Towards Preference Relations in Recommender Systems

Armelle Brun[1], Ahmad Hamad[2], Olivier Buffet[3], and Anne Boyer[1]

[1] LORIA-Nancy Université
{armelle.brun, anne.boyer}@loria.fr,
[2] Sailendra SAS
ahmad.hamad@loria.fr
[3] LORIA - INRIA
olivier.buffet@inria.fr
615, avenue du jardin botanique, 54506 Vandoeuvre les Nancy

Abstract. Collaborative filtering-based recommender systems exploit user preferences about items to provide them with recommendations. These preferences are generally ratings. However, choosing a rating is no easy task for any user; the rating scale is usually reduced and the rating values given by the users may be influenced by many factors. The ratings are thus not completely trustworthy. This paper is a first attempt at studying the expression of preferences in collaborative filtering under the form of preference relations instead of ratings. When using preference relations, users are asked to compare pairs of resources. We propose new measures to compute recommendations using preference relations. First experiments have been conducted on a state of the art corpus of the recommender systems domain and show that this new approach compares with, and in some cases improves the classical one.

Keywords: Preference relations, Recommender systems, Collaborative filtering

1 Introduction

The democratization of the Internet and network technologies has resulted in a large increase in the volume of information easily accessible to everybody. This increase has been an advantage during its first years as the information access became generalized. However, this volume of information is now so huge that users cannot get easily the information they search, they are drowned in the mass of resources. This irrefutable overabundance has thus the consequence to lead to unsatisfied users. Thus a critical issue of the current Web applications is the incorporation of mechanisms for delivering information that fits users' attempts.

Recommender systems (RS) are such a mechanism, they aim at recommending items to users. These items are linked to the users' expectations and tastes. The use of recommender systems results in a decrease of the time spent by users in their search. Moreover, recommender systems suggest users pertinent items that they would not consult on their own initiative (they may not know of the existence of such items). Users' satisfaction is thus increased. An item (also called a resource) can for example be a web page, a book, a movie, music, etc. Many websites have already integrated recommender systems, such as *amazon* [4] (books), or *last.fm* [5] (music).

To recommend items to a given user, the system uses the user's profile, which represents his preferences. To build such a profile, the system has to collect information about the user, either directly (e.g. by using a form) or indirectly (by analyzing traces) [9, 10].

Recommender systems generally fall into three categories, based on the information they use to perform recommendations: content-based systems [15], that use the semantic content of data, knowledge-based systems [4], that use knowledge about the active user (for example demographic information [16]) and pre-established heuristics, and collaborative-filtering systems [11], that analyze user opinions about the items they have consulted. Hybrid recommendation techniques have also been proposed to take advantage of several of the previous approaches [16, 3]. The popularity of the collaborative filtering approach has increased over the last few years.

In this paper we are interested in the collaborative filtering approach. In this framework, the *preferences* of users U about items I are known, a preference being broadly defined as "an ordering relation between two or more items to characterize which, among a set of possible choices, is the one that best fits user tastes" [1]. However, these preferences are partially known, as they are the preferences the users have given to the system; some preferences are thus missing. Collaborative filtering aims at guessing these missing preferences. In collaborative filtering, the most common representation of preferences is under the form of *utilities*, i.e. quantitative votes (/ratings) provided by users about the items. The recommender estimates the votes of the users on the items they have not seen. We assume in this paper that using votes has several drawbacks such as impreciseness, lack of robustness. This paper is a first attempt to cope with these limitations.

We propose to replace utilities by their qualitative counterpart: *preference relations*. Instead of expressing a quantitative interest about resources (utilities), the users express qualitatively their interest about resources. In this preliminary study, we mainly focus on the decrease in quality of the recommendations when exploiting preference relations, due to the loss of the quantitative aspect. This paper presents not only a theoretical but also a first experimental comparative study of the classical and the preference relations based approaches.

Section 2 presents the classical collaborative filtering approach — based on utilities — through its three main steps. The following section then discusses the pros and cons of this approach and introduces preference relations. Then, Section 4 presents our propositions and adaptations to exploit preference relations in a collaborative filtering-based recommender system. The next section

[4] http://www.amazon.com
[5] http://last.fm

is an experimental validation of the proposed approach. Last, we conclude and present perspectives.

2 Classical Collaborative Filtering

Collaborative filtering uses the preferences of a given user u and the preferences of other users to estimate the unknown preferences of u and thus recommend resources he/she does not know [2]. These preferences are classically expressed under the form of votes.

These votes can be viewed as “quantitative” preferences [14] and can be represented by a *utility* function $ut : I \to \mathbb{R}$. The resource j is preferred to the resource i with respect to the utility function ut (noted $i \preceq_{ut} j$) iff $ut(i) \leq ut(j)$.

Figure 1-a shows two users' (u_1 and u_2) utility functions on eight items $\{a, b, c, d, e, f, g, h\}$.

i	u_1	u_2
a	5	5
b	1	2
c	3	4
d	2	3
e	5	5
f	3	4
g	3	4
h	1	2

a- Two utility functions

i	u_1	u_2
a	–	–
b	1	–
c	3	4
d	–	3
e	–	5
f	3	4
g	3	–
h	1	2

b- Two utility profiles

Fig. 1. Two utility functions and possible corresponding profiles

In collaborative filtering, the known part of a user's preferences — which is generally small — is called the user *profile*. Figure 1-b shows two utility profiles corresponding to the utility functions from Figure 1-a. Dashes indicate missing pieces of information.

To estimate the unknown preferences of a user, two approaches can be used:

- memory-based approach, also called non parametric approach. It exploits similarities (of preferences) between users [18]. The input of the system is not pre-processed, the model is made up of the data as it is. This technique has the advantage to exploit up-to-date information, including the last information about the user acquired by the system. However, this technique is well-known to face scalability problems.
- model-based approach, also called parametric approach. In this approach, the data is pre-processed to build a model [10]. The resulting model not only requires less memory but is also generally less complex in terms of running time. However, as the model is pre-computed, this one is less up-to-date than the one of the memory-based approach.

The memory-based approach is the one we focus on.

The memory-based classical collaborative filtering process is divided into the three following steps:

2.1 Collecting User Profiles

Collecting a user's profile comes down to asking a user to vote for some resources. The votes are usually a positive integer value. A resource not voted by the user can have an implicit null value. These resources can be either proposed by the system (this selection is called *gauge set* [12]) or chosen by the user u, for example during his/her navigation. The resulting information is the user profile.

The system can also automatically estimate the interest of u for the resources u has seen when navigating (according to the time spent on the pages, if the user prints the page, the links followed, etc.). This indirect estimation task is however harder to carry out.

2.2 Computing Similarities Between Users

Once the user profiles are built, the second step computes the similarities $sim(u, u')$ between all pairs of users using the users' profiles. The similarity measures we consider here have the following properties, for all $u, v \in U$:

- $sim(u, v) \in [0; 1]$;
- $sim(u, v) = sim(v, u)$;
- $sim(u, v) = 1$ if and only if u and v have the same preferences on common items;
- $sim(u, v) = 0$ if u and v have no common items or have opposite preferences on common items.

Let u_1 and u_2 be two users with profiles on two subsets $I_1 \subseteq I$ and $I_2 \subseteq I$. Several measures can be used to compute similarities between users [5]. We choose here the well-known cosine measure [19], that is computed by:

$$\cos_{ut}(ut_{u_1}, ut_{u_2}) = \frac{\sum_{i \in I_1 \cap I_2} ut_{u_1}(i) \cdot ut_{u_2}(i)}{\sqrt{\sum_{i \in I_1} ut_{u_1}(i)^2} \cdot \sqrt{\sum_{i \in I_2} ut_{u_2}(i)^2}}. \tag{1}$$

This measure takes into account:

- the proportion of items present in both profiles, and
- the distribution of the votes on the items.

In this case two users u_1 and u_2 will be considered as having identical profiles if and only if they are "co-linear", which means that

- $I_1 = I_2$ and
- there exists a constant $k \in \mathbb{R}$ such that, for all $i \in I_1$, $ut_{u_1}(i) = k \cdot ut_{u_2}(i)$.

2.3 Recommending to a User

To estimate the utility of a resource i for a user u, the classical approach simply computes the weighted average of the utilities of this resource among the neighboring users (those highly similar to u). In this average, the vote of each user u' is weighted by the similarity between u and u':

$$\widetilde{ut}_u(i) = \frac{\sum_{u' \in U_{i,u,k}} sim(u,u') \cdot ut_{u'}(i)}{\sum_{u' \in U_{i,u,k}} sim(u,u')}, \tag{2}$$

where $U_{i,u,k}$ is u's k nearest users that have a known preference for i. In this paper, the similarity measure is instantiated by the cosine measure (see section 2.2).

Once these estimates are computed, the resources that have the highest estimates are recommended to u. We can fix *a priori* either the number of resources to recommend, for example 10 resources, or the minimal utility value for a resource: a resource with a utility value above this threshold will be recommended.

3 From Utilities to Preference Relations

3.1 From Utilities

The possible values for a vote (utility) in classical collaborative filtering are positive integer values, and the scale of possible votes is generally reduced, thus imprecise (the scale of possible values may vary depending on the application [13]). Moreover, a user may prefer one item over an other item, but may have no choice but to give the same vote due to the limited rating scale. For example, let a user who has liked an item and has assigned the maximal value to this item. This user then wants to rate a second item that he has preferred over the first one. This user has no choice but to also give the maximal rating value. In addition, the context, the previously rated resources, etc. may influence the choice of the rating. A user in a bad mood tend to lower ratings compared to users in a good mood.

The resulting votes may thus be imprecise and not reliable, which limits the quality of the computed similarities and therefore the quality of the recommendations.

Let us notice that the objective of a recommender system is to provide a user with a list of items this user does not know yet, the items being ordered according to the expected preferences of the user. In other words, the objective is to complete the top of the user's preference relation. Thus there is no explicit need to estimate quantitative information as provided by classical approaches through the estimated utilities.

3.2 To Preference Relations

Based on the drawbacks resulting from the use of votes to express user preferences, we propose here to replace utility functions by *preference relations*. In this case, the user is not asked to vote for resources but to express a qualitative interest about the resources he/she has already seen. For example, the user will say "I prefer resource j to resource i" rather than "I like this item and I give it a 4".

We put forward the idea that a preference relation can be more appropriate than votes:

- First, in a preference relation the discretization problem is avoided. Indeed, choosing a vote value for a resource among a reduced set of integer values is a delicate task; in the contrary comparing two resources is a more robust task. We can ask here about the case of items not directly comparable. We suppose here that this case never happens. First, we focus on systems that deal with items belonging to the same category (e.g. movies, music, books, etc.), this comparability problem is thus mainly avoided. Second, this question can also be asked when assigning utilities about items that do not belong to the same category. In that case, the two rating scales are different, but are considered as if they were identical.
- Second, [7] shows that making preference judgements is faster than absolute judgements (ratings). We can thus hope that using preference relations will lead to a higher participation rate of the users.
- Third, this approach will allow to consider two users as similar when they order resources in the same way even if they do not rate these resources identically or co-linearly.

However, one of the drawbacks of using preference relations is the polynomial increase in the number of comparisons needed in a test collection [6]: placing a resource here requires comparing it to numerous other resources, whereas rating one resource was enough. Moreover, even if recommandation only asks for a qualitative result, it is presumably better to use utility functions as input — rather than preference relations — as it theoretically contains more information, that is, assuming the ratings are reliable. Nevertheless, as previously mentioned, it is more difficult for a user to provide ratings than comparisons. Moreover, these ratings may be unreliable. In a view to make the user task easier, it thus seems pertinent to investigate the use of preference relations for recommender systems.

3.3 Preference Relations in Details

Let us now formally define preference relations before introducing their use for collaborative filtering.

A *preference relation* is a binary relation $i \preceq j$ on I that is:

- reflexive: $\forall i \in I,\ i \preceq i$;
- transitive: $\forall i, j, k \in I,\ (i \preceq j) \wedge (j \preceq k) \Rightarrow (i \preceq k)$;
- total/complete: $\forall i, j \in I,\ (i \preceq j) \vee (j \preceq i)$.

We can notice that, with this definition:

- "j is strictly preferred to i" is written $(i \preceq j) \wedge \neg(j \preceq i)$ and is noted $i \prec j$.
- "i and j are equivalent" or "the user does not mind between i and j" is written $(i \preceq j) \wedge (j \preceq i)$ and is noted $i \simeq j$.

Given two elements i, j there will be three possibilities: $i \prec j$, $j \prec i$, $i \simeq j$ and exactly one of them has to be true (the relation being total).

It is important to notice that all the resource pairs (i, j) are comparable. There is no non-determinism. If all the resource pairs are supposed to be comparable by a given user, this does not mean that the user knows these resources and knows which one he/she prefers. In other terms, even though the relation is total, the user (and the system) only has an incomplete view of it: the user profile. However, this means that, if the user knows all the resources, he/she knows which one is preferred.

Given all these properties, a preference relation can be represented in a simplified way (we do not consider the relations that can be deduced by transitivity) under the form of a strictly ordered chain of equivalence classes.

When dealing with a user *profile*, an unknown information will be noted $(i \,?_u\, j)$ in the case of a "preference relation" profile, and $ut_u(i) =?$ in the case of a "utility" profile.

Figure 2-a shows a preference relation. Figure 2-b shows a profile compatible with this preference relation. Let us notice that, in Figure 2-b, (1) the resource g is not comparable neither to a, nor to e, nor to c, and (2) there is no information regarding f and h.

$$\begin{tikzcd}
(a \simeq e) \arrow[d] \\
(c \simeq f \simeq g) \arrow[d] \\
(d) \arrow[d] \\
(b \simeq h)
\end{tikzcd}$$

a- A preference relation

$$\begin{tikzcd}
(a \simeq e) \arrow[d] & & \\
(c) \arrow[dr] & & (g) \arrow[dl] \\
& (d) \arrow[dl] & \\
(b) & &
\end{tikzcd}$$

b- A compatible profile

Fig. 2. A preference relation and one possible corresponding profile (an arrow means "strictly preferred to")

Given a user $u \in U$, his/her preferences will be noted $i \preceq_u j$ if a preference relation is used, and $ut_u(i)$ if a utility function is used. We can notice here that a utility function corresponds to a unique preference relation. In the contrary, a preference relation $\preceq$ can correspond to several utility functions. Let us suppose for example that ut "matches" $\preceq$ and that $f : \mathbb{R} \to \mathbb{R}$ is strictly monotonic, then $f(ut)$ also matches $\preceq$. So, two users can have different utility functions but can agree with each other about the resources they prefer. They will thus have similar preference relations. This is an advantage of the representation with preference relations. An example of such a case is presented in Figure 2-a where the preference relation corresponds to the two utility functions from Figure 1-a.

Some works on recommender systems have exploited preference relations, as in content-based recommender systems for example. Preference relations have been used to compute unknown preferences (by using the properties of preference relations) when content-based approaches could not compute them [17]. In the frame of collaborative filtering [8] is interested in combining a set of preference functions (as ratings). This combination results in an ordered set of objects: no utility value is deduced from this combination step; the result is a preference relation.

4 Collaborative Filtering using Preference Relations

We focus now on how to exploit preference relations in a collaborative filtering approach. We choose to exploit the same three steps as in classical collaborative filtering and to adapt them to exploit preference relations. We now present these three adapted steps:

4.1 Collecting User Profiles

When preference relations are employed, a user's profile can also be collected through question answering. In this case, the system presents resource pairs (i, j) and then asks which resource the user prefers. There are four possible answers:

- $i \prec j$ item j is preferred over i
- $j \prec i$ item i is preferred over j
- $i \simeq j$ the user does not mind between the two items
- $i \,?\, j$ the user does not know

Collecting the profile of a user as a preference relation is not restricted to a predefined number of equivalence classes. A user's preferences are thus represented more precisely than with most rating scales.

One problem is the number of questions required to get a full profile. If the profile contains n equivalence classes, a dichotomic procedure will require, in the worst case, $\lceil \log_2(n) \rceil$ questions to add a new resource. Obviously, so many questions cannot be asked to the users and one may decide to ask a subset of questions. The resulting profile will be partial but may be sufficient. Choosing the appropriate questions to ask — each new question depending on the preceding answers — is a challenging problem.

4.2 Computing Similarities Between Users

Let $\mathcal{I}_u$ be the set of resource pairs (i, j) present in the profile of user u. Let us also define the function $f_{u_1,u_2}(i, j)$ indicating whether two users u_1 and u_2 agree about their preference on the two resources i and j. Given an item pair (i, j), the value of this function is 1 if the two users u_1 and u_2 have the same preference about i and j (the same order) and 0 otherwise. We thus define a cosine similarity measure adapted to preference relations, this similarity measure is presented in Equation 3.

$$\begin{aligned}
&\cos_{\preceq}(u_1, u_2) \\
&= \frac{\sum_{(i,j)\in\mathcal{I}_1\cap\mathcal{I}_2} f_{u_1,u_2}(i,j)}{\sqrt{\sum_{(i,j)\in\mathcal{I}_1} f_{u_1,u_1}(i,j)} \cdot \sqrt{\sum_{(i,j)\in\mathcal{I}_2} f_{u_2,u_2}(i,j)}} \\
&= \frac{\sum_{(i,j)\in\mathcal{I}_1\cap\mathcal{I}_2} f_{u_1,u_2}(i,j)}{\sqrt{|\mathcal{I}_1|.|\mathcal{I}_2|}}.
\end{aligned} \tag{3}$$

In this Equation, the numerator represents the number of pairs of resources where both users u_1 and u_2 agree about their preference. The denominator is the normalization factor.

As for utilities, this definition accounts for:

- the proportion of resource pairs present in both profiles, and
- the distribution of the preferences on the resource pairs.

4.3 Recommending to a User

Recommending resources when using preference relations will rely on two substeps: first completing the preference relation of u, second recommending him some resources. The completion of the preference relation is a difficult step.

To complete a preference relation, we propose an approach that estimates the position of a resource in the preference relation of a user u. As in the classical collaborative filtering approach, similar users (neighbors) u' are used, the completion is thus collaborative. We first compute the position of an item i in each neighbor's profile. Second we estimate the position of i in the profile of u as the weighted average of the positions of i in the profiles of users u'.

The profile of a user u' being generally partial, we propose to measure the position of a resource i in the profile of u' by counting:

$\#^{\oplus}_{u',i}$ the number of resources that are strictly preferred to i;
$\#^{\simeq}_{u',i}$ the number of resources (other than i) that are equally preferred to i;
$\#^{\ominus}_{u',i}$ the number of resources that are strictly less preferred than i.

Let us notice that, because a profile is typically incomplete, some resources are not comparable to i, and thus are not integrated into the formulas.

The position of i in the profile of u' is thus:

$$\widehat{ut}_{u'}(i) = \frac{-\#^{\oplus}_{u',i} + \#^{\ominus}_{u',i}}{\#^{\oplus}_{u',i} + \#^{\simeq}_{u',i} + \#^{\ominus}_{u',i}}.$$

This formula has several interesting features:

- $\widehat{ut}_{u'}(i) \in [-1; +1]$;

- $\widehat{ut}_{u'}(i) = 0 \Leftrightarrow \#^{\oplus}_{u',i} = \#^{\ominus}_{u',i}$;
- $\widehat{ut}_{u'}(i) = +1 \Leftrightarrow i$ is the preferred resource of u';
- $\widehat{ut}_{u'}(i) = -1 \Leftrightarrow i$ is the less preferred resource of u'.

We can now use the classical prediction formula (from section 2.3) by exploiting the position of the resource i in the profile of neighbor users u' ($\widehat{ut}_{u'}(i)$), instead of their vote on the resource i ($ut_{u'}(i)$):

$$\widetilde{ut}_u(i) = \frac{\sum_{u' \in \hat{U}_{i,u,k}} sim(u,u') \cdot \widehat{ut}_{u'}(i)}{\sum_{u' \in \hat{U}_{i,u,k}} sim(u,u')},$$

where $\hat{U}_{i,u,k}$ is the set of users u' in U:

- that have i in their profile, and
- that belong to the set of u's k nearest neighbors.

Let us notice that the higher the position of a resource, the better its recommendation to the user. Once the positions of resources i have been computed, the resources with the highest estimated positions are recommended to the user by following the same principle as in Section 2.3.

5 Experiments

5.1 Experimental Data

Ideally, an experimental comparison between both approaches — "preference relation" and "utility" — should compare both recommendation processes on the whole process: collecting user profiles, computing similarity between users and recommending resources. However, no dataset exists that contains user preferences both under the form of preference relations and utilities. Building such a dataset is a tedious task that we have not yet carried out. We can thus not directly compare both approaches.

The classical datasets used in collaborative filtering contain utilities (ratings). We thus decide to use such a dataset to compute recommendations by using utilities and preference relations. To get the users' preference relations, we transform the utilities of the dataset under the form of preference relations. For each pair of resources i and j rated by a user u, three cases may be encountered:

- the rating of i is lower than the rating of j. In the resulting preference relation, j will be represented as being preferred over i.
- the rating of i is greater than the rating of j. In the resulting preference relation, i will be represented as being preferred over j.
- the rating of i is equal to the rating of j. In the resulting preference relation, i will be represented in the same equivalence class than j.

Let us notice that when a resource has not been rated, it is not represented in the preference relation neither.

In that case, we are in the worst experimental conditions to quantify the contribution of our approach: not only imprecise data is used, as they come from utilities; but also part of the information is lost when turning them into preference relations: quantitative information has been transformed to qualitative information. Conducting experiments in these conditions, we do not expect to increase the quality of the recommendations. However, as presented in the introduction, we only aim at evaluating the loss in performance when exploiting only qualitative information to determine if this new approach is promising.

We choose to work on the well-known Movielens[6] state of the art corpus. This dataset is made up of a set of user preferences about movies. These preferences are utilities (votes) that are integer values between 1 (dislike) and 5 (like). The dataset contains 1682 users, 943 items and 100k preferences. The dataset is divided into 2 parts, 80% of the data is used to train the recommender system (the training set) and the 20% left are used to evaluate the approach (the test set). After the data is converted into preference relations, we can notice that each resulting preference relation contains at most 5 equivalence classes.

5.2 Similarity Between Users

This section is dedicated to a first evaluation of the interest of exploiting preference relations instead of utilities. The figures about the use of utility functions presented hereafter are considered as state of the art figures and are compared to those related to the use of preference relations.

In collaborative filtering, the recommender classically estimates missing utilities. These estimates are then compared to the utilities (votes) in the test set. The system accuracy is thus usually evaluated in terms of MAE (Mean Absolute Error). The lower the MAE value, the better the accuracy.

In this section, preference relations are used only in the first and second step: the profile collection and the computation of the similarity between users. We will thus evaluate the quality of the similarity measure when using preference relations. The classical recommendation step — based on ratings — is used (Equation 2). It is executed by instanciating $sim(u, u')$ either (1) by $\cos_{ut}(u, u')$ (the similarity on utilities) or (2) by $\cos_{\preceq}(u, u')$ (the similarity on preference relations).

The evaluation can be made in terms of MAE as the votes of the test data are available. The impact of the preference relations on the similarity measure can be quantified. The accuracies are presented in Table 1.

As expected, the mean error is higher when using preference relations compared to using utilities. However, this increase is lower than 3%. It is thus not significant. As stated earlier, an increase was predictable as the preference relation profiles are derived from utility profiles by removing (quantitative) infor-

[6] http://movielens.org

Table 1. MAE according to the approach used to compute the similarity value

Approach	MAE
Utility functions	0.71
Preference relations	0.73

mation. The experimental conditions are thus biased. The small loss in accuracy shows the potential of this new approach.

5.3 Recommending to a User

When using preference relations, no vote should be used, so the recommender cannot compute or estimate missing votes. The accuracy of the complete system cannot be evaluated in terms of MAE.

Thus we propose in this section to evaluate the accuracy of the system by using the precision measure. Precision compares the list of the preferred items of each user (from the test data) to the ordered list of preferred items computed by the recommender. Concretely, it computes the ratio between the number of resources the system judged as being preferred divided by the number of resources actually preferred by the user.

In this section we exploit preference relations in the whole recommendation process: when collecting users' profiles, computing similarity measures and computing recommendations so as to evaluate the accuracy of our approach.

We thus measure, among the user's preferred resources, the number of resources that are evaluated as preferred by the recommender. This measure is computed twice: when using utilities and when using preference relations.

The items we consider here to be preferred by a user are the highly rated ones. On the MovieLens corpus, those preferred resources are rated 4 or 5. We thus evaluate the precision of our approach on two sets of resources: those rated 5 and those rated 4 or 5. The corresponding precisions are presented in Table 2.

Table 2. Precision of the two approaches

Approach	rated 5	rated 4 or 5
Utility functions	0.52	0.75
Preference relations	0.51	0.77

From Table 2 we can notice that, on these two sets of resources, both approaches have similar precisions (the differences are not significant). On the one hand, the utility approach has a slightly greated precision on the set of resources rated 5. On the other hand, the preference relation approach is slightly better on the set of resources rated 4 or 5. No significant loss is observed with this new approach.

We can conclude that exploiting preference relations in the whole process does surprisingly not lead to a decrease in the quality of predictions, although a loss in information has been obtained when transforming rating data in preference relations.

To refine our experimentations, we also evaluate the preference relations approach in terms of ranks. As we are only interested in the ranking quality about the resources preferred by users, it is not pertinent to use rank correlation measures such as Kendall Tau. Moreover, in our experiments the resources preferred by users are equally preferred (they all have a vote equal to 5), no difference in terms of preference can be made between them, thus classical ranking tests cannot be used. Thus we propose to evaluate the quality of the ranking in terms of mean rank. This evaluation will quantify the quality of the recommendation on the whole set of resources that the users preferred. Let us recall that our approach is a "position-based" approach. The evaluation in terms of mean rank is also an evaluation in terms of position, this measure is thus adapted to our approach. We have computed the mean rank on the set of resources rated 5 by the users in the test set. The study of the mean rank shows that, when using preference relations, it is 9% lower than with the utility-based approach. This improvement is significant. The items are thus better ordered. We can thus conclude that, in term of mean rank, the recommendations computed when using preference relations are more accurate than the ones from the utility-based approach.

6 Conclusion and Future Work

This paper presents a new approach to represent preferences in a collaborative filtering-based recommender system. We propose to use preference relations instead of the classically used ratings (utilities). We have first recalled the utility-based approach, before discussing why switching to preference relations could be a good idea. We have then proposed an adaptation of the classical collaborative filtering in order to exploit preference relations. Preference relations are qualitative preferences whereas utilities are quantitative, thus more informative. Thus, in this paper we focused on the resulting loss in quality of recommendations, due to the qualitative preferences. This approach has been evaluated on a state of the art corpus that has been transformed under the form of preference relations. It compares with the classical approach (based on utilities) and even improves significantly performance in terms of mean rank. Exploiting preference relations to acquire users' preference is thus a highly promising approach that we will further investigate.

In a future work, we will first test the hypothesis about robustness and stability of the preference relations compared to ratings. This test has started, we are collecting user preferences and votes in various contexts. Second, we will study the robustness of this new approach by removing some pieces of information in the preference relations (Figures 1-b and Figure 2-b) in order to measure the evolution of the accuracy and to find the minimal quantity of information required to perform accurate recommendations.

We will also implement a complete recommendation system, that includes the tedious task of collecting the users' preferences under the form of preference relations. The resulting corpus will allow us to evaluate the actual performance of our approach.

References

1. Brafman, R., Domshlak, C.: Preference handling - an introductory tutorial. AI Magazine 30(1), 58–86 (2009)
2. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98). pp. 43–52. Morgan Kaufmann, San Francisco, CA (1998)
3. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
4. Burke, R., Hammond, K., Cooper, E.: Knowledge-based navigation of complex information spaces. In: Proceedings of the 13th National Conference on Artificial Intelligence (AAAI'96). pp. 462–468. Menlo Park, Canada (1996)
5. Candillier, L., Meyer, F., Boullé, M.: Comparing state-of-the-art collaborative filtering systems. In: Proceedings of 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLMD'07. pp. 548–562 (2007)
6. Carterette, B., Bennett, P.: Evaluation measures for preference judgments. In: Proceedings of the Annual ACM SIGIR Conference. pp. 685–686 (2008)
7. Carterette, B., Bennett, P., Chickering, D., Dumais, S.: Here or there; preference judgments for relevance. In: Proceedings of the Annual European Conference on Information Retrieval (ECIR). pp. 16–27 (2008)
8. Freund, Y., Iyer, R., Schapire, R., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research 4, 933–969 (2003)
9. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: The Adaptive Web: Methods and Strategies of Web Personalization, chap. User Profiles for Personalized Information Access, pp. 54–89. Springer-Verlag Berlin Heidelberg (2007)
10. de Gemmis, M., Iaquinta, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Preference learning in recommender systems. In: ECML/PKDD-09 Workshop on Preference Learning (PL 09) (2009)
11. Goldberg, D., Nichols, D., Oki, B., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM 35(12), 61–70 (1992)
12. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval Journal 4(2), 133–151 (2001)
13. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22(1), 5–53 (2004)
14. Mukherjee, R., Sajja, N., Sen, S.: A movie recommendation system - an application of voting theory in user modeling. User Modeling and User-Adapted Interaction 13, 5–33 (2003)
15. Pazzani, M., Billsus, D.: The Adaptive Web, chap. Content-Based Recommendation Systems, pp. 325–341. Springer Berlin / Heidelberg (2007)

16. Pazzani, M.J.: A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review 13(5-6), 393–408 (1999), citeseer.ist.psu.edu/pazzani99framework.html
17. Perez, L., Barranco, M., Martinez, L.: Building user profiles for recommender systems from incomplete preference relations. In: Proc. of the Fuzzy Systems Conference (2007)
18. Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., Riedl, J.: GroupLens: An open architecture for collaborative filtering of netnews. In: Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work. pp. 175–186. ACM, Chapel Hill, North Carolina (1994), citeseer.ist.psu.edu/resnick94grouplens.html
19. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating "word of mouth". In: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'95). vol. 1, pp. 210–217 (1995)