Building User profiles for Recommender Systems from incomplete preference relations

Luis G. Pérez, Manuel Barranco, Luis Martínez

Abstract— **In recent years, in the e-commerce area new methods and tools have arisen in order to improve and customize the e-commerce web sites according to users' necessities and tastes. The most successful tool in this field have been the Recommender Systems. The aim of them is to assist people to find the best alternatives that satisfy their necessities using recommendations, leading them to interesting items, or hiding those useless and unattractive ones. Sometimes these systems face situations where there is a lack of information and this implies unsuccessful results. Although some solutions have been proposed, they present some limitations and drawbacks. In this contribution we present a knowledge-based recommender system that is able to compute successful recommendations using a few numbers of examples about the items that the users are looking for.**

I. INTRODUCTION

Recommender Systems have been one of the key issues in the development and success of the e-commerce. Customers usually face a huge range of products that potentially can meet their requests, however only a small set of them fulfil their tastes and/or necessities and many times they are hard to find. Although this may seem that we are dealing with a typical decision making problem it is not. E-commerce web sites offer a quantity of alternative(s) so vast that users are unable to explore all of them. Due to this fact, instead of choosing the most suitable items that fulfil their necessities, they usually select the first ones that fulfil them. Recommender Systems assist people with their searches leading them towards interesting products by means of recommendations and/or limiting and sorting the offered products.

Essentially all the Recommender Systems have the same aim: to lead users through recommendations to those products that are the most suitable for them. However, the techniques utilized to achieve this aim are different from each other, so in the required information as in the necessary processes to compute the recommendations. According to these techniques, we can classify the Recommender Systems in Demographic Recommender Systems [14], Content-based Recommender Systems [18], Collaborative recommender systems [9], Knowledge based recommender systems [3] and Hybrid Recommender Systems [2].

To choose the most suitable products for a user, these systems need to gather information about the products, the users and their necessities. However, in some cases, this information is scarce and insufficient. Classical recommender systems, the collaborative and content-based, are unable to make any suitable recommendation in such cases. To overcome these drawbacks some proposals have been presented. One of them is the knowledge based recommender systems. In this kind of systems, users state their preferences choosing an example that represents their preferences. With the description of this example, a user profile is defined and then, this profile is used to find the most similar products that are returned as recommendations. Sometimes, the information kept in this user profile does not express exactly the user's preferences and it is necessary to ask users to change some of the features of the given example. This task could be tedious depending on the number of features used in the description of the example and not all the users could be willing or trained to accomplish this refinement.

The aim of this contribution is to improve the gathering process of the Knowledge-Based Recommender Systems using more examples and employing fuzzy preference relations. These preference relations will let us model and exploit in a more efficient way the information provided by the user but without forcing him/her to waste time giving information that the system can compute and complete by itself. For instance, users are not required to provide a complete preference relation because the system can complete it by itself using consistency properties. Thus, the system will increase the information provided by the user in order to exploit it and, thereby, it will generate better recommendations.

This contribution is structured as follows: in section 2 we shall review some preliminaries we have to know to understand our problem and how our model works. In section 3 we shall present our proposal. In section 4 we show an example with our model and finally in section 5 some conclusions are point out.

II. PRELIMINARIES

In this section we shall analyze the problem of lack of information in Recommender Systems. Besides, we shall review the consistency in fuzzy preference relations and how we can use this property to fill up an incomplete relation.

Luis G. Pérez is with the Department of Computer Sciences, University of Jaen, Edificio Tecnologico y de Ingenierías A-3, Paraje Las Lagunillas, s/n, 23071 Jaen, Spain (phone: +34 953 213018; email: lgonzaga@ujaen.es).

Manuel J. Barranco is with the Department of Computer Sciences, University of Jaen, Edificio Tecnologico y de Ingenierías A-3, Paraje Las Lagunillas, s/n, 23071 Jaen, Spain (phone: +34 953 212923; email: barranco@ujaen.es).

Luis Martínez is with the Department of Computer Sciences, University of Jaen, Edificio Tecnologico y de Ingenierías A-3, Paraje Las Lagunillas, s/n, 23071 Jaen, Spain (phone: +34 953 211902; email: martin@ujaen.es).

1-4244-1210-2/07/$25.00 ©2007 IEEE.

A. Lack of Information in Recommender Systems

Although the Collaborative [9] and the Content-based Recommender system [18] are the most used and the most well-known recommender systems not always they are suitable and sometimes present important drawbacks. For example, the Collaborative Recommender Systems needs to have a huge database of ratings of products from many users to filter and recommend products to a specific user. In the simplest case, these systems predict the users' preferences as a weighted aggregation of the other users' preferences, in which the weights are proportional to the similarity between users on the basis of their ratings. On the other hand, Content-based Recommender Systems look for new products that are similar to those that the user has bought in the past. Therefore, both systems require that the user has assessed a minimum number of products to suggest good recommendations.

However, in the real world we find situations where the before models are not suitable because of lack of information. Some of the most common problems in these models are [4]:

- *The new user ramp-up problem:* It is very difficult to make recommendations to users that have few ratings. We can find this problem both in Content-based Recommender Systems and in Collaborative ones.
- *New item ramp-up problem:* in Collaborative Recommender Systems if an item has not many ratings, it is not easy to be recommended.
- *Grey sheep problem:* in Collaborative Systems there might exist users whose ratings are not consistently similar with any group of users, and for this reason, they will rarely receive any accurate recommendation.
- *Quality dependent on large historical data set.*

To sort out these problems some alternatives have been presented, such as the Hybrid Recommender Systems [2] or the Knowledge Based Recommender Systems [3]. The first ones combine the collaborative and content-based algorithm to smooth out the disadvantages of both recommender systems (for instance, they do not suffer from new item ramp-up problem). Although they present better results than the Content-based and Collaborative recommender systems and they are able to make recommendations with less information, they are usually still unable to make recommendations to new users (they could suffer from the new user ramp-up problem).

On the other hand, Knowledge-based Recommender Systems have the advantage that they do not suffer from these problems although now they need a knowledge acquisition process[4]. These systems exploit the information provided by the user about their necessities and the knowledge that the system has in a database of products to make recommendations of those products that satisfy the users' necessities. There are several methods to use this knowledge, for example The PersonalLogic used dialogs that lead users through a discrimination tree of product features and Entree [5], [6] uses case based reasoning [10] to make recommendations. In this last type of Recommender Systems users give examples of the product they are looking for, and the system searches and recommends similar products to the given example. Many times, users do not search a product exactly equal to the example and need to refine these searches stating or modifying some of the features of the given example. In both techniques, new users can obtain recommendations from the system stating what they need. However if the products are described with many features, the process of gathering the user's needs could be tedious, and many times users could not be willing or trained to do so.

B. Fuzzy preference relations and consistency properties

In this contribution we shall deal with fuzzy preference relations to gather the users' preferences. They have been widely used to model preferences for decision-making problems [7], [13], [17]. In this representation the intensity of preference between any two alternatives of a set of feasible ones, $X = \{x_1, ..., x_n\}$ $(n \geq 2)$, is measured with a scale $[0, 1]$.

Definition 1. *[7] A fuzzy preference relation P on a set of alternatives X is a fuzzy set on the product set $X \times X$, i.e., it is characterized by a membership function*

$$\mu_P : X \times X \to [0, 1]$$

Every value in the matrix P represents the preference degree or intensity of preference of the alternative x_i over x_j:

- $p_{ij} = 1/2$ indicates the greatest grade of indifference between x_i and x_j $(x_i \sim x_j)$.
- $p_{ij} = 1$ indicates that x_i is absolutely preferred to x_j
- $p_{ij} > 1/2$ indicates that x_i is preferred to x_j $(x_i \succ x_j)$

based on this representation we also know that $p_{ii} = 1/2 \forall i \in \{1, \ldots, n\}$ $(x_i \sim x_i)$

In an ideal situation the information provided by the user should be consistent and complete. However, many times in real situation this is not possible. A fuzzy preference relation is complete if for every alternative (x_i, x_j), p_{ij}, is known. The concept of consistency is usually characterized by the idea of transitivity. Transitivity represents the idea that the preference value obtained by comparing directly two alternatives should be equal to or greater than the preference value between those alternatives obtained using an indirect chain of alternatives [8], [15], [19]. Some of the suggested transitivity properties that we can find in the literature are the *Triangle condition* [15], the *Weak transitivity* [19] or the *Additive transitivity* [19].

The last one seems an acceptable property to characterize consistency in fuzzy preference relations and has been used successfully to construct consistent fuzzy preference relations from incomplete ones [1], [12].

Definition 2. *[1], [12] A fuzzy preference relation is "additive consistent" when for every three options on the problem $x_i, x_j, x_k \in X$ their associated preference degrees p_{ij}, p_{jk}, p_{ik} fulfil the following expression [12]:*

$$(p_{ij} - 0.5) + (p_{jk} - 0.5) = (p_{ik} - 0.5) \forall i, j, k$$

A simple and practical method for constructing a complete preference relation can be:

Step 1. Let $X = \{x_1, ..., x_n\}$ be a discrete set of alternatives. The expert provides a row (or a column) of the preference relation

Step 2. Utilize the known elements in P to determine all the unknown elements, and thus get a consistent preference relation, P', using the following expressions obtained from definition 2:

1) $p_{ij} + p_{jk} + p_{ki} = \frac{3}{2}$
2) $p_{i(i+1)}+p_{(i+1)(i+2)}+\cdots+p_{(j-1)j}+p_{ji} = \frac{j-i+1}{2}\forall i < j$

Step 3. End.

Example Suppose that we have a set of four alternatives $\{x_1, x_2, x_3, x_4\}$. If we now that $\{p_{12} = 0.55, p_{13} = 0.7, p_{14} = 0.95\}$, we shall have the following preference relation:

$$P = \begin{pmatrix} 0.5 & 0.55 & 0.7 & 0.95 \\ & 0.5 & & \\ & & 0.5 & \\ & & & 0.5 \end{pmatrix}$$

if we use the previous algorithm we obtain:

$$P' = \begin{pmatrix} 0.5 & 0.55 & 0.7 & 0.95 \\ 0.45 & 0.5 & 0.65 & 0.9 \\ 0.3 & 0.35 & 0.5 & 0.75 \\ 0.05 & 0.1 & 0.25 & 0.5 \end{pmatrix}$$

III. A Knowledge Based Recommender System based on consistency

In this section we shall present our Knowledge Based Recommender System based on the use of consistent fuzzy preference relation. This system makes its recommendations exploiting the information given by a user about their needs and the information the system has about the products to be recommended. The main advantage of this system is that neither needs a complex gathering information process, nor an historical log about what the users have bought or liked in the past as in Collaborative [9] or Content-based Recommender Systems [18]. Our system gathers information from users asking them for examples about what they need and it refines this information using the user's preference about the chosen examples. With this information a user profile is built and used to find the most suitable product(s) for the user.

Moreover, on the contrary to other knowledge based recommender system [5], [6], in our proposal, users do not need to state or modify products' features to refine their needs. The system defines the user profile from the description of the given example and from his/her preferences about them.

This model is structured as follows (see figure 1):

- *Gathering user's preference information:* The system will offer a set of illustrative products or alternatives and the user chooses the closest ones to their necessities, tastes, or preferences. Secondly, the user provides preference information about which of these examples of his/her necessities are closer to his/her real expectations by means of a preference relation structure. Finally, from this information the system will define a complete preference relation that represents the user's preference.
- *Obtaining the user profile:* In this step the system will set the user profile from the preference relation and from the description of these examples.
- *Recommendation:* The user profile will be used to find those products that can fulfil the user's necessities, taste or preferences.

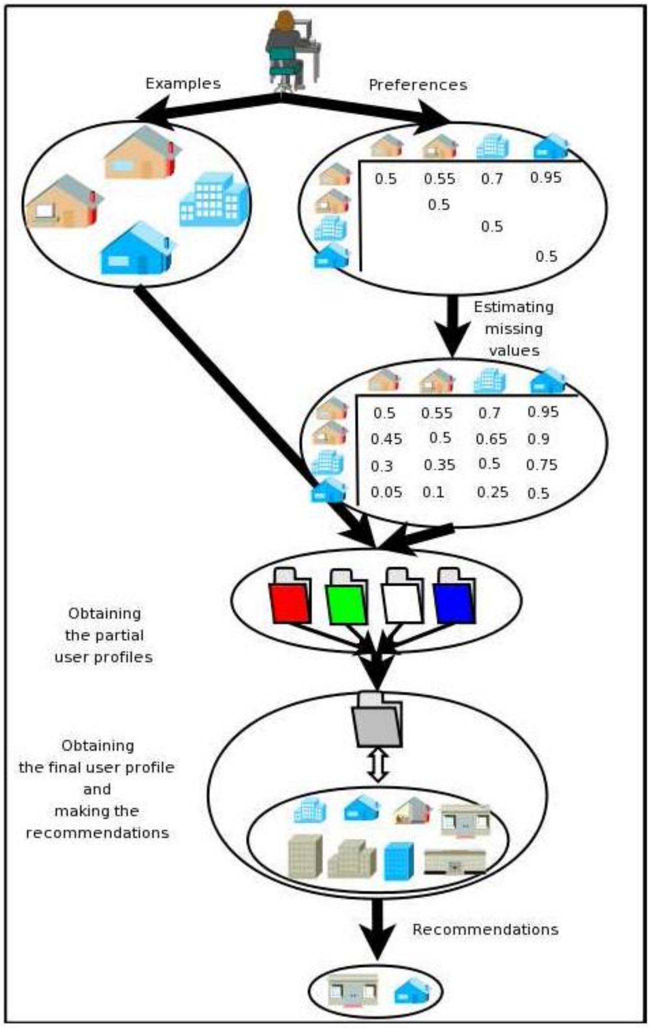

Fig. 1. Recommendation Model

In the next subsections we explain in detail these steps.

A. Gathering user preference information

In this phase, users choose a set of example products, normally four or five, close to what they are looking for to outline their preferences. We should remark that in this step all the products of the product database are not showed but just a small subset of them. This subset is bound to content the most illustrative products of the database, i.e., the most well-known but at the same time products that are not liked by everybody. Moreover this set is required to be big enough to allow users to find suitable examples of their necessities, but not too big because our aim is that users can find easily examples about what they want. If $X = \{x_1, x_2, ..., x_m\}$ is the set of products that can be recommended and each one is described by a vector of features $x_i = \left\{c_i^1, \ldots, c_i^t\right\}$ $c_i^l \in [0,1]$, we have to define a subset $X_r = \{x_1^r, x_2^r, ..., x_{m'}^r\}$ $(m' < m)$ that shall contains the most illustrative products of X $(X_r \subseteq X)$

Moreover, this model offers a preference relation structure to inquire the users their preferences, such that, they have to provide a complete row (or column) of the preference relation. From this information and the algorithm proposed in [12] we shall fill up the incomplete relation (see section II.B) obtaining a complete and consistence preference relation. For example if the system requires four examples and the user provides the values of the first row, after this step we shall obtain the next relationship:

$$P' = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21}^* & p_{22} & p_{23}^* & p_{24}^* \\ p_{31}^* & p_{32}^* & p_{33} & p_{34}^* \\ p_{41}^* & p_{42}^* & p_{43}^* & p_{44} \end{pmatrix}$$

Where $p_{ii} = 0.5$, p_{1j} is the value that the user provided about the preference of example x_1^r over the example x_j^r $(j \neq 1)$, and p_{ij}^* is the value estimated for the preference of the example x_i^r over x_j^r according to the consistency property.

B. *Obtaining the user profile*

The aim of this step is to obtain a user profile. The user profile is used to represent the user's necessities and tastes and it is used to find the most suitable item for that user. Both the representation of the user profile and the gathering process of information play a key role in the recommendation processes.

In our case, each product x_i is described by a set of features $\left\{c_i^1, \ldots, c_i^t\right\}$ and we shall define a user profile from the preferences, p_{ij}, of the relation and from the description of the given examples. To do so, we shall accomplish the following steps:

1) Obtaining the partial user profiles: Given a column of known elements $(p_{1j}, p_{2j}, \ldots, p_{nj})$ of the preference relation P', which represents the preference of every product over x_j^r, we can obtain a partial profile concerning the product x_j^r by aggregating the descriptions of the products that are being compared with x_j^r using a weighted mean aggregation operator. The aggregation function we have chosen is the IOWA operator (Induced OWA operator) proposed by Yager [21].

The IOWA operator is used to aggregate tuples of the form (v_i, a_i). Within these pairs, v_i is called the order inducing value and a_i is called the argument value. The following procedure for performing the IOWA aggregation was suggested:

$$F_W\left(\langle v_1, a_1\rangle, \ldots, \langle v_l, a_l\rangle\right) = W^T B_v$$

where $B_v = (b_1, \ldots, b_l)$ is the result of ordering the vector $A = (a_1, \ldots, a_l)$ according to the value of the order inducing variables, v_i, indeed of the values of the elements a_i, and W^T is the column vector of weights which keeps the following conditions:

$$W = (w_1, \ldots, w_l)$$
$$w_i \in [0,1] \quad \forall i \sum_{i=1}^{l} w_i = 1$$

In our case, we use this operator for aggregating the descriptions $\left\{(c_i^1, \ldots, c_i^t), \forall i \neq j\right\}$ of the products $\{x_i^r, \forall i \neq j\}$ chosen as example of the user's necessities, according to the order induced by the variables, $(p_{1j}, p_{2j}, \ldots, p_{nj})$, the column j from the preference relation, P', that describes the preferences of the other examples over the example x_j^r. The result will a user partial profile, pp_j, according to the product x_j^r given as a tuple $\left(c_{pp_j}^1, \ldots, c_{pp_j}^t\right)$ where each element $c_{pp_j}^k$ is obtained by aggregation of the elements $\left\{c_i^k, \forall i \neq j\right\}$. So, for every attribute we apply the following function:

$$c_{pp_j}^k = F_W\left(\langle p_{1j}, c_1^k\rangle, \ldots, \langle p_{nj}, c_n^k\rangle\right) = W^T B_v$$

where the vector $B_v = (b_1, \ldots, b_{n-1})$ is given by a decreasing order of the elements belonging to the set $\left\{c_i^k, \forall i \neq j\right\}$ according to such variables, $(p_{1j}, \ldots, p_{nj})$ $\{\forall i \neq j\}$.

In the literature there are different methods to compute the weighting vector $W = (w_1, \ldots, w_{n-1})$. We could associate it with a linguistic quantifier [20] or resolve a mathematical problem such as in [16]. The selection of the method will depend on the type of problem, products and on the results we want to obtain.

2) Obtaining the final user profile: Once we have calculated the partial user profiles, we shall combine them to obtain the final user profile that will be used in the recommendation phase. In this step we shall aggregate the partial profiles, $pp_1, \ldots,$ pp_n , using a weighted mean. As in the before step, we propose to use the IOWA operator. In this case we shall aggregate every partial profile, $\left(c_{pp_j}^1, \ldots, c_{pp_j}^t\right)$, obtained for every product x_j^r. So, for every attribute we shall apply the next function:

$$c_{fp}^k = F'_W\left(\langle p_1, c_{pp_1}^k\rangle, \ldots, \langle p_n, c_{pp_n}^k\rangle\right) = W'^T B'_v$$

where the vector $B'_v = (b'_1, \ldots, b'_n)$ is given by a decreasing order of the elements of the set $\left\{c_{pp_i}^k\right\}$ according to such order inducing variables, $(p_1, \ldots, p_n)$, and the weighting vector $W' = (w'_1, \ldots, w'_n)$.

The induced variables $(p_1, \ldots, p_n)$ represents the importance of each alternative. The most important alternative, which is the nearest to the user's needs, will have the greatest value and the furthest the smallest. To obtain these values we can use the same dominance degree that is used in [11]:

$$p_i = \frac{1}{n-1} \sum_{j=0 \,|\, j \neq i}^{n} p_{ij}$$

This choice function computes the dominance degree for each alternative, x_i^r, over the rest of alternatives.

The final user profile will be:

$$FP_u = \{c^1_{fp}, \ldots, c^t_{fp}\}$$

C. Recommendation

To achieve its objective, the last step of the system is the recommendation phase. There, the system will compute the closest products to the user profile. We have aforementioned that our product database $X = \{x_1, x_2, ..., x_m\}$ contains all the products we can recommend and in which the product x_i will be described by the following set of features $x_i = \{c^1_i, \ldots, c^t_i\}$. In the previous steps we have obtained a final user profile $FP_u = \left\{c^1_{fp}, \ldots, c^t_{fp}\right\}$. Now, we shall compute the similarity of the description for each product, x_i with the user profile:

$$\nu\left(x_i, FP_u\right) = \phi\left(\nu_1\left(c^1_i, c^1_{fp}\right), \ldots, \nu_t\left(c^t_i, c^t_{fp}\right)\right) \in [0, t]$$

where ϕ is a weighted aggregation operator. The choice of this operator depends on the application, e.g., we could be interested in aggregating these values by means of a weighted mean, so that we could take into account the relative importance of each attribute. The function ν_j is a similarity measure for each attribute:

$$\nu_j\left(c^j_i, c^j_{fp}\right) = \gamma\left(\left|c^j_i - c^j_{cp}\right|\right)$$

where γ is a increasing function valued into $[0, 1]$ and such that $\gamma(0) = 0$

The final recommendation(s) will be those products that are closer to the final profile, i.e., its overall similarity is closer to zero, but the user has not chosen them yet as examples.

IV. Example

In this section we shall apply our model to a specific problem where a user wants to obtain some recommendations. The system will show the set X_r of the most illustrative examples of the system, and the user will select the four closest examples of his/her necessities (see Table I):

TABLE I
GIVEN EXAMPLES

Product ID	Description
1	$(1.0, 0.2, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0)$
2	$(1.0, 0.3, 1.0, 1.0, 0, 0, 1.0, 0, 1.0, 1.0)$
3	$(0.5, 0.1, 0.4, 0.8, 1.0, 1.0, 1.0, 0.4, 0.9, 0.9)$
4	$(0.1, 0.3, 0.3, 0.9, 1.0, 0, 0.78, 0, 0.85, 0.95)$

Moreover, the user gives their preferences about the given example. In our case, he provides the preference of the first product over the other ones:

$$P = \left(\begin{array}{cccc} 0.5 & 0.25 & 0.4 & 0.65 \\ & 0.5 & & \\ & & 0.5 & \\ & & & 0.5 \end{array}\right)$$

Now, with these preference values the system can find and recommend the most suitable products among all the products it has in its products database (see Table II):

TABLE II
PRODUCTS DATABASE

Product ID	Description
1	$(1.0, 0.2, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0)$
2	$(1.0, 0.3, 1.0, 1.0, 0, 0, 1.0, 0, 1.0, 1.0)$
3	$(0.5, 0.1, 0.4, 0.8, 1.0, 1.0, 1.0, 0.4, 0.9, 0.9)$
4	$(0.1, 0.3, 0.3, 0.9, 1.0, 0, 0.78, 0, 0.85, 0.95)$
5	$(0.74, 0.37, 0.26, 0.41, 0.39, 0.86, 0.22, 0.050, 0.62, 0.62)$
6	$(0.36, 0.52, 0.74, 0.28, 0.42, 0.14, 0.76, 0.12, 0.36, 0.59)$
...	...
11	$(0.20, 0.18, 0.61, 0.93, 0.28, 0.49, 0.78, 0.88, 0.49, 0.67)$
...	...
21	$(0.82, 0.30, 0.89, 0.46, 0.38, 0.12, 0.26, 0.27, 0.57, 0.49)$

First of all, the system can fill the user's preference using the algorithm reviewed in section II.B and obtain a complete and consistent fuzzy preference relation:

$$P' = \left(\begin{array}{cccc} 0.5 & 0.25 & 0.4 & 0.65 \\ 0.75 & 0.5 & 0.65 & 0.9 \\ 0.6 & 0.35 & 0.5 & 0.75 \\ 0.35 & 0.1 & 0.25 & 0.5 \end{array}\right)$$

In the next phase the system will compute the user profile, but before computing it, we compute the weights that will be used to obtain the partial profiles and the final user profile. To obtain these weights we shall use the following function [20]:

$$w_i = Q\left(\frac{i}{m}\right) - Q\left(\frac{i-1}{m}\right), \quad i = 1, \ldots, m$$

where m is the number of values we are going to aggregate, and Q is the '*'at least half*" linguistic quantifier [20]:

$$Q^{alh}(x) = \begin{cases} 0 & si\ x < a \\ \frac{x-a}{b-a} & si\ a \leq x \leq b \quad with\ a = 0,\ b = 0.5 \\ 1 & si\ x > b \end{cases}$$

We shall use the above function to obtain the weighting vector, W and W', that will be used to obtain the partial user profiles and the final user profile respectively. The values obtained for the first vector are $W = \{0.67, 0.33, 0\}$ and for the second one $W' = \{0.5, 0.5, 0, 0\}$.

With these weights and using the user's preference relation we can aggregate the products descriptions to obtain a partial profile. For example, to obtain the first value of partial profile related to the first example, pp_1, the system shall compute:

$$c^1_{pp_1} = F_W\left(\langle 0.75, 1\rangle, \langle 0.6, 0.5\rangle, \langle 0.35, 0.1\rangle\right) = 0.83$$

The partial profiles are (see Table III):

To obtain the final user profile we shall aggregate the partial profiles using the weights W':

$$c^1_{fp} = F'_W(\langle 0.57, 0.83\rangle, \langle 0.23, 0.67\rangle, \langle 0.43, 1\rangle,$$

TABLE III

PARTIAL PROFILES

Partial profile	Description
pp_1	$(0.83, 0.23, 0.8, 0.93, 0.33, 0.33, 1, 0.13, 0.97, 0.97)$
pp_2	$(0.67, 0.13, 0.6, 0.87, 1, 1, 1, 0.6, 0.93, 0.93)$
pp_3	$(1, 0.27, 1, 1, 0.33, 0.33, 1, 0.33, 1, 1)$
pp_4	$(0.83, 0.23, 0.8, 0.93, 0.33, 0.33, 1, 0.13, 0.97, 0.97)$

$$\langle 0.77, 0.83\rangle) = 0.83$$

If we compute all the values we obtain the following results (see Table IV):

TABLE IV

FINAL PROFILE

Final profile
$(0.83, 0.23, 0.8, 0.93, 0.33, 0.33, 1., 0.13, 0.97, 0.97)$

The last step of our model is the recommendation phase. In this phase the system will compare the final user profile with the description of each product of the product database and it will recommend those products that are the closest to the user's necessities. In this problem we shall use the Euclidean distance to measure the similarity between the final user profile and the products (see Table V).

TABLE V

RECOMMENDATIONS

Product ID	Score (Distance)
21	2.65
11	2.81
6	2.99
...	...

Therefore, according to these results the closest product to the user necessities is the product 21, the second one is the 11, the next one is the 6 and so on.

V. CONCLUSIONS

There are some situations where Classical Recommender Systems, the Content-based and the Collaborative ones, cannot be applied because they have a limited knowledge about the users tastes and preferences. We propose the use of a Knowledge Based Recommender System to solve this problem. Our proposal consist of gathering the information from the user using a fuzzy preference relation structure that only requires to be filled with a small number of values. Then, using the consistency property the system will complete the preference relation and it will exploit it to obtain recommendations. The main advantage of this model is that it does not force the user to spend much time in the generation of his/her profile and provides accurate recommendations.

In future works we shall study the use of more complex representations for the products such as it is defined in [22]

VI. ACKNOWLEDGEMENTS

This work is partially supported by the Research Projects TIN-2006-02121, JA031/06 and FEDER funds

REFERENCES

[1] S. Alonso, F. Chiclana, F. Herrera, E. Herrera-Viedma, J. Alcalá, and C. Porcel. A consistency based procedure to estimate missing pairwise preference values. *International Journal of Intelligent Systems*, Forthcoming, 2006.

[2] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and contentbased information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 714–720, 1998.

[3] R. Burke. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69(32), 2000.

[4] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[5] R. D. Burke, K. J. Hammond, and B. C. Young. Knowledge-based navigation of complex information spaces. In *AAAI/IAAI, Vol. 1*, pages 462–468, 1996.

[6] R. D. Burke, K. J. Hammond, and B. C. Young. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40, 1997.

[7] F. Chiclana, F. Herrera, and E. Herrera-Viedma. Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations. *Fuzzy Sets and Systems*, (97):33–48, 1998.

[8] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, 1980.

[9] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61 – 70, 1992.

[10] K.J. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press Inc, 1989.

[11] F. Herrera, L. Martinez, and P.J. Sanchez. Managing non-homogeneous information in group decision making. *European journal of operational research*, 166:115–132, 2005.

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

[13] J Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18(2):105–118, 1986.

[14] B. Krulwich. Lifestyle finder: intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.

[15] R.D. Luce and P. Suppes. *Handbook of Mathematical Psychology*, chapter Preferences, Utility and Subject Probability, pages 249–410. Wiley, New York, 1965.

[16] M. O'Hagan. Aggregating template rule antecedents in real-time expert systems with fuzzy set logic. In *22th Annual IEEE Asilomar Conference on Signals, Systems and Computers*, 1988.

[17] S.A. Orlovsky. Decision-making with a fuzzy preference relation. *Fuzzy Sets Systems*, 1:155–167, 1978.

[18] M. J. Pazzani, J. Muramatsu, and D. Billsus. Syskill webert: Identifying interesting web sites. In *AAAI/IAAI, Vol. 1*, pages 54–61, 1996.

[19] T. Tanino. *Non-Conventional Preference Relations in Decision Making*, chapter Fuzzy Preference Relations in Group Decision Making, pages 54–71. Springer-Verlag, 1988.

[20] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183 – 190, 1988.

[21] R. R. Yager. Induced aggregation operators. *Fuzzy Sets and Systems*, 137(1):59 – 69, 2003.

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