Towards recommender systems based on a fuzzy preference aggregation

Samia Boulkrinat[1] **Allel Hadjali**[2] **Aicha Mokhtari**[3]

[1]RIIMA/CSD,USTHB Algiers, Algeria, sboulkrinat@usthb.dz
[2]LIAS/ENSMA Poitiers, France, allel.hadjali@ensma.fr
[3]RIIMA/CSD,USTHB Algiers, Algeria, aissani_mokhtari@yahoo.fr

Abstract

An approach to deal with user preference relations, instead of absolute ratings, in recommender systems is discussed. User's preferences are then ratings expressed qualitatively by using linguistic terms. This is a suitable technique when preferences are imprecise and vague. On the other hand, to avoid that the overall item rating may hide the users' preferences heterogeneity and mislead the system when predicting the items that users are interested in, multi-criteria ratings are used as well. User's items ratings are represented through a preference graph which highlight better items relationships. Similarity between users is performed on the basis of the similarity of their preference relations which can better capture similar users' ratings patterns. Some preliminary results shows that, our approach enhances the classical recommender system precision thanks to the graphs used for prediction which are more informative and reflect user's initial ratings relations in a better way.

Keywords: Recommender system, fuzzy preferences, preference graph, fuzzy sets.

1. Introduction and motivation

One of the main aims of recommender systems is to predict the user ratings for items that (s)he didn't rated yet and may be interested in. Two aspects are of interest when performing prediction: (i) the ratings of items the user has already evaluated if it's a content-based system or, (ii) the similar users'ratings while working with a collaborative filtering [5][16]. The later, is one of the most frequently used techniques in recommender systems; it's based on the assumption that similar users with similar opinions may rate items similarly and so can help each other to find out useful information [17]. Furthermore, incorporating multi-criteria ratings is a promising technique to improve the current recommender systems accuracy, since the systems can better understand user preferences on their multiple facets, unlike the overall rating that may hide the users' preferences heterogeneity and mislead the system when predicting items that users are interested in [1][13]. In fact, a user u is assumed to be similar to the user u_1 regarding their overall items ratings, but it turns out the user u is likely to be more similar to user u_2 if we take into account their criteria's items ratings since they show similar multi-criteria rating patterns [2]. On the other hand, there are situations where the information cannot be assessed precisely in a quantitative way but may be suitable in a qualitative one. Especially, when attempting to qualify phenomena related to human perception, we are often led to use words in natural language instead of numerical values [18]. That's why, rating an item quantitatively like his/her first *laptop* with several technical features, may be a difficult task for a user (D_1). However, users with similar tastes may rate the same items differently, because rating task is subjective and depends on individual habits. Furthermore, the ratings are not completely trustworthy since they depend on reduced rating scale and the rating value given by users may be influenced by many factors such as his/her character, experience, context, etc.[2]. So, considering preference relations instead of absolute item's rating can better reflect the similarity between users since they have similar relation items rating patterns [2]. An item a is more preferred than an item b, if the rating of a is greater than the rating of b. In [6], the items rating are represented through a preference graph where nodes denote the rated items and the edges denote the preference relations between items. An edge is then weighted with 0.5 if two items are equally preferred and an edge directed towards the most preferred item with a value of 1, if one item is more preferred than another. This weighting approach is limited since it does not measure how much an item is more preferred than another. Moreover, the predicting process is based on the edges values which don't reflect the initial user's ratings. One way to recover the lost information about ratings is to deploy machine learning techniques [6] [9]. However, despite their relevance, such techniques are unable to recover exactly the initial ratings and so, the quality of the prediction is not really precise (D_2). In this paper, we discuss two main contributions in order to overcome the above major drawbacks (D_1) and (D_2). Such contributions are summarized as follows:

- We use a qualitative rating, to express imprecise and vague user preferences such as:

very low, *medium*, and *high*. This can be achieved by using linguistic terms represented by means of fuzzy sets [3].

- We suggest attributing a gradual weighting (i.e., a value in [0, 1] instead of a value in {0, 1}) to edges in order to better quantify the user's preferences scale for either an item or a criterion. User preferences are then gradually weighted and represented through items and criteria graphs which better highlight their inherent relationships. Such gradual weights are then carried throughout the prediction process, without losing any information about his/her initial ratings. In so doing, one guarantees both faithfulness and precision of the prediction.

The remaining of the paper is organized as follows. In Section 2, we present our preference modeling approach. The aggregation process is outlined in Section 3. The ultimate prediction step is described in Section 4. In Section 5, we present a preliminary evaluation of our approach. We conclude and discuss some future works in Section 6.

2. Preference modeling

2.1. Fuzzy sets

Fuzzy sets [8] were introduced to model object classes with unclear borders. Formally, a fuzzy set F on a referential U is characterize by a membership function $\mu_F : U \to [0, 1]$ where $\mu_F(u)$ is the degree of membership of u to F. There are different kind of membership functions, such as triangular, trapezoidal, etc. For instance, a triangular fuzzy number [3] is particular fuzzy set, whose membership function is represented by three parameters t_1, t_2 and t_3 (see Formula (1) and Figure 1):

$$f(x; t_1, t_2, t_3) = \begin{cases} 0 & \text{for } x < t_1 \\ \frac{x-t_1}{t_2-t_1} & \text{for } t_1 \le x \le t_2 \\ \frac{t_3-x}{t_3-t_2} & \text{for } t_2 \le x \le t_3 \\ 0 & \text{for } x > t_3 \end{cases} \tag{1}$$

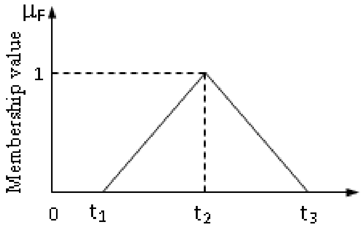

Figure 1: Membership function of triangular fuzzy number $F = (t_1, t_2, t_3)$

2.2. Preference statements

In this study, the user's preferences are expressed through linguistic terms such as *Very high*, *Medium high*, *High*, *Medium*, *Medium low*, *Low*, *Very low* [11][20]. Each preference level is associated with a Triangular Fuzzy Number (TFN), as described in Table 1 and Figure 2:

Preference level	Fuzzy number
Very low (VL)	(0,1,2)
Low (L)	(1,2,3)
Medium low (ML)	(2,3,4)
Medium (M)	(3,4,5)
Medium high (MH)	(4,5,6)
High (H)	(5,6,7)
Very high (VH)	(6,7,8)

Table 1: Linguistic terms and their semantics

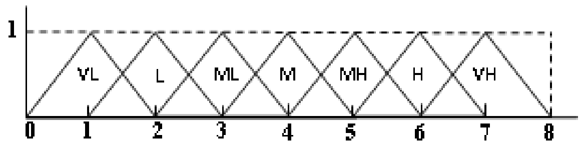

Figure 2: Fuzzy triangular numbers

In recommender systems' framework, a user preference for *ibis hotel* item for instance, may be expressed on the basis of the following criteria: *Clean*, *Comfort*, *Location*, *Services*, *Staff* and *Value of money*. Each criterion preference, is associated with a TFN, among the seven commensurable values (see the preference levels of Table 1).

2.3. Preference comparison

There are several linguistic tools to deal with preferences expressed linguistically [12] and other methods that manage numerically this kind of preferences, through their fuzzy numbers' ranking as mentioned by Chen and Hwang [4]. In this regard, they identified four major classes of ranking fuzzy numbers methods, among them, the area measurement, that we've used here. Thereby, to compare two preferences, for instance *Very low* and *Low*, we use the surface α which separate their corresponding TFNs, namely $\tilde{T_a} = (0, 1, 2)$ and $\tilde{T_b} = (1, 2, 3)$. The larger this surface is the more distinguishable the two preferences will be, from the less to the more preferred one. That's why this separating surface α, can quantify how much a preference level is better than an other. The magnitude of this surface area depends on the location of each fuzzy number on the real line. There are several methods to measure a surface area, such as surface integrals [15] that can compute any kind of surface. Now, to compute the area α, we've to find all points (x, y) so that x is bounded by the both lines $f_1(y)$ and $f_2(y)$ and y by the line ν_1, which is equal to the ordinate of the intersection point I of lines $f_1(y)$, $f_2(y)$ and $\nu_2 = 1$, i.e.

$$\alpha = \{(x, y) \in \Re^2 / f_1(y) < x < f_2(y), \nu_1 < y < \nu_2\} \tag{2}$$

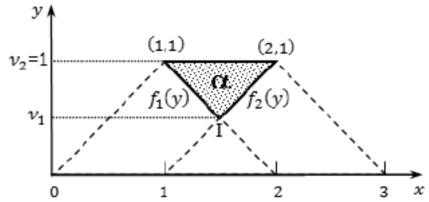

Figure 3: Surface between two preferences: Very low and Low.

The surface α will be equal to double integral between x and y axis; where the y's values extend from ν_1 up to ν_2 while the x's values varied from $f_1(y)$ up to $f_2(y)$.

$$S(\alpha;\tilde{T}_a,\tilde{T}_b)=\int\int_{\alpha} dxdy=\int_{\nu_1}^{\nu_2}\int_{f_1(y)}^{f_2(y)} dxdy \qquad (3)$$

One can check that the lines $f_1(y){=}{-y}+2$ and $f_2(y)=y+1$, whose intersection point is I with coordinates $(\frac{3}{2},\frac{1}{2})$, hence the line $y{=}\nu_1{=}\frac{1}{2}$. Then, the surface will be equal to :

$$S(\alpha;\tilde{T}_a,\tilde{T}_b)=\int\int_{\alpha} dxdy=\int_{\frac{1}{2}}^{1}\int_{-y+2}^{y+1} dxdy=0.25 \qquad (4)$$

Possible surfaces α separating two TFNs $\tilde{T}_a$ and $\tilde{T}_b$ are *0* when the two TFNs are equal or *0.25*, *1*, *2*, *3*, *4* or *5* as illustrated in Figure 4.

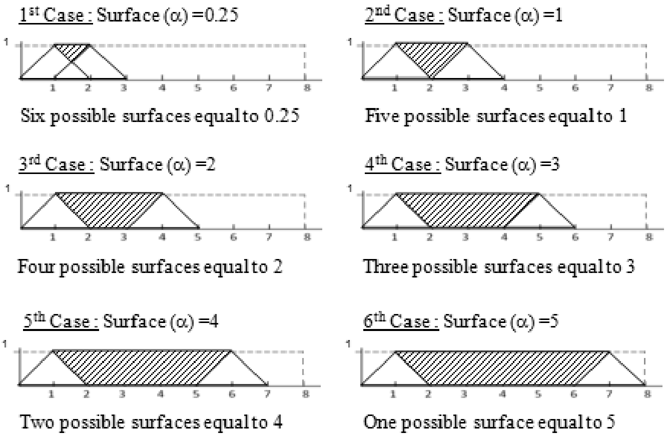
1st Case : Surface (α) =0.25
Six possible surfaces equal to 0.25
2nd Case : Surface (α) =1
Five possible surfaces equal to 1
3rd Case : Surface (α) =2
Four possible surfaces equal to 2
4th Case : Surface (α) =3
Three possible surfaces equal to 3
5th Case : Surface (α) =4
Two possible surfaces equal to 4
6th Case : Surface (α) =5
One possible surface equal to 5
1
1 2 3 4 5 6 7 8

Figure 4: Possible surfaces between two preferences.

Consider a fuzzy partition (in the sense of Ruspini) with n TFNs. The possible separating surface α between each two TFNs $\tilde{T}_a$ and $\tilde{T}_b$ according to our scale values is then:

$$S(\alpha;\tilde{T}_a,\tilde{T}_b)=\begin{cases} 0 & \text{si } \tilde{T}_a=\tilde{T}_b \\ 0.25 & \text{si } \tilde{T}_a\cap\tilde{T}_b\neq\emptyset \\ 1,2,3,...,or(n-2) & \text{si } \tilde{T}_a\cap\tilde{T}_b=\emptyset \end{cases} \qquad (5)$$

2.4. Measuring preferences average

2.4.1. Weighting of preference levels

The criteria may have different importance for a user, depending on his/her needs and therefore should not be handled identically. For instance, a businessman looking for a *hotel* may prefer *comfort* and *location* instead of *value of money*, *cleanliness* and *service* better than the *staff.* In this context, weighting the preferences allow to assign more importance to the most predominant criteria. To this end, for each TFN representing a level of preference among the seven ones, we assign a weight w_i depending on its position i on the real line. In our approach, and for the sake of illustration, we assign the weight 1 to the seventh level and the weights w_i of the remaining levels are summarized in the table 2 (in practice, the weights w_i can be provided by the user):

Order i	Fuzzy number	Weight w_i
1	(0,1,2)	0,14
2	(1,2,3)	0,29
3	(2,3,4)	0,43
4	(3,4,5)	0,57
5	(4,5,6)	0,71
6	(5,6,7)	0,86
7	(6,7,8)	1,00

Table 2: Weights distribution

2.4.2. Evaluating the overall preference rating

Given n preferences represented by TFNs $\tilde{T}_i=(t_1^{(i)},t_2^{(i)},t_3^{(i)})$ where $i=1,...,n$, we can estimate the overall preference by means of the fuzzy weighted arithmetic average of these n numbers, considering their respective weights w_i, as follows:

$$\begin{aligned}\tilde{T} &= \frac{\sum_{i=1}^{n} w_i\times\tilde{T}_i}{\sum_{i=1}^{n} w_i} \\ &= \frac{\left(\left(\sum_{i=1}^{n} w_i\times t_1^{(i)}\right),\left(\sum_{i=1}^{n} w_{(i)}\times t_2^{(i)}\right),\left(\sum_{i=1}^{n} w_i\times t_3^{(i)}\right)\right)}{\sum_{i=1}^{n} w_i} \\ &= (t_1,t_2,t_3)\end{aligned} \qquad (6)$$

For example, assume that the preference levels for the six criteria of an item *ibis hotel*, represented by TFNs $\tilde{T}_i{=}(t_1^{(i)},t_2^{(i)},t_3^{(i)})$, for a user u, are given in Table 3. Then, the fuzzy weighted arithmetic average $\tilde{T}=(t_1,t_2,t_3)$ expressing the overall preference level for *ibis hotel*, is assessed as follows:

Criterion(c_i)	Preference level(i)	Fuzzy number	Coefficient (w_i)
Clean	Medium	(3,4,5)	0.57
Comfort	Medium high	(4,5,6)	0.71
Location	High	(5,6,7)	0.86
Services	Medium low	(2,3,4)	0.43
Staff	Very low	(0,1,2)	0.14
Value of money	Very low	(0,1,2)	0,14

Table 3: Criteria preferences for *ibis hotel*

$$t_1{=}\frac{(0.57\times3)+(0.71\times4)+(0.86\times5)+(0.43\times2)+(0.14\times0)+(0.14\times0)}{0.57+0.71+0.86+0.43+0.14+0.14}{\approx}3$$

$$t_2{=}\frac{(0.57\times4)+(0.71\times5)+(0.86\times6)+(0.43\times3)+(0.14\times1)+(0.14\times1)}{0.57+0.71+0.86+0.43+0.14+0.14}{\approx}4$$

$$t_3{=}\frac{(0.57\times5)+(0.71\times6)+(0.86\times7)+(0.43\times4)+(0.14\times2)+(0.14\times2)}{0.57+0.71+0.86+0.43+0.14+0.14}{\approx}5 \qquad (7)$$

Which leads to $\tilde{T}=(t_1,t_2,t_3)=(3,4,5)$, corresponding to *'Medium'* level.

2.5. Preference graph

In order to represent, the preference relationships of a user u, we build out two kind of preference graphs : the criteria graph G_u^t for each item t and the overall items graph G_u.

2.5.1. Building the criteria graph

For each item t, construct its criteria graph $G_u^t = (V_u^t, E_u^t)$, where (V_u^t) is the set of nodes (η_a) describing the criteria (c_a) and (E_u^t) the set of edges, expressing preferential relations between nodes:

1. Add a node (η_a) in the graph for each criterion (c_a) of item t.
2. for each pair nodes (η_a, η_b) of (V_u^t), add an edge (η_a, η_b) with a value α so that α is the area between the TFNs representing the two criteria ratings (c_a) and (c_b). A value α of any edge (η_a, η_b), implies that the edge (η_b, η_a), will have a value $-\alpha$. We assume that, if the criterion (c_a) is α times more preferred that the criterion (c_b), then (c_b) is $-\alpha$ times less preferred than (c_a).
3. Add an edge (η_a, η_b) , with a valuation 0, if the criteria (c_a) and (c_b) are represented by the same fuzzy number rating.

Example 1 : Criteria graph for *ibis hotel*
Consider the user u of table 3. The criteria graph for the *ibishotel* item is represented as follows:

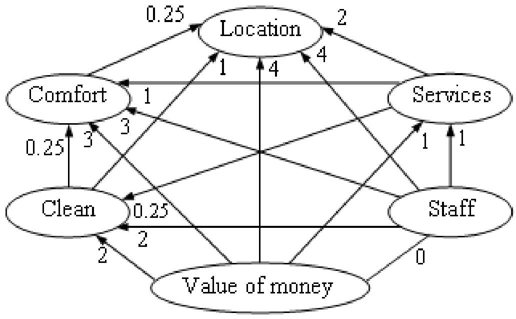

Figure 5: Criteria graph of the user u.

2.5.2. Building the items graph

From each user u's criteria ratings, we derive his/her items ratings which allow us to built out his/her items graph G_u=(V_u, E_u), where (V_u) is the set of nodes (φ_a) describing the items (a) and (E_u) the set of edges, expressing preferential relations between nodes, by proceeding as follows:

1. Add a node (φ_a) in the graph for each item a rated by the user u. An item is rated by a user, if (s)he rated all its criteria (c_a) or some of them. For the remaining unrated criteria, we assume their ratings equal to *0*. The overall item rating a is computed using the fuzzy weighted arithmetic average of its fuzzy criteria ratings, rounded down, for any fraction less than 5 and up otherwise (cf. §2.4.2).
2. For each pair nodes (φ_a, φ_b) of (V_u), add an edge (φ_a, φ_b) with the value α so that α is the surface separating the TFNs representing the two preference levels about items a et b. The edge's value (φ_b, φ_a), will be $-\alpha$.
3. Add an edge (φ_a, φ_b) into the graph, with a valuation 0, if the preferences on a and b are two equal fuzzy numbers.

Example 2 : Item graph
Let's take three hotel items : *ibis*, *hilton* and *sofitel* which are rated by the user u as shown in Table 4. The item graph corresponding to this table is sketched in Figure 6:

Item	Preference level	Fuzzy number
Ibis	Medium	(3,4,5)
Hilton	Medium high	(4,5,6)
Sofitel	Medium low	(1,2,3)

Table 4: Items preferences of u.

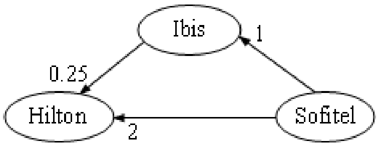

Figure 6: Item graph of user u.

3. Aggregating preferences

3.1. Selecting similar users

There are different methods to compute similarity between each pair of users [14]. One of the most used approaches is the Pearson correlation coefficient. It considers that users are different regarding to how they interpret the rating scale. Some users tend to rate items very highly, whereas others will never give a high rating to any item. The Pearson coefficient takes these averages out in the calculation to make users comparable – that is, although the absolute values of two users' ratings may be quite different, a rather clear linear correlation of their ratings and thus their similarity can be detected. To adapt this coefficient to our context, we make use of the following formula:

$$Sim_{(u,v)} = \frac{\sum_{i\in S_{(u,v)}}(r_{u_i} - \overline{r_u})(r_{v_i} - \overline{r_v})}{\sqrt{\sum_{i\in S_{(u,v)}}(r_{u_i} - \overline{r_u})^2}\sqrt{\sum_{i\in S_{(u,v)}}(r_{v_i} - \overline{r_v})^2}} \quad (8)$$

Where u and v is a pair of users, r_{u_i} is the user u's rating of an item i, $S_{(u,v)}$ is the items set that users u et v have both rated in common and $\overline{r_u}$ and $\overline{r_v}$ are the average of nonzero ratings of the two users. This method has the advantage of supporting also the difference between users ratings' system. Given both criteria and items preference graphs of all users; we choose a target user for whom we want to predict some unrated items. For this purpose, we select the most similar user to him/her. Two users are considered as similar when they order resources the same way, even if they don't rate them identically. The similarity between two users can be performed by a two-step procedure, as follows:

1. For each candidate user u_i, we choose one by one, all the items (s)he rated in common with the target user u. For each item t, we explore its criteria graph, in order to compare it to the criteria graph of the target user. First of all, we compute the criteria Pearson similarity measure $Sim^t_{(u,u_i)}$, which takes into account the values of edges having the same orientation instead of initial criteria ratings. When an edge (η_a, η_b) valued α has a different orientation with the corresponding edge in the other graphs, we take the edge (η_b, η_a) with a value equals to $-\alpha$.
2. The second step, we evaluate the aggregated similarity $Sim_{(u,u_i)}$ of all items rated in common by the similar users and the target user, by using the arithmetic average of all criteria similarities. Then, we choose the top K most similar users, to the target user to predict the values of the unrated items, as its described through the K most similar users (*KMostSimU*) algorithm :

Algorithm 1 : *KMostSimU*

```
Input : $u$ : Target user , $u_i$ : Other users
$n$ : Number of users
$T_{(u,u_i)}$ : Common rated items list of $u$ and $u_i$
$nb$ : Number of common rated items by $u$ and $u_i$
$K$ : Number of most similar users to $u$ to be selected
$S_u$=<$sim_{(u,u_1)}$, $sim_{(u,u_2)}$, ..., $sim_{(u,u_n)}$> :
similarity degrees list, between ($u$,$u_i$) / $i$=1, ..., $n$
Output : The $K$ most similar users to $u$ , $S_u$ =<>
for all user $u = 1$ to $n$ do
    for all user $u_i = 1$ to $n$ do
        if $u \neq u_i$ then
            for all item $t \in T_{(u,u_i)}$ do
                Compute $sim^t_{(u,u_i)}$ ; $nb_{++}$
            end for
            $sim_{(u,u_i)} = \frac{\sum^{nb} sim^t_{(u,u_i)}}{nb}$
        end if
        $S_u = S_u \oplus sim_{(u,u_i)}$ {add $sim_{(u,u_i)}$ to list $S_u$}
    end for
    Select $K$ most similar users to $u$ from $S_u$
end for
```

3.2. Aggregating criteria graphs

Given the test items list T_u of a target user u, we randomly picked up one item and select its criteria graphs list G of the most K similar users to u. Next, we initialize the criteria graph A^t_u of the target user u, with the first one picked in G, that belongs to a certain user u_i. Each edge of this criteria graph, is associated with a new value which is equal to the product of the similarity degree $sim_{(u,u_i)}$ with its initial value. The main idea behind this is, we want to give more emphasis to the criteria graph that belongs to the most similar user, in the aggregation process. The more a user is similar to the target user u, the more his/her criteria graph is implicated in this process. The same revision process is applied to each remaining criteria graphs in G. Then, for each edge in the graph A^t_u , we add or subtract to it, the value of each corresponding edge of the remaining graphs, taken one by one, depending on whether the two edges have the same or opposite orientation. The Aggregating criteria graph (*AggrCG*) algorithm, is performed as follows :

Algorithm 2 : *AggrCG*

```
Input : $u$ : Target user , $T_u$: Test items list of $u$
$G$={$G^t_{u_1}, \cdots, G^t_{u_K}$}:K Criteria graphs of test item $t$
$[\alpha]^u_{(\eta_a,\eta_b)}$ : Value of the edge $(\eta_a, \eta_b)$ in $A^t_u$
$[\alpha]^{u_i}_{(\eta_a,\eta_b)}$ : Value of the edge $(\eta_a, \eta_b)$ in $G^t_{u_i}$
$S_u$=($sim_{(u,u_1)}$ , $sim_{(u,u_2)}$, ..., $sim_{(u,u_K)}$) : K similarity degrees list, between ($u$,$u_i$) / $i$=2, 3, ..., $K$
Output : $A^t_u$: Aggregated criteria graph of the test item $t$
for all test item $t \in T_u$ do
    Initialize $A^t_u$ with $G^t_{u_1}$
    for all edge $(\eta_a, \eta_b) \in A^t_u$ do
        $[\alpha]^u_{(\eta_a,\eta_b)} = ([\alpha]^u_{(\eta_a,\eta_b)} \times sim_{(u,u_1)})$
    end for
    for all $G^t_{u_i} \in G$ s.t. $i = 2$ to $K$ do
        for all edge $(\eta_a, \eta_b)$ of $G^t_{u_i}$ do
            if $\big((\eta_a, \eta_b) \in A^t_u\big)$ and $\big((\eta_a, \eta_b) \in G^t_{u_i}\big)$ have the same orientation then
                $[\alpha]^u_{(\eta_a,\eta_b)}$= $[\alpha]^u_{(\eta_a,\eta_b)} + ([\alpha]^{u_i}_{(\eta_a,\eta_b)} \times sim_{(u,u_i)})$
            else
                $[\alpha]^u_{(\eta_a,\eta_b)}$= $[\alpha]^u_{(\eta_a,\eta_b)} - ([\alpha]^{u_i}_{(\eta_a,\eta_b)} \times sim_{(u,u_i)})$
            end if
        end for
    end for
end for
```

Let's take the criteria graphs $G^t_{u_1}$ and $G^t_{u_2}$ of a hotel test item *Sheraton* of two users u_1 and u_2, similar to the target user u, with $sim_{(u,u_1)} = 0.97$ and $sim_{(u,u_2)} = 0.95$:

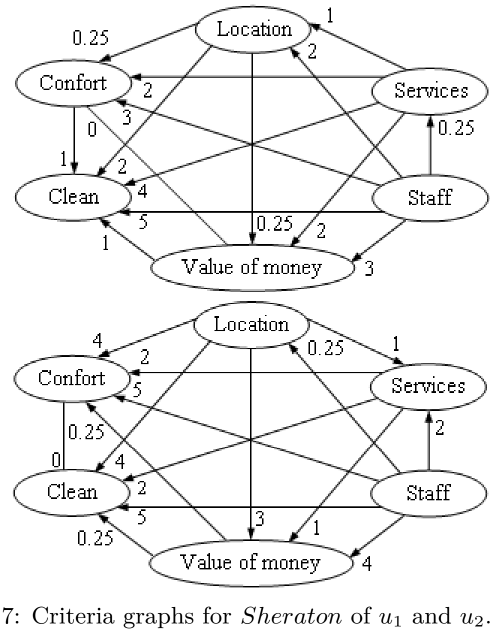

Figure 7: Criteria graphs for *Sheraton* of u_1 and u_2.

The aggregated criteria A^t_u of the target user u for this test item will be:

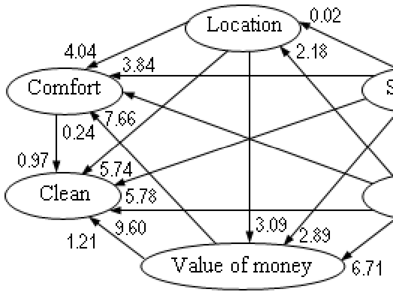

Figure 8: Aggregated graph for *Sheraton*.

4. Predicting preferences

4.1. Predicting criteria valuations

The aggregated criteria graph for each test item t for the target user u, is generated by adding/subtracting to each edge the value of related ones, in the criteria graphs for the same test item, of similar users to u. The overall valuation of each edge will be equal to the fuzzy weighted arithmetic average of all the values of involved edges. This valuation is then, adjusted to match one of the possible surface values α, that express the relationship between preference pairs. For this purpose, the rating's result, will be either 0.25, if the fraction is equal to 5, or rounded-down, for any fraction less than 5 and up, otherwise. It is worth noticing that a review of predicted ratings may be necessary to avoid the Condorcet paradox [10] while performing the prediction, to preserve some coherence properties of the relations (such as transitivity). From this revised graph, we put-up the candidates list of criteria of TFNs, for computing the rating of the test item t, which will be equal to the criteria weighted arithmetic average. Thus, the revised predicted graph (of Fig. 8) for the item *Sheraton* is now:

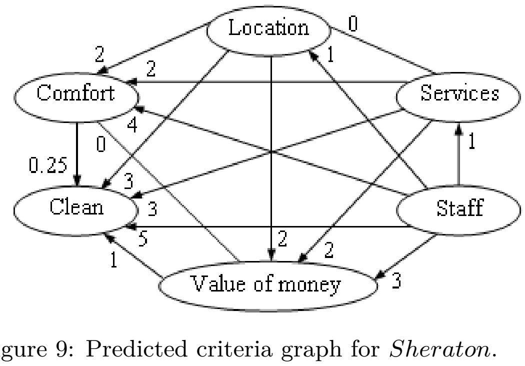

Figure 9: Predicted criteria graph for *Sheraton*.

The values of the edges (*Value of money*, *Clean*) and (*Staff*, *Value of money*) are incoherent with the other relations. Since *Comfort* and *Value of money* are equal they must have the same separating area with the rest of criteria. Its turns out that the surface between *Value of money* and *Staff* isn't the same as *Comfort* and *Staff*, and the surface between *Value of money* and *Clean* is different than *Comfort* and *Clean*. These edges values will be revised to preserve the total order of preferences. The values that fit for those edges are *0.25* and *4* respectively. The only set of possible TFNs generating this criteria graph for the test item Sheraton is:

Clean	Comfort	Location	Services	Staff	Value of money
(6,7,8)	(5,6,7)	(2,3,4)	(2,3,4)	(0,1,2)	(5,6,7)

Table 5: Criteria preferences for *Sheraton hotel*

The list L_u^t of possible ratings for the item *Sheraton* is now composed by the fuzzy weighted arithmetic average and we obtain as preference level to this item $\tilde{A}_1^t$=$(4, 5, 6)$.

4.2. Predicting the items valuations

The final predicting step is to decide among the list L_u^t containing possible scores for each item test t, the one that best suits the target user u. These scores must be consistent with his/her initial graph items, so they don't corrupt its initial total order due to possible intransitive relations. A review step of predicted test items ratings may also be necessary to avoid the Condorcet paradox. To achieve this, we build the overall items graph in which we add one by one the suitable test items. The predicted items graph (*PredIG*) is described as follows:

Algorithm 3 : *PredIG*

```
Input : $G_u$ Initial items graph of the target user $u$
$T_u$: Test items list of the target user $u$
$L_u^t = \{\tilde{A}_1^t, \tilde{A}_2^t, \cdots\}$: Possible TFNs list $\tilde{A}_a^t$ of the test item $t$
$\varphi_a$ : Node representing the test item $t$
$\tilde{T}_b$ : TFN of the node $\varphi_b$
$S(\alpha; \tilde{A}_a^t, \tilde{T}_b)$ : Surface $\alpha$ between $\tilde{A}_a^t$, $\tilde{T}_b$
Output : $G_u$ Predicted items graph of the target user $u$
for all item $t \in T_u$ do
  Add a new node ($\varphi_a$) to $G_u$
  for all ($\varphi_b$) $\in G_u$ do
    if $\tilde{A}_a^t > \tilde{T}_b$ then
      Add an edge ($\varphi_b, \varphi_a$)
    else
      if $\tilde{A}_a^t < \tilde{T}_b$ then
        Add an edge ($\varphi_a, \varphi_b$)
      else
        Add a non-directed edge ($\varphi_a, \varphi_b$)
      end if
    end if
    Assign the weight $S(\alpha; \tilde{A_a}^t, \tilde{T}_b)$ to the arc
  end for
end for
```

By applying Algorithm 3, the overall items graph of the user u is now :

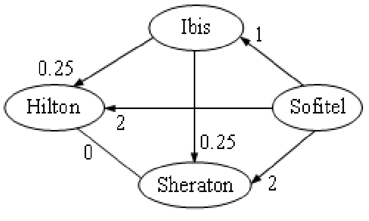

Figure 10: Items graph for the target user u.

5. Analysis

Let us now provide some analysis about our approach to recommender systems. First, note that it has been shown in [2] that using preference relations in recommendation does not degrade the quality of the prediction and decrease by 9% the average ranking of items to recommend comparing to the utility function approach. So, the recommendations based on preference relationships are more precise than those based on utility-based function.

5.1. Metrics

To evaluate recommender systems performance, a myriad of techniques has been proposed. The most well-known of them are the statistical accuracy metrics, such as the Mean Absolute Error *(MAE)* which computes the average deviation between predicted ratings $\hat{r}_{u_i}$ and actual ratings r_{u_i} for all users $u \in U$ and all items of the testing set R_{test}, and the Root Mean Square Error *(RMSE)*, which is similar to *(MAE)*, but places more emphasis on larger deviation [19][14]. The latter is performed as follows:

$$RMSE = \sqrt{\frac{\sum_{r_{u_i} \in R_{test}} (\hat{r}_{u_i} - r_{u_i})^2}{|R_{test}|}} \qquad (9)$$

were $\hat{r}_{u_i}$ is the predicted rating for user u of the item i, r_{u_i} the actual rating of user u for the item i and $|R_{test}|$ is the size of the testing set.

To perform this metric, the users ratings dataset R were divided into $\frac{2}{3}$ for the training set R_{train} used in the prediction step, and $\frac{1}{3}$ for the test step R_{test} used to evaluate the system prediction accuracy.

5.2. Dataset

As far as we know, no dataset exists that handles user preferences under both preference relations and utility function. That's the reason why, we construct a little dataset composed of *38* users, *36* hotels and where each user evaluated the criteria of three hotels, at least. The criteria and the ratings scale were inspired from booking.com website. The criteria are the Clean, Comfort, Location, Facilities, Staff and Value for money and their ratings. The extend ratings values are from *3* up to *10*, since we noticed that the minimum that has been attributed to a hotel was no less than *3* for a large number of hotel we consulted on booking.com website and the maximum rating value was *10*. After, we split the values scale of criteria' ratings, into seven intervals corresponding to our *7* linguistic terms, so that, to assign a qualitative rating for each criterion, from the value *very low* to the value *very high*, as mentioned in the table below.

User criteria rating	Linguistic value
[3.1 – 4]	Very low
[4.1 – 5]	Low
[5.1 – 6]	Medium low
[6.1 – 7]	Medium
[7.1 – 8]	Medium high
[8.1 – 9]	High
[9.1 – 10]	Very high

Table 6: Linguistic values of Criteria' ratings

5.3. Results and discussion

We perform the prediction for *12* users who initially evaluated some hotels taken from the R_{test}, in order to measure how close are the predicted ratings of these hotels to the users initial ratings. We then plot the fuzzy ratings of users and those predicted by the system. The analysis of the experiment showed that our approach does pretty well, because the system predicted ratings coincide with the majority of users' initial ratings except for small values, where we notice a slight gap between them, as illustrated on the following graph.

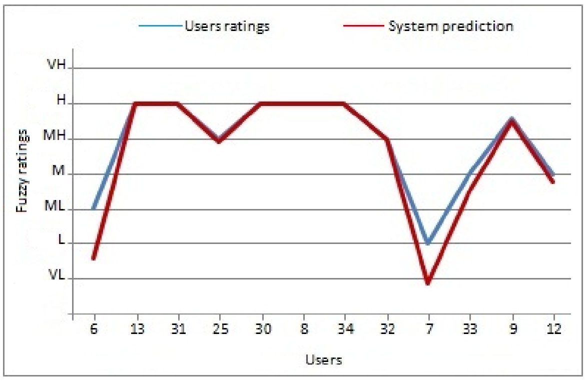

Figure 11: Prediction scores breakdown.

The estimated *RMSE* is equal to *0.8979*, which denotes a high precision of the recommender system, comparing to Naive baseline implementation that use constant rating such as the average of all the available ratings. It *RMSE* is around *1.25*, regarding Netflix Cinematch, it's equal to *0.9514* [21].

5.4. Limits

Unfortunately, using preference relations leads to an additional cost, comparing to classical recommendation systems due to the required processing time to turn rough preferences into preference relationships and the increasing comparison number between preferences with the growing users number. For instance, for n_c criteria number, $\frac{n_c \times (n_c - 1)}{2}$ preference relations are needed in the criteria graph for each item i and for n_i items number, $\frac{n_i \times (n_i - 1)}{2}$ preference relations are involved in the items graph, for each user u. If we take into account N_u users of the systems, the whole the recommendation process requires $\frac{(N_u) \times (n_i)^2 \times (n_i - 1) \times (n_c) \times (n_c - 1)}{4}$ preference relations to manage. That may reach *570000* preference relations, just for *10* users, *20* items and

6 criteria of the dataset, for instance. On the other hand and from an optimistic point of view, each user u evaluates just few items in recommender system framework, while the remaining ratings are missed values. This leads to a high sparsity of users/items ratings, which can reach 99% of unrated items such as Netflix movies dataset [7].

6. Conclusion

Using preference relation ratings instead of absolute ratings is a promising technique, recently introduced. In this work, we proposed adding to it, a fuzzy modeling of user multi-criteria preferences by using linguistic terms. This fuzzy modeling enhance the precision of current recommender system, since it faithfully translates the user's qualitative preferences. To achieve that, we represented the preferences and their relationships through a graph. These latter were quantified by values, which measure how much a preference is better compared to others. As, we are dealing with multi-criteria preferences, a criteria graph for each rated item was first built, to generate afterward the items graph whose ratings are the fuzzy weighted arithmetic average of all involved criteria ratings. The aggregated graph for a target user is then composed of criteria graphs of k first similar users. The prediction of unrated items, are finally derived from the aggregated criteria graph, while maintaining coherence between preferences relationships and the consistency of the user's initial ratings. Preliminary results show the advantage of using fuzzy preference relations instead of absolute ratings, to enhance the accuracy of current recommender systems and as future work, we intend to carry on a more profound experimental study with a larger dataset to prove the validity of our approach in improving the predictions' quality and the recommender system accuracy.

References

[1] G. Adomavicius, N. Manouselis and Y. Kwon, Multi-Criteria Recommender Systems, *Recommender Systems Handbook*, 769–803, 2011.

[2] A. Brun, A. Hamad, O. Buffet and A. Boyer, Towards Preference Relations in Recommender Systems, *(ECML-PKDD, workshop on Preference Learning)*, 2010.

[3] Y. Cao and Y.Li, An intelligent fuzzy-based recommendation system for consumer electronic products, *Expert Systems with Applications*, 33:230–240, 2007.

[4] C. Cheng, A new approach for ranking fuzzy numbers by distance method, *Fuzzy sets and Systems*, 95:307–317, 1998.

[5] M. Degemmis, L. Iaquinta, P. Lops, C. Musto, F. Narducci and G. Semeraro, Preference Learning in Recommender Systems, *(ECML PKDD 2009)*, 41–54, 2009.

[6] M.S. Desarkar, S.Sarkar, P.Mitra, Aggregating Preference Graphs for Collaborative Rating Prediction, *(RecSys 2010)*, 21–28 , Barcelona (Spain), 2010.

[7] M.S. Desarkar, S.Sarkar, Rating Prediction Using Preference Relations Based Matrix Factorization, *(UMAP 2012)*, Montreal (Canada), 2012.

[8] D. Dubois and H. Prade, Fundamental of Fuzzy Sets, *Handbooks of Fuzzy Sets Series*, 7:147–156, 2000.

[9] Y. Freund and R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *(EuroCOLT '95)*, 23–37, 1995.

[10] W.V. Gehrlein, editor. *Condorcet's paradox vol. 40*, Theory and decision library, Springer-Verlag, Germany, 2006.

[11] F. Herrera, L. Martinez. A2-Tuple Fuzzy Linguistic Representation Model for Computing with Words, *IEEE transactions on fuzzy systems*, 8(6):746–752, IEEE, 2000.

[12] F. Herrera, E. Herrera-Viedma. Linguistic decision analysis: steps for solving decision problems under linguistic information, *Fuzzy Sets and Systems*, 115:67–82, 2000.

[13] E. Herrera-Viedma, F. Herrera, F. Chiclana, M.Luque. Some issues on consistency of fuzzy preference relations, *European Journal of Operational Research*, 154:98–109, 2004.

[14] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, editor. *Recommender Systems An Introduction*, Cambridge University Press, USA, 2011.

[15] I. Liousse. Calcul intégral. Technical Report, Mathematic department, Lille1 University, France, December 2008.

[16] H. Ma, I. King and M.R. Lyu, Effective Missing Data Prediction for Collaborative Filtering, *Routing and Filtering (SIGIR)*, 39–46, 2007.

[17] K. Palanivel, R. Sivakumar. Fuzzy multicriteria decision-making approach for Collaborative recommender systems, *International Journal of Computer Theory and Engineering*, 2(1): 1793–8201, 2010.

[18] C. Porcel , A.G. Lopez-Herrera and E. Herrera-Viedma, A recommender system for research resources based on fuzzy linguistic modeling, *Expert Systems with Applications*, 36:5173–5183, 2009.

[19] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, editor. *Recommender Systems Handbook*, Springer, London, 2011.

[20] J.H. Wang. A new version of 2-tuple fuzzy linguistic representation model for computing with words, *IEEE transactions on fuzzy systems*, 14(3):435–445, 2006.

[21] Z. Wen, Recommendation System Based on Collaborative Filtering. Technical Report, CS229, Stanford University, USA, December 2008.