
U_1	(0.5 + 0.5 + 2 + 0.33 + 0.5)/5 = 0.76
U_2	(3.5 + 1 + 3 + 0.5)/4 = 2
U_3	(1.67 + 1.34 + 0 + 0 + 0)/5 = 0.6
U_4	(1 + 0.5 + 0.5 + 0.33)/4 = 0.58
U_5	(0 + 1 + 1 + 1)/ 4 = 0.75
m	(0.76 + 2 + 0.6 + 0.58 + 0.75)/5 = 0.938

Fig. 2 shows the MAE results obtained on MovieLens 1M using various similarity measures and two aggregations approaches commonly used in CF (Eqs. (16) and (17)). The calculations have been made in the range K = 2 to K = 1500, by averaging their results; as we can see, the lowest error values are obtained using Pearson Correlation (PC), particularly when Deviation From Mean (DFM) is used as the aggregation approach. These results lead us to use PC-DFM as the reference combination which acts as a way of testing future metrics proposed by the scientific community, although it still needs to be tested with standardization methods, analysis of its coverage, quality of recommendations, etc.

When selecting a similarity measure we must take into account that the averaged results may lead to a false idea of the integrity of the real results, as can be seen in Fig. 3 where we can notice that, although PC-DFM presents a lower global MAE, when we use values of K-neighbors under 350 (which is quite common), CPC-WS offers better error measures. This situation must be considered in the accuracy analysis obtained in the RS.

2.6. Standardization process

2.6.1. Introduction

When using CF, at times it maybe a good idea to carry out a data standardization process. The z-scores, or normal scores distribute a group of data in line with a normal distribution.

$$z = \frac{x - \mu}{\sigma} \tag{31}$$

where x is a raw score to be standardized, μ is the mean of the population and σ is the standard deviation of the population. z is negative when the raw score is below the mean and positive when above.

Although the most obvious application is the standardization of the users' votes (of the input values), it is also possible to apply this process to improve the predictions: the similarity values $sim(u,n)$ obtained by applying the selected similarity measure are used to weight the importance of the votes of each K-neighbors (Eqs. (16)–(18)). In some cases, most of the neighbors show very high similarity values, and therefore, the weighting process loses effectiveness; in these cases it is effective to make use of z-scores to better differentiate the contribution that each neighbor will have in the prediction results.

2.6.2. Case of study

Fig. 4 shows the result of applying z-scores to the input data or to the similarity values $sim(u,n)$. Except in the case of cosine, which is greatly improved by applying z-scores to the input data, no

Table 7
Predictions that each user can receive using 3-neighbors.

$P_{u,i}$	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}
U_1	4.5	2	•	3.5	•	•	3	3	4.66	4.33	•	•	4.5	•
U_2	4.5	•	•	3	•	4	2	3	4.5	4.33	•	2	4.5	•
U_3	3.33	•	•	2.66	4	2.5	1	•	5	4	•	2	4	1
U_4	5	2	•	3.5	•	4	2	3	4.5	4.33	•	2	4.33	•
U_5	3.33	2	•	3	4	1	•	3	5	4	•	•	4	1

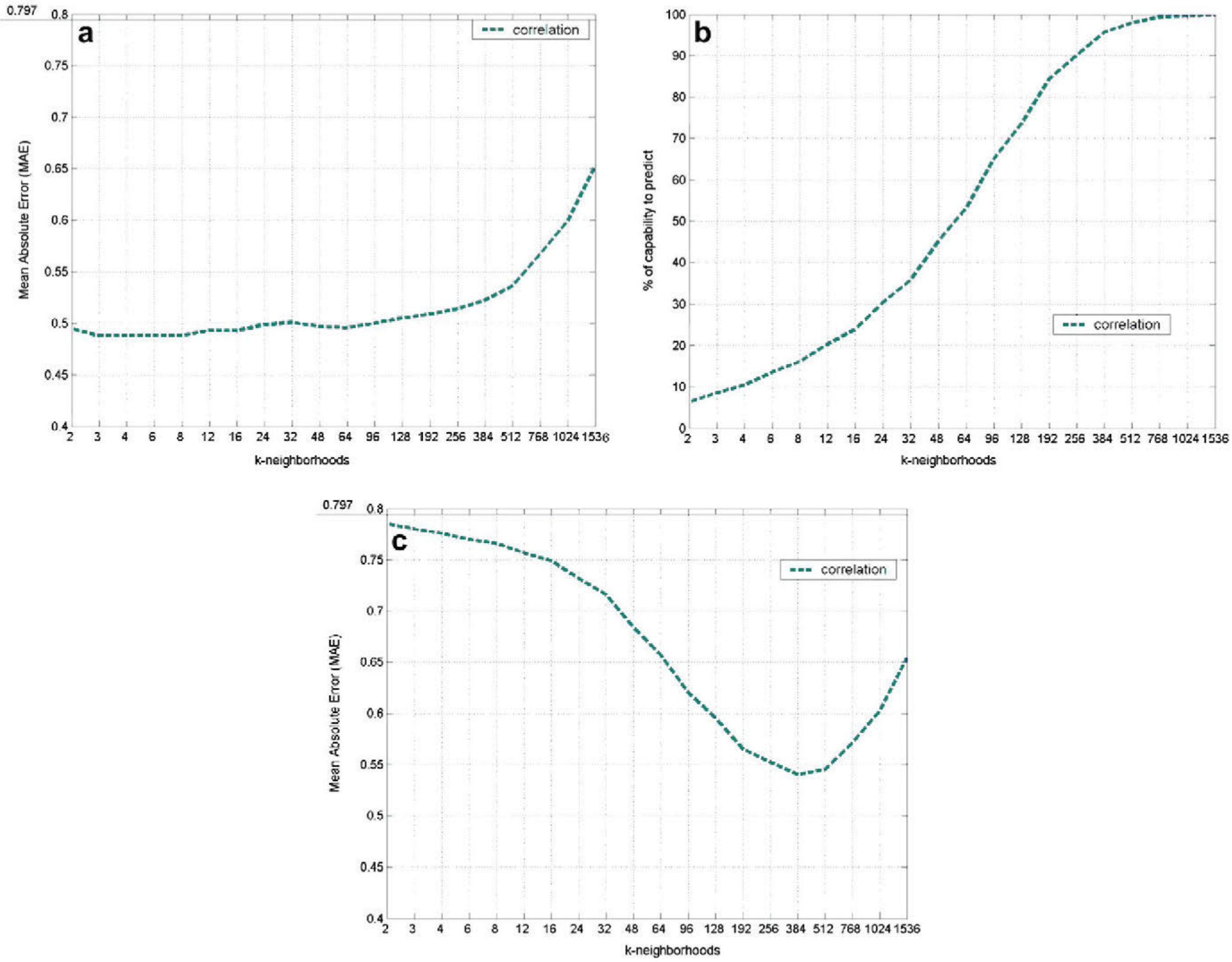

Fig. 1. (a) MAE obtained by only using the votes of the *K*-neighbors of each active user; (b) coverage; (c) MAE obtained using the votes of all the training users when the *K*-neighbors cannot make a prediction.

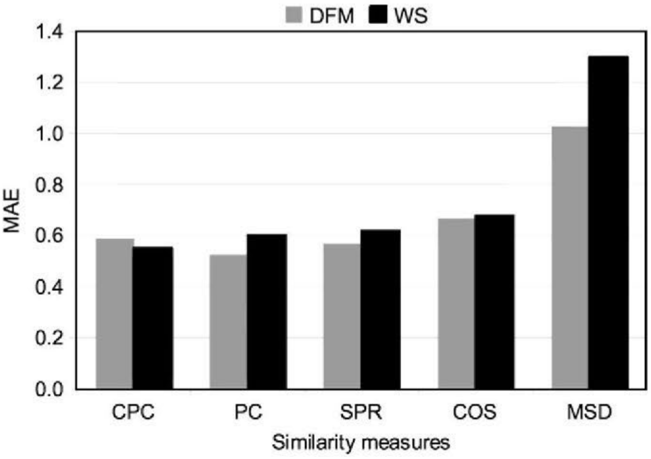

Fig. 2. MAE results obtained on MovieLens 1M using the similarity measures: Constrained Pearson Correlation (CPC), Pearson Correlation (PC), Spearman rank correlation (SPR), cosine (COS) and Mean Squared Differences (MSD), and making use of the aggregation approaches: Weighted Sum (WS) and Deviation From Mean (DFM).

significant improvements can be seen in the other metrics, however, by studying the details of the impact of the standardization processes for different *K*-neighbors (Fig. 5), we can see that by using Pearson Correlation DFM the effects of using *z*-scores on the similarity values (PC-DFM-Z) begin to produce improvements after a certain value of *K*: in the case of MovieLens 1M from *K* = 500 and with MovieLens 100 K from *K* = 100.

2.7. Obtaining the coverage

2.7.1. Formalization

The coverage could be defined as the capacity of predicting from a metric applied to a specific RS. In short, it calculates the percentage of situations in which at least one *K*-neighbor of each active user can rate an item that has not been rated yet by that active user.

$$\text{Let } C_u = \{i \in I | r_{u,i} = \bullet \wedge G_{u,i} \neq \varnothing\} \tag{32}$$

$$\text{Let } D_u = \{i \in I | r_{u,i} = \bullet\} \tag{33}$$

Coverage of user *u*:

$$c_u = 100 \times \frac{\#C_u}{\#D_u} \Longleftrightarrow D_u \neq \varnothing, \quad c_u = \bullet \Longleftrightarrow D_u = \varnothing \tag{34}$$

Coverage of the system:

$$\text{Let } C = \{(u,i) | u \in U, i \in I, r_{u,i} = \bullet, G_{u,i} \neq \varnothing\} \tag{35}$$

$$\text{Let } D = \{(u,i) | u \in U, i \in I, r_{u,i} = \bullet\} \tag{36}$$

$$c = 100x\frac{\#C}{\#D} \tag{37}$$

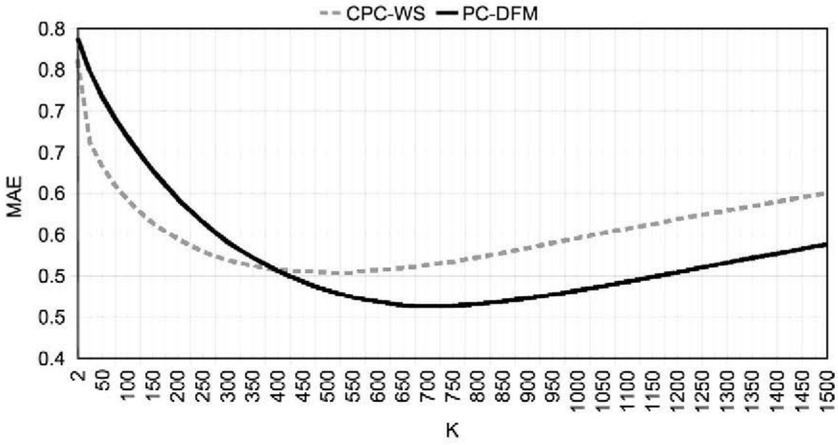

Fig. 3. Breakdown of the MAE obtained on MovieLens 1M using the similarity measures: Constrained Pearson Correlation (CPC) combined with Weighted Sum (WS) and Pearson Correlation (PC) combined with Deviation From Mean (DFM).

2.7.2. Running example

Table 9 shows the coverage measures using MSD and values $K = 2$ and $K = 3$.

2.7.3. Case of study

By comparing Fig. 6 with Fig. 4 we can see that there is a reverse trend between accuracy and coverage, to the extent that when choosing a metric we must not take only one of these measures as a reference. In Fig. 6 the similarity measure Mean Square Differences (MSD) shows much better results than the other metrics, however, as we have seen, it also has the worst accuracy. Along the same lines, Pearson Correlation using z-scores provides us with very low coverage values, in contrast to its good accuracy results.

Fig. 7 shows the breakdown of the coverage results using Pearson correlation with and without z-scores. As we can see, the use of this standardization process is justified in order to improve the accuracy, due to its minimum impact on the coverage.

2.8. Top N recommendations

2.8.1. Formalization

We define X_u as the set of recommendations to user u, and Z_u^* as the set of **N** recommendations to user u.

The following must be true:

$$X_u \subset I \wedge \forall i \in X_u, \quad r_{u,i} = \bullet, \quad p_{u,i} \neq \bullet, \tag{38}$$

$$Z_u \subseteq X_u, \quad \#Z_u = N, \quad \forall x \in Z_u, \quad \forall y \in X_u : p_{u,x} \geqslant p_{u,y} \tag{39}$$

If we want to impose a minimum recommendation value: $\theta \in R$, we add $p_{u,i} \geqslant \theta$

2.8.2. Running example

By making use of Eqs. (38) and (39), as an example, we obtain the recommendations that can be made to user U_3 with $\mathbf{N} = 2$ to $\mathbf{N} = 5$, using $K = 2$. Table 10 shows these values.

2.9. Quality of the recommendation: precision and recall

2.9.1. Formalization

First, we redefine the Eq. (38)

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

We will use t_u to represent the quality precision measure for recommendations obtained by making N test recommendations to the user u, taking a θ relevancy threshold. Similarly, x_u will represent the recall measure obtained by making the same N recommendations to user u.

Assuming that all users accept N test recommendations:

$$t_u = \frac{\#\{i \in Z_u | r_{u,i} \geqslant \theta\}}{N} \tag{40}$$

$$x_u = \frac{\#\{i \in Z_u | r_{u,i} \geqslant \theta\}}{\#\{i \in Z_u | r_{u,i} \geqslant \theta\} + \#\{i \in Z_u^c | r_{u,i} \neq \bullet \wedge r_{u,i} \geqslant \theta\}} \tag{41}$$

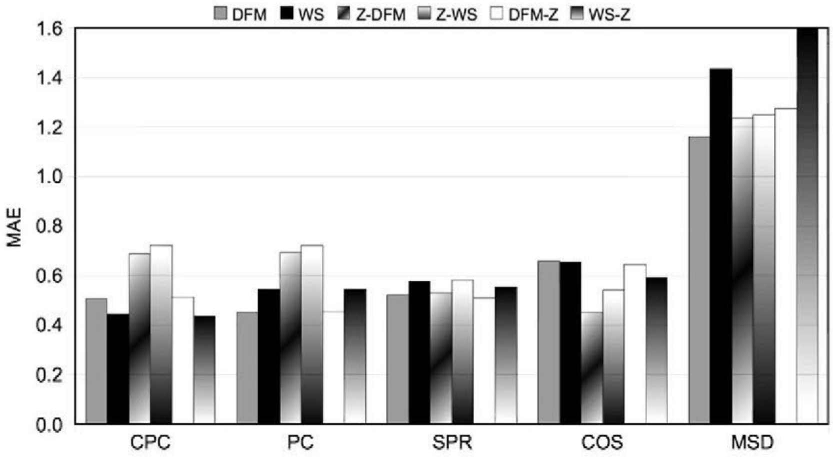

Fig. 4. MAE results obtained on MovieLens 1M using the similarity measures: Constrained Pearson Correlation (CPC), Pearson Correlation (PC), Spearman rank correlation (SPR), cosine (COS) and Mean Squared Differences (MSD), making use of the aggregation approaches: Weighted Sum (WS) and Deviation From Mean (DFM) and using z-scores in the input data ($Z-$) or in similarity values during the prediction process ($-Z$).

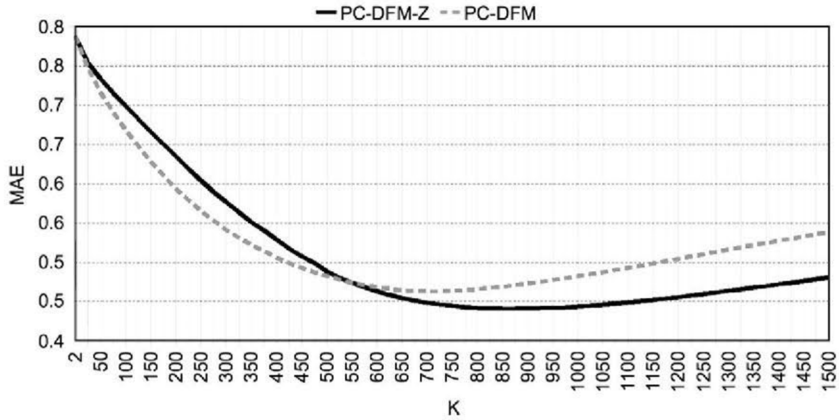

Fig. 5. Breakdown of the MAE obtained on MovieLens 1M using Deviation From Mean (DFM) Pearson Correlation (PC) using and not using *z*-scores in the similarity values during the prediction process (–*Z*).

Table 9
Coverage measures using MSD and values $K = 2$ and $K = 3$.

	$\#D_u$	$K = 2\#C_u$	$K = 3\#C_u$	$K = 2c_u$	$K = 3c_u$
U_1	7	$\#\{I_2, I_8, I_9\} = 3$	$\#\{I_2, I_8, I_9\} = 3$	42, 85%	42, 85%
U_2	8	$\#\{I_7, I_8, I_9, I_{10}\} = 4$	$\#\{I_7, I_8, I_9, I_{10}, I_{12}\} = 5$	50%	62.5%
U_3	7	$\#\{I_6, I_7, I_{12}\} = 3$	$\#\{I_5, I_6, I_7, I_{12}, I_{14}\} = 5$	42, 85%	71, 42%
U_4	10	$\#\{I_2, I_6, I_7, I_8, I_{12}, I_{13}\} = 6$	$\#\{I_2, I_6, I_7, I_8, I_{12}, I_{13}\} = 6$	60%	60%
U_5	9	$\#\{I_1, I_2, I_4, I_5, I_6, I_{14}\} = 6$	$\#\{I_1, I_2, I_4, I_5, I_6, I_{14}\} = 6$	66, 66%	66, 66%
			c	22/41%	25/41%

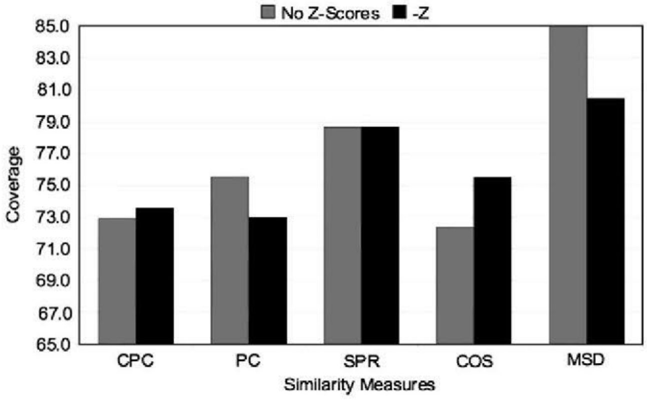

Fig. 6. Coverage results obtained on MovieLens 1M using the similarity measures: Constrained Pearson Correlation (CPC), Pearson Correlation (PC), Spearman rank correlation (SPR), cosine (COS) and Mean Squared Differences (MSD).

$$t = \frac{1}{\#U} \sum_{u \in U} t_u \tag{42}$$

$$x = \frac{1}{\#U} \sum_{u \in U} x_u \tag{43}$$

2.9.2. Running example

In this example we will give parameters N and θ values 4 and 4. Table 11 shows the recommendations Z_u made to each $\boldsymbol{U}_i$ (bottom row of each user) and the votes issued (top row of each user) and Table 12 shows the precision and recall values obtained using $N = 4$ and $\theta = 4$.

2.9.3. Case of study

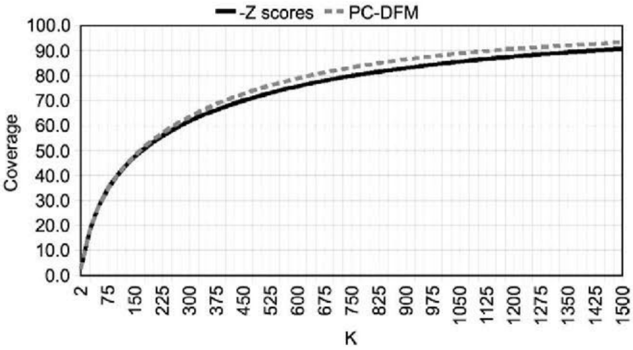

Fig. 7. Breakdown of the coverage obtained on MovieLens 1M using Pearson Correlation (PC).

Table 10
Sets of recommendations that user $U3$ could receive, $K = 2$.

$\tilde{Z}_u$	$N = 2$	$N = 3$	$N = 4$	$N = 5$
U_3	$\{I_5, I_6\}$	$\{I_5, I_6, I_{12}\}$	$\{I_5, I_6, I_{12}, I_7\}$	$\{I_5, I_6, I_{12}, I_7, I_{14}\}$

Fig. 8 shows the average results for precision and recall obtained using the most common similarity measures. Mean Square Differences (MSD) has been rejected due to its poor general results. As may be seen, there is a direct relation between the precision and recall values obtained in each of the cases. It is also important to highlight the fact that, by using similarity measures, high values of accuracy (low values of MAE) do not guarantee the best values for recommendation quality (as is the case in our case study with CPC-WS), in the same way that bad accuracy values can be combined with good results for recommendation quality (see PC-Z-WS). We must take into account that MAE provides us with a measure of the quality of the predictions, whilst precision and recall provide us with a measure of the quality of a small subgroup of the predictions: the N with the highest rating and which are over a certain threshold.

2.10. Quality of novelty: novelty-precision and novelty-recall

2.10.1. Formalization

We will define the novelty group Y as the group of items which have been voted at the most by γ users.

$$Y = \{i \in I | \#\{u \in U | r_{u,i} \neq \bullet\} \leqslant \gamma\} \tag{44}$$

We will use n_u to represent the quality precision measure for novelty obtained by making N test recommendations to user u and requiring a novelty measure γ. Similarly, l_u will represent the recall measure obtained by making the sameN recommendations to user u.

Assuming that all the users accept N test recommendations:

$$n_u = \frac{\#\{i \in Z_u | i \in Y\}}{N} \tag{45}$$

$$l_u = \frac{\#\{i \in Z_u | i \in Y\}}{\#\{i \in Z_u | i \in Y\} + \#\{i \in Z_u^c | i \in Y\}} \tag{46}$$

$$n = \frac{1}{\#U} \sum_{u \in U} n_u \tag{47}$$

$$l = \frac{1}{\#U} \sum_{u \in U} l_u \tag{48}$$

2.10.2. Running example

Firstly, we find the number of votes received for each item, which we represent in the last row of Table 13; later we establish a threshold of novelty ($\gamma = 3$). The set of items belonging to the novelty set is as follows:

$$Y = \{2, 3, 5, 6, 7, 8, 9, 11, 12, 14\}$$

Table 14 shows the recommendations made to each of the users using $N = 4$ and $\theta = 4(Z_u)$; the items belonging to Y (in the first row) are framed. Table 15 shows the novelty-precision and novelty-recall results obtained by each of the users and the total novelty-precision and novelty-recall obtained in the example.

2.10.3. Case of study

Firstly, in order to be able to adjust parameter γ to a suitable value in the RS used, it is valuable to know the distribution of the votes regarding the items. As an example, Fig. 9 shows this data obtained in MovieLens 100 K. Thus, we can determine, for instance, that 600 items of the RS have been voted for by 13 or less users.

Figs. 10 and 11, respectively, show the novelty-precision and novelty-recall results obtained using MovieLens 100 K with values of γ:13, 17, 21 and 25. A general increase in the precision may be noted as we take higher values of γ, due to the gradual increase that this implies in the number of relevant recommended elements.

2.11. Quality of trust: trust-precision and trust-recall

2.11.1. Formalization

As follows from actual results obtained in an experiment carried out on a group of users of the filmaffinity.com website, the trust of user x towards another user y could be based on the following 3 aspects:

- Similarity in the votes.
- Greater importance to the last items voted.
- Number of items that both x and y have voted for ($r_{x,i} \neq \bullet \wedge r_{y,i} \neq \bullet$) in relation to the total number of items voted for by both.

Table 12
Values of precision and recall using $N = 4$, $\theta = 4$.

	U_1	U_2	U_3	U_4	U_5	Average
t_u	3/4	1/4	4/4	3/4	3/4	$t = 0.70$
x_u	3/(3 + 1)	1/(1 + 1)	4/(4 + 1)	3/(3 + 0)	3/(3 + 0)	$x = 0.81$

In order to include time in our model, we extend formulas (4) and (5) to contain a time value in timestamp format.

$$R_u = \{(i, v, t) | i \in I, v \in V, t \in \bullet\} \tag{49}$$

$$r_{u,i} = v \wedge t_{u,i} = t \tag{50}$$

We define $E_{x,y}$ as the group of items that both x and y have voted for most recently. Most recently means within a period of β days as regards the current time (t_c)

$$E_{x,y} = \{i \in I | r_{x,i} \neq \bullet \wedge r_{y,i} \neq \bullet \wedge t_c - t_{x,i} \leqslant \beta \wedge t_c - t_{y,i} \leqslant \beta\} \tag{51}$$

We define S_u as the group of votes of user u which have been made in the time interval β as regards the current time.

$$S_u = \{(i, v, t) | i \in I, v \in V, t \in \bullet, t_c - t_{u,i} \leqslant \beta\} \tag{52}$$

If the votes' time information is not available, Eq. (51) can be simplified in the following way:

$$E_{x,y} = A_{x,y} = \{i \in I | r_{x,i} \neq \bullet \wedge r_{y,i} \neq \bullet\} \tag{53}$$

From the group of items defined in (51), or failing that, in (53), we use the similarity measure which each user will intuitively use to compare their votes with those of each of their neighbors: the Mean Absolute Difference (MAD):

$$\begin{aligned} MAD(x, y, \beta) &= MAD(y, x, \beta) \\ &= \frac{1}{\#E_{x,y}} \sum_{i \in E_{x,y}} |r_{x,i} - r_{y,i}| \Longleftrightarrow E_{x,y} \neq \phi \end{aligned} \tag{54}$$

As a list of common votes among users, as regards the total, we use Jaccard:

$$Jaccard(x, y, \beta) = Jaccard(y, x, \beta) = \frac{S_x \cap S_y}{S_x \cup S_y} \tag{55}$$

Willing to obtain similar importance to metrics (54) and (55), we place the MAD results on the scale [0, 1], where 1 represents the greatest possible similitude and 0 the least possible. We combine both metrics by multiplying them, so that when either of them is low the total similitude is highly affected.

Table 11
Relevant recommended: values with diagonal lines; relevant not recommended: values with horizontal lines.

		I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}
U_1	$r_{1,i}$	5	•	•	3	•	4	1	•	•	4	•	2	4	•
	$\tilde{Z}_1$	4.5	•	•	3.5	•	•	•	•	•	4.33	•	•	4.5	•
U_2	$r_{2,i}$	1	•	•	2	4	1	•	•	•	•	•	•	4	1
	$\tilde{Z}_2$	4.5	•	•	3	•	4	•	•	•	•	•	•	4.5	•
U_3	$r_{3,i}$	5	2	•	4	•	•	•	3	5	4	•	•	4	•
	$\tilde{Z}_3$	3.33	•	•	•	•	•	•	•	5	4	•	•	4	•
U_4	$r_{4,i}$	4	•	•	3	•	•	•	•	5	4	•	•	•	•
	$\tilde{Z}_4$	5	•	•	3.5	•	•	•	•	4.5	4.33	•	•	•	•
U_5	$r_{5,i}$	•	•	•	•	•	•	3	3	4	5	•	•	5	•
	$\tilde{Z}_5$	•	•	•	•	•	•	•	3	5	4	•	•	4	•

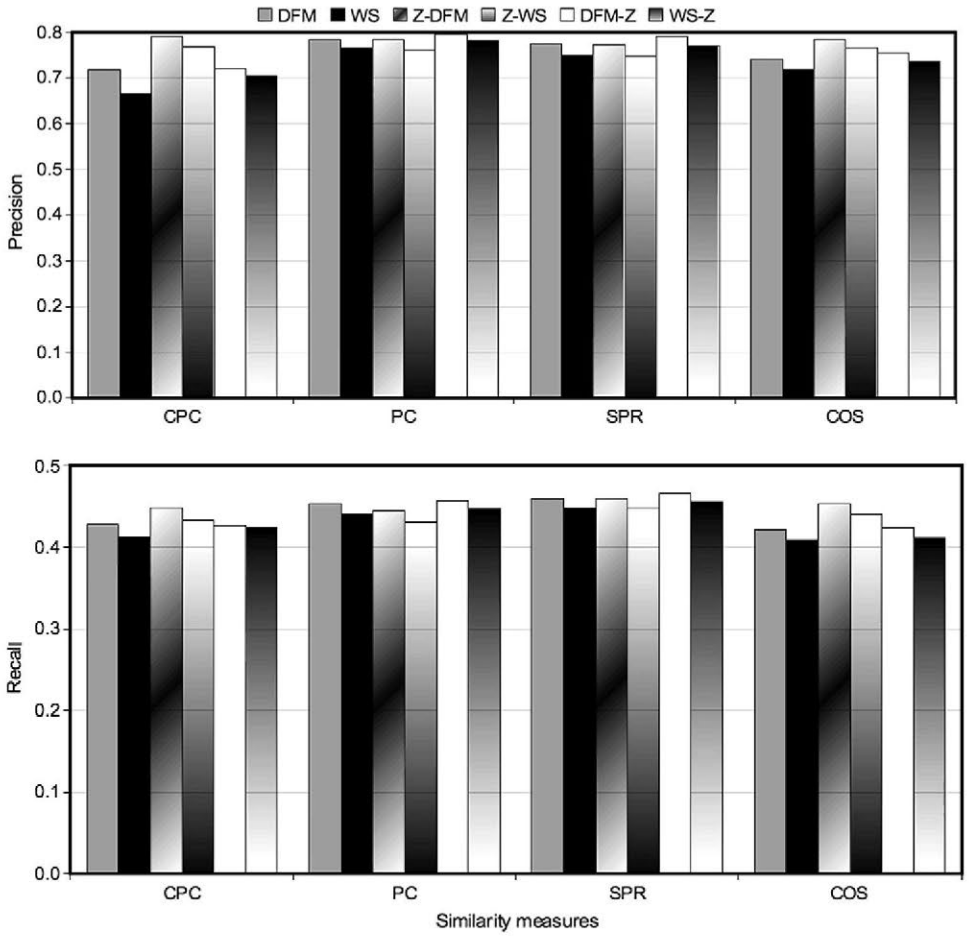

Fig. 8. Precision and Recall results obtained on MovieLens 1M using the similarity measures: Constrained Pearson Correlation (CPC), Pearson Correlation (PC), Spearman rank correlation (SPR) and cosine (COS), making use of the aggregation approaches: Weighted Sum (WS) and Deviation From Mean (DFM), using and not using *z*-scores in the similarity values during the prediction process (−*Z*).

Table 13
Number of users who have voted for each of the items.

$P_{u,i}$	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}
U_1	5	•	•	3	•	4	1	•	•	4	•	2	4	•
U_2	1	•	•	2	4	1	•	•	•	•	•	•	4	1
U_3	5	2	•	4	•	•	•	3	5	4	•	•	4	•
U_4	4	•	•	3	•	•	•	•	5	4	•	•	•	•
U_5	•	•	•	•	•	•	3	3	4	5	•	•	5	•
#	4	1	0	4	1	2	2	2	3	4	0	1	4	1

Table 14
Recommended novelty: values with diagonal lines to the right; not recommended novelty: values with horizontal lines.

Z_u	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}
U_1	4.5	•	•	3.5	•	•	•	•	•	4.33	•	•	4.5	•
U_2	4.5	•	•	3	•	4	•	•	•	•	•	•	4.5	•
U_3	3.33	•	•	•	•	•	•	•	5	4	•	•	4	•
U_4	5	•	•	3.5	•	•	•	•	4.5	4.33	•	•	•	•
U_5	•	•	•	•	•	•	•	3	5	4	•	•	4	•

Table 15
Values of novelty-precision and novelty-recall using $N = 4$, $\theta = 4$, $\gamma = 3$.

	U_1	U_2	U_3	U_4	U_5	Average
n_u	0/4	1/4	1/4	1/4	2/4	$n = 0.25$
l_u	0/(0 + 10)	1/(1 + 9)	1/(1 + 9)	1/(1 + 9)	2/(2 + 8)	$l = 0.10$

Using the ratings times information:

$$trust(x,y,\beta) = trust(y,x,\beta)$$

$$= \frac{S_x \cap S_y}{S_x \cup S_y} \times \left[1 - \frac{\frac{1}{\#E_{x,y}} \sum_{i \in E_{x,y}} |r_{x,i} - r_{y,i}|}{max - min}\right] \iff E_{x,y} \neq \phi \quad (56)$$

$$trust(x,y,\beta) = trust(y,x,\beta) = \bullet \iff E_{x,y} = \phi \quad (57)$$

Without the ratings times information:

$$trust(x,y) = trust(y,x) = \frac{R_x \cap R_y}{R_x \cup R_y} \times \left[1 - \frac{\frac{1}{\#A_{x,y}} \sum_{i \in A_{x,y}} |r_{x,i} - r_{y,i}|}{max - min}\right]$$

$$\iff A_{x,y} \neq \phi \quad (58)$$

$$trust(x,y) = trust(y,x) = \bullet \iff A_{x,y} = \phi \quad (59)$$

Eqs. (10) and (11) can be adapted as:

We define T_u as the set of test *K*-neighbors of user *u* taking into account factor β of proximity in the issuing of votes to the current time.

The following must be true:

$$T_u \subset U \wedge \#T_u = k \wedge u \notin T_u \quad (60)$$

$$\forall x \in T_u, \quad \forall y \in (U - T_u), \quad trust(u,x,\beta) \geqslant trust(u,y,\beta) \quad (61)$$

The set of test *K*-neighbors of a given user (T_u) can be compared with the group of *K*-neighbors obtained using the similarity metric

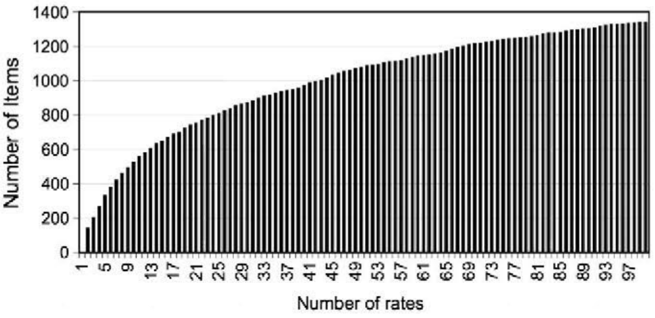

Fig. 9. Number of items (*y* axis) which have been voted for a maximum of *n* (*x* axis) times. MovieLens 100 K Database.

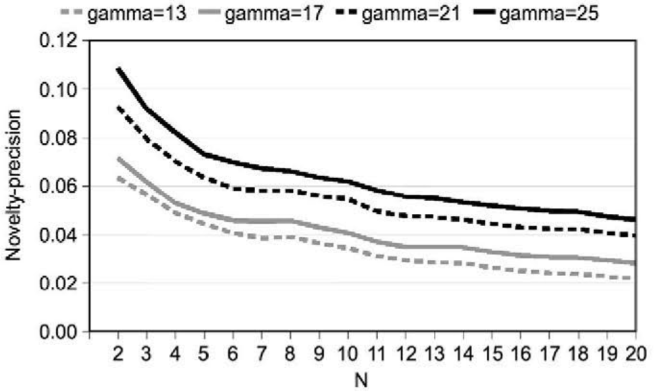

Fig. 10. Precision results obtained on MovieLens 100 K, by taking values of *N* = [2…20], *K* = 250, and using Pearson Correlation.

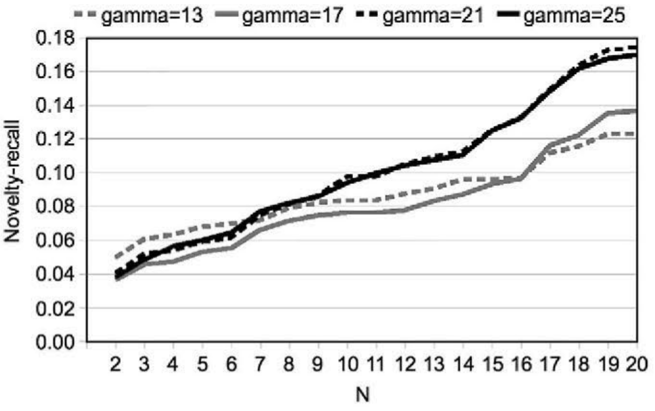

Fig. 11. Recall results obtained on MovieLens 100 K, by taking values of *N* = [2…20], *K* = 250, and using Pearson Correlation.

to be tested (K_u), this way we can create the confusion matrix and obtain the measures of trust-precision and trust-recall.

In addition, this approach allows us to offer each user u the following personalized information regarding the measure of trust offered to them by the recommendation process: the set of "q" users over which the recommendations have been made offering the most trust to the active user (recommended), and for each of these users, group of "h" items whose ratings have been closest to that of the active user and have been made most recently (β). This information aims to show the active user the subgroup of data that played a part in the recommendation that best matches what an average user understands to be their soul mates (users who voted similarly in the most relevant items).

$$\begin{aligned} Q_u = \{K_u \cap T_u\} | \#Q_u \leqslant q \wedge \forall x \in Q_u,\\ \forall y \in Q_u^c trust(u,x,\beta) \geqslant trust(u,y,\beta) \end{aligned} \tag{62}$$

$$\begin{aligned} H_u = \{(u,i), u \in Q_u, i \in I, \#H_u \leqslant h \big| \forall x \in Q_u, \forall y \in Q_u^c \rightarrow |r_{u,i} - r_{x,i}| \\ \leqslant |r_{u,i} - r_{y,i}|\} \end{aligned} \tag{63}$$

We will use w_u to represent the quality precision measure for trust requiring a measure of proximity in time to the current time β. Similarly, z_u will represent the measure recall.

$$w_u = \frac{\#\{u \in K_u | u \in T_u\}}{\#K_u} \tag{64}$$

$$z_u = \frac{\#\{u \in K_u | u \in T_u\}}{\#\{u \in K_u | u \in T_u\} + \#\{u \in K_u^c | u \in T_u\}} \tag{65}$$

$$w = \frac{1}{\#U} \sum_{u \in U} w_u \tag{66}$$

$$z = \frac{1}{\#U} \sum_{u \in U} z_u \tag{67}$$

2.11.2. Running example

Let's assume that by establishing a time value β, the values of Table 16 which are crossed out are excluded from the process to obtain trust measures. Table 17 shows the trust values between users "*trust*(*x*,*y*)" obtained by applying Eqs. (56) and (57). As an example, Table 18 shows the group of $K = 2$ trust neighbors of each of the users, obtained from Table 17, by applying Eqs. (60) and (61).

If we wish to test a similarity metric between users applicable to the CF of the RS, we would obtain, via the usual procedure, the K-neighbors of each user by applying Eqs. (10) and (11). In our running example we will assume that, by using $K = 2$, we obtain the groups represented in Table 19. By creating the confusion matrix between the relevant (Table 18) and the retrieved (Table 19) values and by applying Eqs. (64)–(67) we can obtain the measures of trust-precision and trust-recall outlined in Table 20.

Finally, Table 21 shows the information considered most appropriate to provide to each with the aim that they can easily understand relevant aspects of the recommendation process. Its results are obtained by applying Eqs. (62) and (63) with values $k = 2$, $q = 2$ and $h = 2$.

2.11.3. Case of study

In order to suitably adjust parameter β in the RS used, it is valuable to know the distribution of the votes with regard to their dates. As an example, Fig. 12 shows this data obtained in Movielens

Table 16
Users' votes; recent not crossed out, not recent (as regards $t_c - \beta$) crossed out.

$r_{u,i}$	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}
U_1	5	•	•	3	•	4	1	•	•	4	•	2	4	•
U_2	1	•	•	2	4	1	•	•	•	•	•	•	4	1
U_3	5	2	•	4	•	•	•	3	5	4	•	•	4	•
U_4	4	•	•	3	•	•	•	•	5	4	•	•	•	•
U_5	•	•	•	•	•	•	3	3	4	5	•	•	5	•

Table 17
Measures of trust between users.

Trust	U_1	U_2	U_3	U_4	U_5
U_1	–	–	–	–	–
U_2	1/8∗1 = 0.12	–	–	–	–
U_3	2/10 ∗ 1 = 0.2	1/8 ∗ 1 = 0.12	–	–	–
U_4	1/6 ∗ 1 = 0.16	0/4 ∗ • = •	1/6 ∗ 1 = 0.16	–	–
U_5	3/8 ∗ 0.66 = 0.24	1/7 ∗ 0,75 = 0.10	4/7 ∗ 0.81 = 0.46	1/5 ∗ 0,75 = 0.15	–

Table 18
Set of trust users for each user, using $K = 2$.

T_u	U_1	U_2	U_3	U_4	U_5
$K = 2$	$\{U_5,U_3\}$	$\{U_3,U_1\}$	$\{U_5,U_1\}$	$\{U_3,U_1\}$	$\{U_3,U_1\}$

Table 19
Set of 2-neighbors for each user using a similarity measure that we wish to test.

K_u	U_1	U_2	U_3	U_4	U_5
$k = 2$	$\{U_3,U_2\}$	$\{U_1,U_3\}$	$\{U_2,U_4\}$	$\{U_3,U_2\}$	$\{U_1,U_2\}$

Table 20
Trust-precision and trust-recall obtained.

	U_1	U_2	U_3	U_4	U_5	Average
w_u	1/2	2/2	0/2	1/2	1/2	$w = 0.5$
z_u	$1/(1+1)$	$2/(2+0)$	$0/(0+2)$	$1/(1+1)$	$1/(1+1)$	$z = 0.5$

1M. For instance, we can determine that in the last 4 months, with regard to the last vote issued, approximately 50,000 votes have been made.

Figs. 13 and 14 show, respectively, the results of trust-precision and trust-recall obtained using Movielens 1 M with values of β: 150, 180 and 210. The low values in precision and recall are produced due to the fact that small values of parameter β have also been chosen.

3. Proposed framework and results

The framework with which we propose to test the different CF similarity measure metrics includes the quality analysis of the following aspects: predictions, recommendations, novelty and trust. Once a suitable reference metric is considered, we will be able to compare the results obtained with the proposed metric to those obtained with the reference metric.

In each RS in operation we can decide the importance given to the quality of each of the four aspects included in the framework (predictions, recommendations, novelty and trust) in such a way that the results obtained with a trial metric can be considered positive even if they only improve some of these four aspects.

This section presents the four graphs proposed in the framework backed by the quality results obtained using the MovieLens 1M database, making use of 20% of test users and 20% of test items.

Table 21
Information proposed to be provided to users to increase their trust in the recommendation process (using $k = 2$, $q = 2$ and $h = 2$).

	$q = 1$	$q = 2$
U_1	$U3$ $\{\langle I10,4\rangle,\langle I13,4\rangle\}$	•
U_2	$U1$ $\{\langle I13,4\rangle\}$	$U3$ $\{\langle I13,4\rangle\}$
U_3	Ø	•
U_4	$U3$ $\{\langle I10,4\rangle\}$	•
U_5	$U1$ $\{\langle I10,4\rangle,\langle I13,4\rangle\}$	•

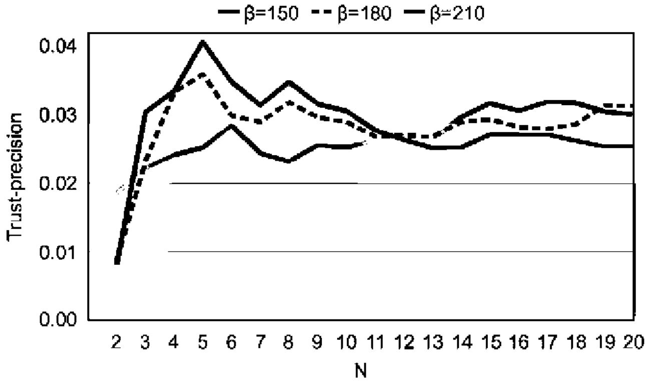

Fig. 13. Results of trust-precision obtained on Movielens 1M, taking values of K=[2...20], and using Pearson Correlation.

In the last subsection we display the results obtained using a more representative database of a large RS in operation: NetFlix.

3.1. Quality of the predictions: accuracy versus coverage

The most widely accepted measure to determine the quality of the estimations is the mean absolute error (MAE), or any of its related metrics: mean squared error, root mean squared error, and normalized mean absolute error. Additionally, it is appropriate for the coverage to remain as high as possible in order to achieve greater extension in the predictions. As established in previous sections, a reverse trend exists between accuracy and coverage, which makes it advisable to evaluate them together when determining the integrity of a metric in the quality of the estimations.

As a measure of the quality of the predictions we will compare, on one graph, the accuracy (y axis) with the coverage (x axis), where the coverage is transferred from traditional percentage to interval [0, 1]. The curve obtained will be considered to be the better, the closer it is to the top right edge (maximum values of

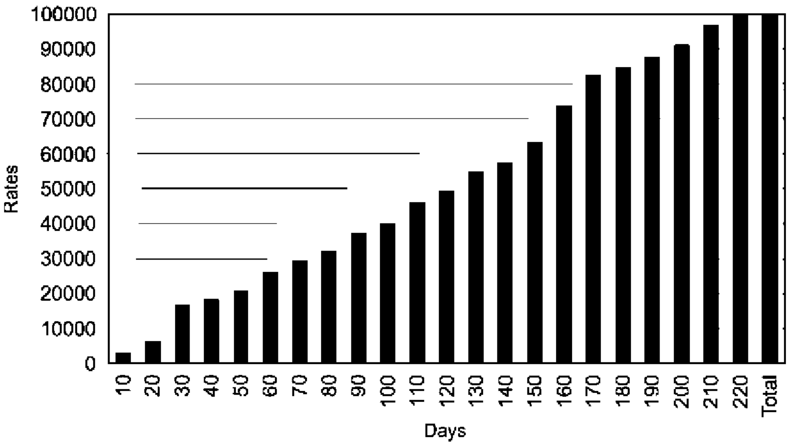

Fig. 12. Number of votes (y axis) that have been issued in a period of n days (x axis) with regard to the last vote issued. Movielens 1M database.

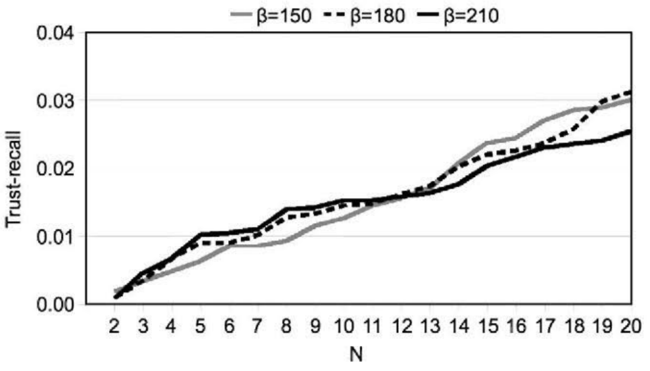

Fig. 14. Results of trust-recall obtained on Movielens 1 M, taking values of $K = [2\ldots20]$, and using Pearson Correlation.

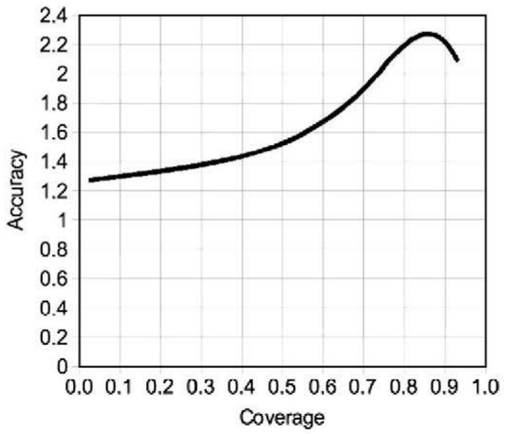

Fig. 15. Quality results of the predictions obtained on MovieLens 1 M using Pearson Correlation, making use of Deviation From Mean and using z-scores in the similarity values during the prediction process (–Z). Range of K's: [2...1500].

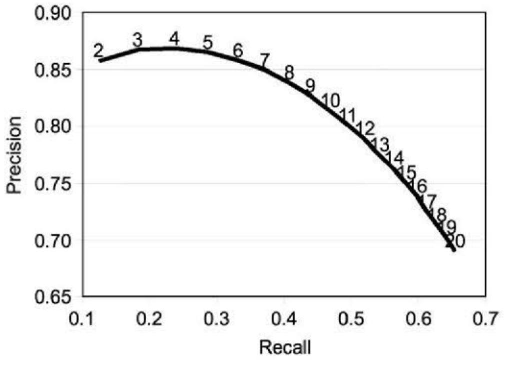

Fig. 16. Precision and recall groups obtained on MovieLens 1 M using Pearson Correlation (PC-DFM-Z), $N = [2\ldots20]$.

coverage and accuracy). The graph is obtained by drawing on the plane each pair of values (coverage, accuracy) obtained for different numbers of *K*-neighbors.

Fig. 15 shows the result of comparing the accuracy with the coverage obtained by applying PC-DFM-Z on the MovieLens 1M database and a range of *K*'s [2...1500], step 25.

3.2. Quality of the recommendations: precision versus recall

In order to measure the quality of the recommendations made, we will use the classic graphs which show the precision/recall combination on one curve, representing the precision on the *y* axis and the recall on the *x* axis. The relevant information is the proximity of the curve to the top right-hand point of the graph (where the maximum values of precision and recall converge). The graphs are developed by selecting different values of *N* (number of recommendations) and displaying the position of each point on the plane (recall, precision) which has been obtained with each of the *N*'s.

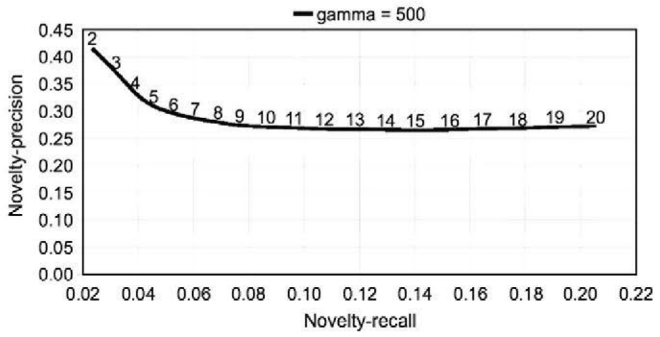

Fig. 17. Results of novelty-precision versus novelty-recall obtained on MovieLens 1 M, taking values of $N = [2\ldots20]$, $K = 800$, $\gamma = 500$, using Pearson Correlation.

Fig. 16 shows the combined results of precision and recall obtained using PC-DFM-Z on MovieLens 1M; the *K* selected is the one which minimizes the system's MAE. As precision and recall tend to be inversely related measures, this type of graph appears with slopes in the style of the one shown in the figure, although the curve varies according to the nature of the RS, the similarity measure and the aggregation approach selected.

3.3. Quality of the novelty: novelty-precision versus novelty-recall

As stated in the previous section, we will compare, in an only graph, the quality measures precision and recall referring to the novelty of the recommendations. Fig. 17 shows the results obtained on the MovieLens 1M database, taking a high value of γ.

3.4. Quality of the trust: trust-precision versus trust-recall

Fig. 18 displays a graph of the trust quality on the MovieLens 1M database, taking a high value of β, with the aim of displaying, in turn, high values of precision.

3.5. Results using NetFlix

Fig. 19 groups the 4 quality graphs obtained using the database NetFlix with 5% of test users, 20% of test items. The quality of the predictions (graph A) is similar to MovieLens and the quality of the recommendations (graph B) is lower. The quality of the novelty and trust factors (graphs C and D) is strongly dependent on the

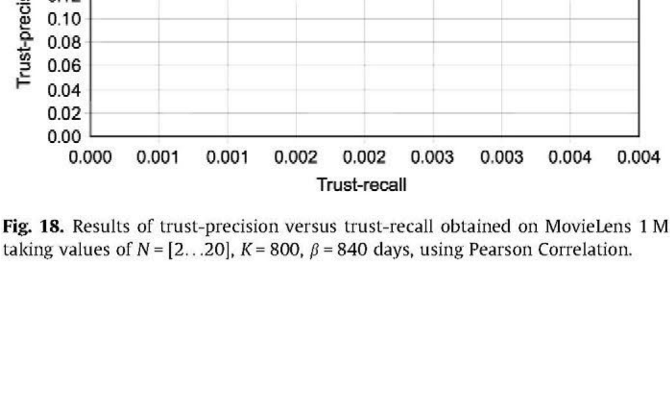

Fig. 18. Results of trust-precision versus trust-recall obtained on MovieLens 1 M, taking values of $N = [2\ldots20]$, $K = 800$, $\beta = 840$ days, using Pearson Correlation.

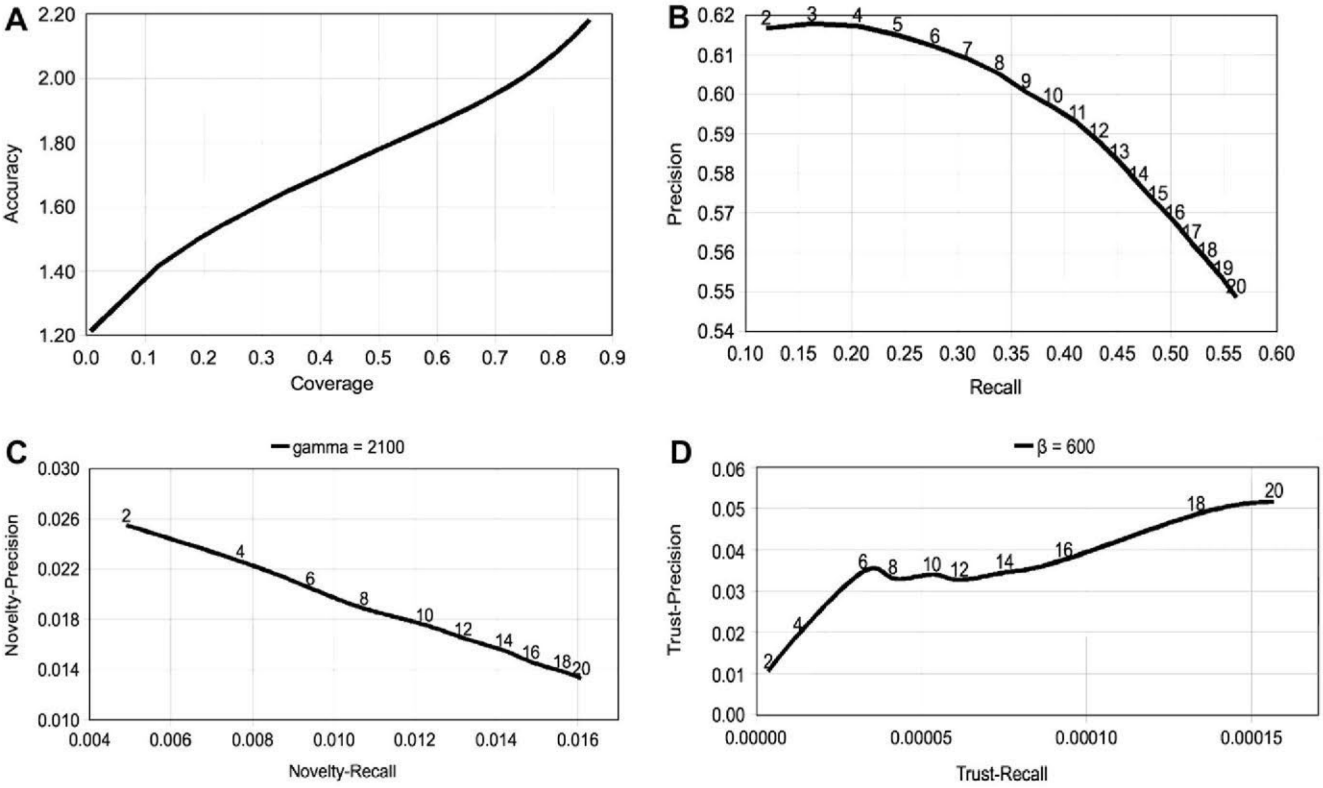

Fig. 19. NetFlix results using the proposed framework. Pearson correlation, $N = [2...20]$, $K = 7000$, β=600 days, $\gamma = 2100$, $\theta = 5$, Accuracy: $K = [2...1000]$.

actual characteristics of the RS analyzed and on the values γ and β selected, which provides us with the possibility of improving the results by analyzing the behavior of the graphs for different values of these parameters, all within the framework provided.

4. Conclusions

It is important for CF frameworks to include the specification of equations used to evaluate the results of the similarity metrics and methods, so that we can make certain that all the experiments are reproducible and comparable and, therefore, it is possible to establish them in a unified way to compare the advantages and disadvantages of the various methods and metrics proposed by the scientific community.

In the field of CF, even though RS show a broad tradition and extensive experience in measuring the quality of the predictions and recommendations, the same is not true for measuring the quality of the novelty and trust results. The novelty and trust quality measures proposed in our paper are based on simple and reasonable cases which offer convincing results tested on existing RS.

The measure proposed to evaluate a user's trust in their K-neighbors entails the additional advantage of providing the active user with *user, item* pairs as a reference of the CF process, with the aim of increasing the trust in the recommendations received.

The results obtained using the framework are expressed in four graphs (quality of the predictions, the recommendations, the novelty and the trust) of which their individual importance can be easily weighted by each researcher according to their purpose, and which above all enables the immediate comparison of the results obtained by applying various similarity measures and metrics to the RS analyzed.

The tests carried out to determine the metric with the better general results in predictions and recommendations have confirmed already published notions about the suitability of Pearson Correlation combined with the Deviation-From-Mean aggregation approach. This metric has been the one used as a reference to generate the results of the framework on MovieLens 1M and NetFlix, establishing a template for use and results, which is suitable for comparing the evaluation values which will be obtained using new similarity metrics or methods.

References

Adomavicius, E., & 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.

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

Baraglia, R., & Silvestri, F. (2004). An online recommender system for large web sites. In *Proceedings of the IEEE/WIC/ACM international conference on web intelligence* (pp. 199–205). doi:10.1109/WI.2004.10158.

Bobadilla, J., Ortega, F., & Hernando, A. (in press). A collaborative filtering similarity measure based on singularities. *Information Processing and Management*. doi:10.1016/j.ipm.2011.03.007.

Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge Based Systems, 23*, 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.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *14th conference on uncertainty in artificial intelligence* (pp. 43–52). Morgan Kaufmann.

Buhwan, J., Jaewook, L., & Hyunbo, Ch. (2009). User credit-based collaborative filtering. *Expert Systems with Applications* (36), 7309–7312. doi:10.1016/j.eswa.2008.09.034.

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

Cho, S. B., Hong, J. H., & Park, M. H. (2007). Location-based recommendation system using bayesian user's preference model in mobile devices. *LNCS, 4611*, 1130–1139.

Fesenmaier, D. R., Gretzel, U., Knoblock, C., Paris, C., Ricci, C., Stabb, S., et al. (2002). Intelligent systems for tourism. *Intelligent Systems, 17*(6), 53–66.

Fuyuki, I., Quan, T. K., & Shinichi, H. (2006). Improving accuracy of recommender systems by clustering items based on stability of user similarity. In *IEEE international conference on intelligent agents, web technologies and internet commerce* (p. 61). doi:10.1109/CIMCA.2006.123.

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

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval, 4*(2), 133–151.

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

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.

Hernández, F., & Gaudioso, E. (2008). Evaluation of recommender systems: A new approach. *Expert Systems with Applications, 1*(35), 790–804. doi:10.1016/j.eswa.2007.07.047.

Hijikata, Y., Shimizu, T., & Nishida S. (2009). Discovery-oriented collaborative filtering for improving user satisfaction. In *Proceedings of 14th international conference on Intelligent user interface* (pp. 67–76). doi:10-1145/1502650.1502663.

Ingoo, H., Kyong, J. O., & Tae, H. R. (2003). The collaborative filtering recommendation based on SOM cluster-indexing CBR. *Expert Systems with Applications, 25*, 413–423.

Jinghua, H., Kangning, W., & Shaohong, F. (2007). A survey of e-commerce recommender systems. In *Proceedings of the international conference on service systems and service management* (pp. 1–5). doi:10.1109/ICSSSM.2007.4280214.

Kitisin, S., & Neuman, C. (2006). Reputation-based trust-aware recommender system. *Securecomm*, 1–7. doi:10.1109/SECCOMW.2006.359555.

Kong, F., Sun, X., Ye, S. (2005). A comparison of several algorithms for collaborative filtering in startup stage. In *Proceedings of IEEE network, sensing and control* (pp. 25–28).

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

Koutrika, G., Bercovitz, B., & Garcia, H. (2009). FlexRecs: Expressing and combining flexible recommendations. *SIGMOD*, 745–757.

Kwiseok, K., Jinhyung, Ch., & Yongtae, P. (2009). Multidimensional credibility model for neighbor selection in collaborative recommendation. *Expert Systems with Applications* (36), 7114–7122. doi:10.1016/j.eswa.2008.08.071.

Li, P., & Yamada, S. (2004). A movie recommender system based on inductive learning. In *IEEE conference on cybernetics and intelligent systems* (Vol. 1, pp. 318–323). doi:10.1109/ICCIS.2004.1460433.

Manolopoulus, Y., Nanopoulus, A., Papadopoulus, A. N., & Symeonidis, P. (2007). Collaborative recommender systems: Combining effectiveness and efficiency. *Expert Systems with Applications, 4*(34), 2995–3013. doi:10.1016/j.eswa.2007.05.013.

O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. In *IUI'05* (pp. 9–12).

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

Schafer, J. B., Frankowski, D., Herlocker, J. L., & Sen, S. (2007). Collaborative filtering recommender systems. *LNCS, 4321*, 291–324.

Serrano, J., Viedma, E. H., Olivas, J. A., Cerezo, A., & Romero, F. P. (2011). A Google wave-based fuzzy recommender system to disseminate information in university digital libraries 2.0. *Information Sciences, 181*(8), 1503–1516.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2–19. doi:10.1155/2009/421425.

Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2008). Providing justifications in recommender systems. *IEEE Transactions on Systems, Man and Cybernetics, 38*(6), 1262–1272. doi:10.1109/TSMCA.2008.2003969.

Yager, R. R. (2003). Fuzzy logic methods in recommender systems. *Fuzzy Sets and Systems, 136*(2), 133–149.

Zhang, F. (2008). Research on recommendation list diversity of recommender systems. *ICMECG*, 72–76. doi:10.1109/ICMECG.2008.32.