A new collaborative filtering metric that improves the behavior of recommender systems

J. Bobadilla *, F. Serradilla, J. Bernal

Universidad Politécnica de Madrid, Computer Science, Ctra. de Valencia, Km 7, 28031 Madrid, Spain

ARTICLE INFO

Article history:
Received 21 July 2009
Received in revised form 18 March 2010
Accepted 19 March 2010
Available online 23 March 2010

Keywords:
Collaborative filtering
Recommender systems
Metric
Jaccard
Mean squared differences
Similarity

ABSTRACT

Recommender systems are typically provided as Web 2.0 services and are part of the range of applications that give support to large-scale social networks, enabling on-line recommendations to be made based on the use of networked databases. The operating core of recommender systems is based on the collaborative filtering stage, which, in current user to user recommender processes, usually uses the Pearson correlation metric. In this paper, we present a new metric which combines the numerical information of the votes with independent information from those values, based on the proportions of the common and uncommon votes between each pair of users. Likewise, we define the reasoning and experiments on which the design of the metric is based and the restriction of being applied to recommender systems where the possible range of votes is not greater than 5. In order to demonstrate the superior nature of the proposed metric, we provide the comparative results of a set of experiments based on the MovieLens, FilmAffinity and Netflix databases. In addition to the traditional levels of accuracy, results are also provided on the metrics’ coverage, the percentage of hits obtained and the precision/recall.

1. Introduction

Recommender systems (RS) cover an important field within collaborative services that are developed in the Web 2.0 environment [21,19,26] and enable user-generated opinions to be exploited in a sophisticated and powerful way. Recommender systems can be considered as social networking tools that provide dynamic and sophisticated and powerful way. Recommender systems can be considered as social networking tools that provide dynamic and collaborative communication, interaction and knowledge.

Recommender systems cover a wide variety of applications [20,4,10,28,5,14,27], although those related to movie recommendations are the most well-known and most widely-used in the research field [23,3,25].

The recommender systems stage that normally has the greatest influence on the quality of the results obtained is the collaborative filtering (CF) phase [1,17,6,34,33]. Collaborative filtering is based on making predictions about a user’s preferences or tastes based on the preferences of a group of users that are considered similar to this user. A substantial part of the research in the area of collaborative filtering centers on how to determine which users are similar to the given one; in order to tackle this task, there are fundamentally 3 approaches: memory-based methods, model-based methods and hybrid approaches.

Memory-based methods [22,37,35,40] 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 [1] 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 [8], neural networks [18] and fuzzy systems [39]. Generally, commercial recommender systems use memory-based methods, whilst model-based methods are usually associated with research recommender systems.

Regardless of the method used in the collaborative filtering stage, the technical objective generally pursued is to minimize the prediction errors, by making the accuracy [12,11,29] of the recommender systems as high as possible; nevertheless, there are other objectives that need to be taken into account [38]: avoid overspecialization phenomena, find good items, credibility of recommendations, precision and recall measures, etc.

To date, various publications have been written which tackle the way the recommender systems are evaluated, among the most significant we have [17] which reviews the key decisions in evaluating collaborative filtering recommender systems: 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 [9] 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 et al. [13] focuses on the aspects not related to the evaluation, Breese et al. [6] compare the predictive accuracy of various methods in a set of representative problem domains. Candillier et al. [7] and Schafer et al. [36] review the main collaborative filtering methods proposed in the literature.

The rest of the paper is structured as follows:

- In Section 2 we provide the basis for the principles on which the design of the new metric will be based, we present graphs which show the way in which the users vote, we carry out experiments which support the decisions made, we establish the best way of selecting numerical and non-numerical information from the votes and, finally, we establish the hypothesis on which the paper and its proposed metric are based.
- In Section 3 we establish the mathematical formulation of the metric.
- In Sections 4 and 5, respectively, we list the experiments that will be carried out and we present and discuss the results obtained.
- Section 6 presents the most relevant conclusions of the publication.

2. Approach and design of the new similarity metric

2.1. Introduction

Collaborative filtering methods work on a table of $U$ users who can rate $I$ 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 ($k$-neighborhoods) for the same item $i$, where $K_u$ denotes the set of $k$-neighborhood of user $u$ and $r_{ui}$ denotes value of the user $n$ rating on the item $i$ (if there is not rating value).

Once the set of $K$ users (neighborhoods) similar to active user 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). We will use the auxiliary set $G_u$, in order to define Eqs. (2)-(5):

$$G_u = \{ n \in K_u | \exists r_{ui} \neq \bullet \},$$

$$p_{ui} = \frac{1}{|G_u|} \sum_{n \in G_u} r_{ui} \iff G_u \neq \emptyset,$$

$$p_{ui} = \mu_g \sum_{n \in G_u} \text{sim}(u, n) r_{ni} \iff G_u \neq \emptyset,$$

$$p_{ui} = \bar{r}_u + \mu_g \sum_{n \in G_u} \text{sim}(u, n) (r_{ni} - \bar{r}) \iff G_u \neq \emptyset,$$

where $\mu$ serves as a normalizing factor, usually computed:

$$\mu_g = 1 / \sum_{n \in G_u} \text{sim}(u, n) \iff G_u \neq \emptyset.$$

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

$$\text{sim}(x, y) = \frac{\sum (r_{xi} - \bar{x})(r_{yi} - \bar{y})}{\sqrt{\sum (r_{xi} - \bar{x})^2 \sum (r_{yi} - \bar{y})^2}},$$

$$\text{sim}(x, y) = \frac{\sum r_{xi} r_{yi}}{\sqrt{\sum r_{xi}^2 \sum r_{yi}^2}},$$

$$\text{sim}(x, y) = \frac{\sum (\text{rank}_ix - \text{rank}_x)(\text{rank}_iy - \text{rank}_y)}{\sqrt{\sum (\text{rank}_ix - \text{rank}_x)^2 \sum (\text{rank}_iy - \text{rank}_y)^2}},$$

$$\text{sim}(x, y) = \frac{\sum (\text{rank}_ix - \text{rank}_x)(\text{rank}_iy - \text{rank}_y)}{\sqrt{\sum (\text{rank}_ix - \text{rank}_x)^2 \sum (\text{rank}_iy - \text{rank}_y)^2}}.$$
to represent a positive or non-positive rating of the items, and to a lesser extent a range of these ratings; for instance, in a RS with possible votes situated in the interval \([1..5]\), 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 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. 3 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 determine whether an item is recommended (based on the number of \(k\)-neighborhoods 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).

Fig. 1. Distribution of votes in RS: (A) MovieLens 1M and (B) NetFlix.

Fig. 2. Arithmetic average and standard deviation on the MovieLens 1M and NetFlix ratings of the items. (A) MovieLens arithmetic average, (B) NetFlix arithmetic average, (C) MovieLens standard deviation, (D) NetFlix standard deviation.

Fig. 3. 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, \(\theta = 4\).
2.3. Non-numerical information: Jaccard

As the numerical values of the votes appear to lose relevance in the recommendation process (not in the prediction process), we are obliged to search for similarity information between users by looking beyond the specific values of their votes. In this sense, it is reasonable to focus our attention on two related aspects:

1. To not grant too much credibility to the similarity of two users based only on the similarity of a very limited set of common items: traditional metrics provide their measure of similarity without taking into account whether it has been obtained based on few or many items voted by both users; this way, it is not improbable to find the highest similarity measurements associated with pairs of users with very few items commonly voted.
2. The users who have voted for a large number of items should be compared, in so far as is possible, with other users with whom they have a large number of items voted in common, this way, for example, in users with around one thousand votes cast, a similarity of 0.78 calculated with some 150 common items is more convincing than a similarity of 0.82 calculated with 60 common items. In short, the proportion between the common votes and the total votes should be taken very much into consideration.

The most direct way to quantify the above-mentioned aspects is by using the Jaccard metric [24], which calculates the proportion between the number of items that two users have voted in common and the number of different items that both users have voted for in total, i.e. the intersection divided by the union of the items voted. In order to design our new metric correctly, we must discover the impact that Jaccard can have as a factor of similarity between users. As Jaccard does not operate with the numerical values of the votes, it seems improbable that, on its own, it will be able to have a positive impact on the MAE of the RS, however, it is more apparent that the coverage can improve by selecting users with more common votes and therefore, in general, with more votes cast.

On the other hand, it is reasonable to suspect that users who have voted for a sufficient proportion of items in common display common tastes: we know that the negative ratings are not very widely-used (Figs. 1 and 2) and therefore, we can theorize that part of the common absences of votes between 2 users can mean common tastes: we know that the negative ratings are not very widely-used (Figs. 1 and 2), and therefore, many votes in common could denote a large number of positive ratings in common.

In order to try to test this hypothesis, we have designed the following experiment: for every possible pair of users of MovieLens 1M we have calculated the Jaccard value, the MAE and the coverage obtained by establishing the first user of the pair as the active user and the second one as their only neighborhood. On the x axis we represent the possible Jaccard values, represented in the interval [0..1], where 0 indicates that there are no items in common and 1 indicates that all the items voted are in common.

In Fig. 4, graph 4A shows the number of pairs of users (y axis) which display the Jaccard values indicated on the x axis. As is to be expected, most of the cases present a very low overlap between the items voted by each pair of users. Graph 4B shows a direct relationship between the increase in the value of Jaccard and the accuracy obtained in the interval [0..0.4], in which the great majority of the cases are grouped together (Graph 4A). Graph 4C shows a direct relationship between the Jaccard value and an improvement in the coverage.

The reasoning given and the results obtained justify the incorporation of the Jaccard metric as an integral part of a new metric for the similarity between users, which aims to improve the results provided by traditional metrics.

2.4. Numerical information: mean squared differences

By unifying the concepts and results set out in Section 2, we find the following:

1. The similarity between two users (core of the CF RS) is being based on numerical metrics of statistical origin (such as Pearson correlation) which should be applied to continuous variables, but which in fact are applied to discrete variables which, for the purposes of recommendation, only contain 2 values of use (positive/non-positive).
2. We are not making use of non-numerical information of the votes which could be valuable in order to complement the numerical information and provide a metric which satisfactorily groups these two sources of information.

The most commonly used metrics (constrained Pearson correlation, Spearman rank correlation, cosine, Pearson correlation, etc.) display, to a greater or lesser extent, the deficiencies set out in reference to Pearson correlation; however, mean squared differences (MSD), based on the geometrical principles of the Euclidean distance, provide different characteristics which could be suitably complemented with Jaccard. This metric has been tested into [35]; this study highlights the good accuracy results obtained using MSD, but at the cost of coverage values that make it unviable in general RS applications. The following paragraph of the conclusions of [35] sums up its possibilities: “the MSD metric offers very interesting results due to its different behavior compared to the other two studied (cosine and Pearson correlation) and its good results in all aspects except in the coverage, which is undoubtedly its weak point”.

2.5. Hypothesis

The hypothesis on which this paper is based is that a suitable combination of Jaccard and MSD could complement Jaccard with the numerical values of the votes, and could mitigate the deficiencies in the coverage entailed in the use of MSD, in such a way that their joint use would enable the improvement of the results of traditional metrics in general and of Pearson correlation in particular, which will be used as the metric of reference for which the results must be improved.

Although, a priori, the choice of the similarity measure MSD seems to be the most suitable, it is advisable to test Jaccard combined not only with MSD, but also with the most common metrics: Pearson correlation (PC) [6], cosine (COS) [7], constrained Pearson’s correlation (CPC) [8] and Spearman rank correlation (SRC) [9].

In Fig. 5, graph 5A shows the evolution of the MAE using the metrics PCA, Jaccard + PCA, Jaccard + COS, Jaccard + SRC, Jaccard + CPC, Jaccard + SRC and Jaccard + (1 – MSD) applied to the MovieLens 1M database. As we expected, the best results are obtained by MSD. Graph 5B shows the coverage obtained by applying these same metrics; in this case, COS and MSD give the best results up to K = 300 and SRC gives the best results from this value of K. In short, the similarity measure MSD is confirmed as the best option in order to test the hypothesis in this paper.

3. Formalization of the new metric

The metric presented in this paper takes as reference to be improved the most widely-used metric in user to user memory-based CF: Pearson correlation; however, the operating principals that rule this metric will not be taken as a base, but rather, we will use the mean squared difference (MSD) metric, which is much less
commonly used due to its low capacity to produce new recommendations.

MSD offers both a great advantage and a great disadvantage at the same time; the advantage is that it generates very good general results: low average error, high percentage of correct predictions and low percentage of incorrect predictions: the disadvantage is that it has an intrinsic tendency to choose as similar users to one given user those users who have rated a very small number of items \([35]\), e.g. if we have 7 items that can be rated from 1 to 5 and three users \(u_1, u_2, u_3\) with the following ratings:

- \(u_1: (\bullet, \bullet, 4, \bullet, \bullet, \bullet)\),
- \(u_2: (3, 4, 5, 5, 1, 4, \bullet)\),
- \(u_3: (3, 5, 4, 5, \bullet, 3, \bullet)\) (\(\bullet\) means not rated item),

the MSD metric will indicate that \((u_1, u_3)\) have a total similarity (0), \((u_1, u_2)\) have a similarity 0.5 and \((u_2, u_3)\) have a lower similarity (0.6). This situation is not convincing, as intuitively we realize \(u_2\) and \(u_3\) are very similar, whilst \(u_1\) is only similar to \(u_2\) and \(u_3\) in 2 ratios, and, therefore, it is not logical to choose it as the most similar to them, and what is worse, if it is chosen it will not provide us with possibilities to recommend new items.

The strategy to follow to design the new metric is to considerably raise the capacity to generate MSD predictions, without losing along the way its good behavior as regards accuracy and quality of the results.

The metric designed is based on two factors:

- The similarity between two users calculated as the mean of the squared differences (MSD): the smaller these differences, the greater the similarity between the 2 users. This part of the metric enables very good accuracy results to be obtained.
- The number of items in which both one user and the other have made a rating regarding the total number of items which have been rated between the two users. E.g. given users \(u_1: (3, 2, 4, \bullet, \bullet, \bullet)\) and \(u_2: (\bullet, 4, 3, \bullet, 1)\), a common rating has been made in two items as regards a joint rating of five items. This factor enables us to greatly improve the metric’s capacity to make predictions.

An important design aspect is the decision whether not to use a parameter for which the value should be given arbitrarily, i.e. the result provided by the metric should be obtained by only taking the values of the ratings provided by the users of the RS.

By working on the 2 factors with standardized values \([0..1]\), the metric obtained is as follows: Given the lists of ratings of 2 generic users \(x, y\) with \(r_i^x, r_i^y\) \(i \in [1..n]\), the metric is calculated as:

\[
\text{MSD}(x, y) = \frac{1}{n} \sum_{i=1}^{n} (r_i^x - r_i^y)^2
\]

where \(n\) is the number of ratings provided by the users of the RS.

Fig. 4. Measurements related to the Jaccard metric on MovieLens. (A) Number of pairs of users that display the Jaccard values represented on the x axis. (B) Averaged MAE obtained in the pairs of users with the Jaccard values represented on the x axis. (C) Averaged coverages obtained in the pairs of users with the Jaccard values represented on the x axis.

Fig. 5. MAE and coverage obtained with Pearson correlation and by combining Jaccard with Pearson correlation, cosine, constrained Pearson’s correlation, Spearman rank correlation and mean squared differences. (A) MAE, (B) Coverage. MovieLens 1M, 20% of test users, 20% of test items, \(k \in [2..1500]\) step 25.
vote is "not voted", which we represent with the symbol ●. All the lists have the same number of elements: \( l \).

**Example:**

\[
\begin{align*}
 r_x^b & \in \{1..5\} \cup \{\bullet\}, \\
 r_x : (4.5, \bullet, 3.2, \bullet, 1.1), & \quad r_y : (4.3, 1.2, \bullet, 3.4, \bullet).
\end{align*}
\]

using standardized values \([0..1]\):

\[
\begin{align*}
 r_x : (0.75, 1.\text{●}, 0.5, 0.25, \text{●}, 0.0), & \quad r_y : (0.75, 0.5, 0.25, \text{●}, 0.5, 0.75, \bullet).
\end{align*}
\]

We define the cardinality of a list: \#l as the number of elements in the list \( l \) different to \( \bullet \).

1. We obtain the list

\[
\begin{align*}
 d_{xy} : (d_{xy}^1, d_{xy}^2, d_{xy}^3, \ldots, d_{xy}^l)
\end{align*}
\]

\[
\begin{align*}
 d_{xy}^i = (r_{xy}^i - r_{xy}^j)^2; \forall i \neq j; \forall r_{xy}^i \neq \bullet, & \quad d_{xy} = \bullet \forall i \neq j; \forall r_{xy}^i = \bullet,
\end{align*}
\]

in our example:

\[
\begin{align*}
 d_{xy} = (0.25, \bullet, 0.0625, \bullet, 0.5625, \bullet).
\end{align*}
\]

2. We obtain the MSD(\( x, y \)) measure computing the arithmetic average of the values in the list \( d_{xy} \):

\[
\begin{align*}
 MSD(x, y) = \bar{d}_{xy} = \frac{\sum_{i=1}^{l} d_{xy}^i}{\#d_{xy}}.
\end{align*}
\]

in our example:

\[
\begin{align*}
 d_{xy} = (0 + 0.25 + 0.0625 + 0.5625) / 4 = 0.218
\end{align*}
\]

MSD(\( x, y \)) tends towards zero as the ratings of users \( x \) and \( y \) become more similar and tends towards 1 as they become more different (we assume that the votes are normalized in the interval \([0..1]\)).

3. We obtain the Jaccard(\( x, y \)) measure computing the proportion between the number of positions \([1..l]\) in which there are elements different to \( \bullet \) in both \( r_x \) and \( r_y \) regarding the number of positions \([1..l]\) in which there are elements different to \( \bullet \) in \( r_x \) or \( r_y \):

\[
\begin{align*}
 Jaccard(x, y) = \frac{\#d_{xy} \cap \#d_{xy}}{\#d_{xy} \cup \#d_{xy} - \#d_{xy}}.
\end{align*}
\]

in our example:

\[
\begin{align*}
 4 / (6 + 6 - 4) = 0.5
\end{align*}
\]

4. We combine the above elements in the final equation:

\[
\begin{align*}
 \text{newmetric}(x, y) = Jaccard(x, y) \times (1 - MSD(x, y)).
\end{align*}
\]

in the running example:

\[
\begin{align*}
 \text{newmetric}(x, y) = 0.5 \times (1 - 0.218) = 0.391.
\end{align*}
\]

If the values of the votes are normalized in the interval \([0..1]\), then:

\[
\begin{align*}
 (1 - MSD(x, y)) \land Jaccard(x, y) \land \text{newmetric}(x, y) \in [0..1].
\end{align*}
\]

4. **Planning the experiments**

The RS databases [2,30,32] that we use in our experiments present the general characteristics summarized in **Table 1**.

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 rated by that user for that film.

The previous experiments were carried out, depending on the size of the database, for each of the following \( k \)-neighborhoods values: MovieLens \([2..1500]\) step 50, FilmAffinity \([2..2000]\) step 100, NetFlix \([2..10000]\) step 100, due to the fact that depending on the size of each particular RS database, it is necessary to use a different number of \( k \)-neighborhoods in order to display tendencies in the graphs that display their results. The precision/recall recommendation quality results have been obtained using a range \([2..20]\) of recommendations 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 2** shows the numerical data exposed in this section.

5. **Results**

In this section we present the results obtained using the databases specified in **Table 1**. **Fig. 6** shows the results obtained with MovieLens, **Fig. 7** shows those obtained with NetFlix and **Fig. 8** corresponds to FilmAffinity.

**Graph 6A** shows the MAE error obtained in MovieLens by applying Pearson correlation (dashed) and the proposed metric (continuous). The new metric achieves significant fewer errors in practically all the experiments carried out (by varying the number of \( k \)-neighborhoods). The percentage improvement average is around 0.2 stars in the most commonly used values of \( k \) (50, 100, 150, 200).

**Graph 6B** shows us the coverage. Small values of \( k \) produce small percentages in the capacity for prediction, as it is more improbable that the few neighbors of a test user have voted for a film that this user has not voted for. As the number of neighbors increases, the probability that at least one of them has voted for the film also increases, as shown in the Graph.

### Table 1

<table>
<thead>
<tr>
<th>Main parameters of the databases used in the experiments.</th>
<th>MovieLens</th>
<th>FilmAffinity</th>
<th>NetFlix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of users</td>
<td>4382</td>
<td>26447</td>
<td>480189</td>
</tr>
<tr>
<td>Number of movies</td>
<td>3952</td>
<td>21128</td>
<td>17770</td>
</tr>
<tr>
<td>Number of ratings</td>
<td>1000209</td>
<td>19126278</td>
<td>100480507</td>
</tr>
<tr>
<td>Min and max values</td>
<td>1–5</td>
<td>1–10</td>
<td>1–5</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th>K (MAE, coverage, perfect predictions)</th>
<th>Precision/recall</th>
<th>Test users (%)</th>
<th>Test items (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>N</td>
<td>( \theta )</td>
<td></td>
</tr>
</tbody>
</table>
The comparative results in Graph 6B show improvements of up to 9% when applying the new metric as regards the correlation. This is a very good result, as higher values of accuracy normally imply smaller capabilities for recommendation.

Graph 6C shows the percentage of perfect estimations as regards the total estimations made. Perfect estimations are those which match the value voted by the test user, taking as an estimation the rounded value of the aggregation of the $k$-neighborhoods. The values obtained in Graph 6C show us a convincing improvement in the results of the new metric regarding correlation, even by 15% in some cases.

Graph 6D shows the recommendation quality measure: precision versus recall. Although the prediction results (graphs A and C) of the new metric greatly improve the Pearson correlation ones, that improvement is not transferred to the same extent to the recommendation quality results (approximate improvement of 0.02). In order to better understand this detachment between prediction quality and recommendation quality we must remember that with

The comparative results in Graph 6B show improvements of up to 9% when applying the new metric as regards the correlation. This is a very good result, as higher values of accuracy normally imply smaller capabilities for recommendation.

Graph 6C shows the percentage of perfect estimations as regards the total estimations made. Perfect estimations are those which match the value voted by the test user, taking as an estimation the rounded value of the aggregation of the $k$-neighborhoods. The values obtained in Graph 6C show us a convincing improvement in the results of the new metric regarding correlation, even by 15% in some cases.

Graph 6D shows the recommendation quality measure: precision versus recall. Although the prediction results (graphs A and C) of the new metric greatly improve the Pearson correlation ones, that improvement is not transferred to the same extent to the recommendation quality results (approximate improvement of 0.02). In order to better understand this detachment between prediction quality and recommendation quality we must remember that with
precision and recall we are using the concept of relevant recommendations (determined by the threshold \(\theta = 5\) in our experiment). Based on an improvement in the MAE of 0.2 stars, we will be capable, on many occasions, of suitably classifying the items which obtained a prediction of 4.3–4.49 (considered irrelevant) with Pearson correlation and which, with the new metric, we will place, often correctly, above 4.5, and therefore, we will consider them relevant.

As we can see, in the recommendation quality measurements we are dealing with more restrictive numerical margins than those which use the prediction quality measurements, and therefore, it is advisable to consider all kinds of improvements positively.

In short, using the MovieLens database, the proposed metric improves the prediction quality measures and the coverage. The recommendation quality measures are slightly improved.

The MAE and perfect predictions (graphs 7A and 7C) results obtained with Netflix are similar (although slightly lower) to those of MovieLens. Nevertheless, the coverage drops using Netflix (graph 7B); this behavior is logical and is to be expected, as the proposed metric is capable of finding more similar neighbors (which improve the measure of accuracy); the more similar the neighbors are to the test user, in general, not only will they have more similar vote values, but also they will have a greater tendency to vote for the same subset of the total films rated (the same genres, in the same years, etc.); although this factor has been alleviated by the use of Jaccard, its impact has not been completely eliminated.

The Netflix precision/recall quality measure (graph 7D) improves significantly when the number of recommendations is not high (between 2 and 5), remaining similar to Pearson correlation when the number of recommendations is high.

Fig. 8 shows the results obtained using the FilmAffinity database. In summary, the new metric offers results which cannot be considered better than those provided by Pearson correlation. Indeed, Pearson correlation requires 2 subtraction operations and 3 multiplication-addition operations (assimilating the squaring operation with the multiplication) for every pair of items voted in common (inside the summations), whilst the MSD only requires one subtraction operation and one multiplication-addition operation. Outside the summation (and therefore less significant in the average times), Pearson correlation requires a square-root calculation and the proposed metric only requires 2 divisions and one multiplication. Finally, the time required for the Jaccard calculation has proven to have very little relevance, as the values are obtained efficiently in the same process (loop) used in the summations.

6. Conclusions

In recommender systems which base the votes on small ranges of values (e.g. from 1 to 5), it occurs that users tend to cast their ratings as “positive” or “non-positive”, transferring those conceptual levels towards numerical values, which is reflected in low uniformity in the range of votes cast and little variation in the ratings of each item by the set of users of the recommender systems.

---

**Table 3**

Quality of the results obtained by applying the quality measurements on the selected databases.

<table>
<thead>
<tr>
<th>Metric</th>
<th>MovieLens</th>
<th>Netflix</th>
<th>FilmAffinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>++</td>
<td>++</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Perfect predictions</td>
<td>++</td>
<td>++</td>
<td></td>
</tr>
<tr>
<td>Coverage</td>
<td>+</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>Precision/recall</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
</tbody>
</table>
To complement the numerical values with additional non-numerical information, focused on the arrangement of the votes of each pair of users, we manage to improve the results of Pearson correlation; for this purpose we use the Jaccard measure and the traditional metric that best adapts to its characteristics: mean square differences. The new metric only operates with the data (ratings) provided by the users of the recommender systems, and does not require any arbitrary parameters for adjustment or weighting.

With the aim of achieving representative results the experiments have been carried out on three different recommender systems databases (MovieLens, FilmAffinity and NetFlix) which provide a sufficient volume and variety of data in order to offer reliable comparative results and general conclusions. The results confirm the integrity of the metric proposed when applied to MovieLens and NetFlix (with range of votes [1..5]), whilst the results do not improve that of Pearson correlation when it is applied to FilmAffinity (with range of votes [1..10]).

Acknowledgements

Our acknowledgement to the GroupLens Research Group, and to the FilmAffinity and NetFlix companies.

References


