

## A KNOWLEDGE BASED RECOMMENDER SYSTEM WITH MULTIGRANULAR LINGUISTIC INFORMATION

Luis Martínez, Manuel J. Barranco, Luis G. Pérez, Macarena Espinilla  
*Dpt. of Computer Science, University of Jaen*  
*23071 - Jaen, Spain*  
*E-mail: {martin,barranco,lgonzaga,mestevez}@ujaen.es*

Received: 03-01-2008

Revised: 30-06-2008

Recommender systems are applications that have emerged in the e-commerce area in order to assist users in their searches in electronic shops. These shops usually offer a wide range of items that cover the necessities of a great variety of users. Nevertheless, searching in such a wide range of items could be a very difficult and time-consuming task. Recommender systems assist users to find out suitable items by means of recommendations based on information provided by different sources such as: other users, experts, item features, etc. Most of the recommender systems force users to provide their preferences or necessities using an unique numerical scale of information fixed in advance. In spite of this information is usually related to opinions, tastes and perceptions, therefore, it seems that is usually better expressed in a qualitative way, with linguistic terms, than in a quantitative way, with precise numbers. We propose a Knowledge Based Recommender System that uses the fuzzy linguistic approach to define a flexible framework to capture the uncertainty of the user's preferences. Thus, this framework will allow users to express their necessities in scales closer to their own knowledge, and different from the scale utilized to describe the items.

### 1. Introduction

One of the main problems users face when they are surfing in Internet is the vast quantity of information, being most of it useless. For instance, most of the e-shops offer thousand of products that conform a search space that the user can not evaluate carefully in order to find out the most suitable products according to his/her necessities. In such cases, users can feel disappointed because they do not find what they really want among so huge amount of alternatives despite wasting much time. Different e-services have risen to help them to reach easy by quickly the products that meet their necessities. In this paper, we focus in recommender systems, a class of software<sup>5,21,22</sup> that has emerged in the last years within E-Commerce area<sup>23</sup>. Its aim is to assist users in their searches in order to find out the

most suitable item/s according to their preferences, necessities or tastes. To do so, the systems provide recommendations and/or hide or remove the useless information.

Essentially, all the recommender systems follow the same steps to make recommendations: first the systems gather preference information from users, experts, etc., related to their preferences, tastes, and opinions such that using this information they rank the items and make recommendations about which items are more attractive for them. Depending on the information gathered by the system, and the technique that ranks the items to suggest recommendations, the recommender systems can be classified in different types:

- *Demographic recommender systems*<sup>16</sup>: They classify users into demographic groups, using

personal attributes. A user will receive recommendations according to the group in which was classified.

- *Content-based recommender systems*<sup>20</sup>: This type of recommender systems compute recommendations according to the past user behaviour as well as to the features of items that the user liked before.
- *Collaborative filtering recommender systems*<sup>8</sup>: They gather ratings about products instead of item features and group the users according to their similarity. The recommendations are computed by means of a prediction about how much does a user like an item taking into account the other group members ratings.
- *Knowledge based recommender systems*<sup>4</sup>: These systems compute their recommendations using case based reasoning processes, i.e., the users provide an example similar to his/her aims and the systems infers a profile in order to find the better match product in the search space.
- *Utility based recommender systems*<sup>9</sup>: They compute recommendations based on the calculation of the utility of each item according with the user interests.
- *Hybrid recommender systems*<sup>1,5</sup>: This kind of recommender systems arise with the aim of solving drawbacks that the other systems present in some situations. In order to do that, these systems combine different techniques of the before recommender systems.

All these types of recommender systems have been developed and applied to different situations, being the most used the content-based and collaborative ones. However these systems are not always successful because they need big amount of users and items information to obtain good results and it is not always available. Therefore, different solutions have been proposed to avoid unsuccessful recommendations when there is not information available such as hybridization that can be useful in several cases but not always, or the use of knowledge based recommender systems, when there is not user

information available. In this paper we focus in the last type of recommender systems.

The information gathered by recommender systems is usually vague and incomplete because it is related to users' own perceptions. In spite of this fact, most of Recommender systems force their users to provide the information in a numerical scale fixed a priori<sup>10</sup>. This obligation implies a lack of expressiveness and hence a lack of precision in the suggested recommendations. To overcome this drawback, our proposal for a Knowledge Based Recommender System will offer the users the possibility of expressing their preference information using linguistic assessments instead of numerical ones, since the linguistic information is usually more suitable to assess qualitative information (human perceptions, taste, necessities)<sup>17,25</sup>. In addition, our system allows the users to use their own linguistic term set to express their preferences according to their knowledge about the items. Thus, the context on which the recommendations are computed is a multi-granular linguistic context<sup>11,13,14</sup>

In this paper we present a Knowledge based recommender system that will filter and recommend the closest items to the user's necessities computing the similarity among the descriptions of the items and the user profile inferred from the examples provided by the user according to their necessities. The system accomplishes the following steps to make the recommendations:

- 1 *Profiling process*: the user profile is an information structure that express the user preferences (necessities, tastes, etc). In this phase, the system infers the user profile from an example provided by the user.
- 2 *Recommendation process*: To find out the most suitable items for the users, the system will measure the similarity between the user profile and the items of the item database. The system ranks these items according to the similarity measure and recommends the most similar ones.

This paper is structured as follow. In section 2 we review some preliminaries about recommender systems and linguistic information that are useful to understand our proposal. In section 3 we present our linguistic multigranular knowledge based recommender system. In section 4 we show, by means

of an example, how this system works. Finally, in section 5 some conclusions are pointed out.

## 2. Preliminaries

Before introducing our proposal, we shall review, firstly, the drawbacks that classical recommender systems present. Secondly we shall make a brief review of knowledge base recommender systems. And finally we review the Fuzzy Linguistic Approach, which is needed in our proposal in order to model the vague and imprecise information provided by humans.

### 2.1. Drawbacks of Classical Recommender Systems

Recommender systems use <sup>5</sup>:

- (i) Background data: the information that the system has before the recommendation process begins,
- (ii) Input data: the information that user must communicate to the system in order to generate a recommendation, and
- (iii) An algorithm that combines background and input data to achieve the suggestions.

Classical recommender systems, both the Collaborative and Content-based ones, need a great amount of information about the users to exploit the background and the input data. Sometimes, this is an important drawback, mainly when a new user accesses to the system ('ramp-up' problem). When collaborative filtering recommender systems have not background data about the target user, it is not possible to compare him/her with other users in order to make recommendations. In a similar manner, content-based ones do not work properly in this situation because they are not able to build the user profile when background information about the user does not exist.

Also, we can see the 'ramp-up' problem when a new item is added in collaborative filtering system. In this case, the new item could be interesting to be recommended to some user, but the system does not use it because it has not any rating about the item yet. Therefore, this item will not be taken into account in the recommendation process until it has received enough votes. Due to the fact that it is not

being recommended, it is not likely that the item receives enough rating to be recommended.

Other problem we observed in collaborative recommenders is the 'gray sheep' problem. It occurs when a user falls on a border between existing groups of users. The target user is equally similar to two or more groups of users and, hence, the recommendation he/she will receive may be inaccurate.

The need of a large historical data set to ensure successful results is also an important requirement for both collaborative filtering and content-based recommender systems. These systems do not work properly when the historical data set is too small or sparse because, in this case, the probability of matching between the target user and other users will be low.

In order to overcome these drawbacks, some proposals have been given, such as hybrid recommender systems and Knowledge Based Recommender Systems.

In this paper we present a Knowledge Based Recommender Systems that deals with linguistic information. In the next subsection we shall explain the classical knowledge based recommender system in further details.

### 2.2. Knowledge Based Recommender Systems

Our interest in this paper is the Knowledge Based Recommender Systems <sup>4</sup>. These systems use case based reasoning <sup>15</sup> to make recommendations, i.e., they work starting with an example that the user points out, according with his/her tastes or necessities, from which the system infers a user profile utilized to find the best match products in the search space. For reaching its purpose, this system matches the user profile and the possible recommendations. So, when the user provides some knowledge about his/her needs or preferences, the recommender system is able to make recommendations using such matching process. This "user knowledge" can be any knowledge structure that allows to build a user profile. The simplest case could be that the user chooses, among all the available products, one of them that acts as an example of his/her necessities or tastes. These systems manage three types of knowledge:

- Catalog knowledge: It is the knowledge that

the Recommender System has about the products and their features. For example, a recommender system for new cars needs to know the characteristics of a car X: safety, comfort, price, velocity, fuel-consumption, etc.

- **Functional knowledge:** The system needs some knowledge to relate products to the user's necessities. For example, it is important to know that a need for a travelling salesman may be a car with low fuel-consumption and very high safety.
- **User's knowledge:** To provide good recommendations, the system needs to gather information about the user profile. For example, the user chooses a car which satisfies, more or less, his/her expectations. Then, the system will use this information to build an initial user profile consisted of the features of the chosen car.

A good example of this kind of recommender systems may be "The restaurant recommender Entree"<sup>6</sup>. This recommender system makes its recommendations by finding restaurants in a new city similar to restaurants the user knows and likes. The system allows users to navigate by stating their preferences with respect to a given restaurant, thereby refining their search criteria.

Knowledge based recommender systems are specially suitable for casual searching, when the information about the user does not exist or is scarce. Other systems have a period of start-up until the system gathers historical information about the user. The quality of the recommendations during this period is low. Knowledge based ones do not suffer this drawback because they do not need such kind of historical information. They work very well with only a small amount of knowledge about the user. In a typical situation, the user knowledge is stated by the user by means of an example of his/her necessities that he/she must point out. Starting with this example, the system builds a user profile and searches the products that best fit with this profile. So, we can enumerate the following advantages of this kind of systems:

- (i) They do not suffer ramp-up problems (both "new user" and "new item" problems).
- (ii) The "gray sheep" problem does not appear in these systems.
- (iii) They do not depend on large historical data set

On the other hand, these systems present two disadvantages related to the gathering of user knowledge:

- (i) When the amount of products is very large, the process of providing an example to express the user necessities may be a hard task. It may be a reason for the user to give up his/her search because he/she can not easily find a suitable example.
- (ii) It is possible that the user does not find an example that fits exactly his/her necessities. So, the system recommends him/her products that perhaps do not satisfy the user. So, it may be another reason for leaving the search too.

In section 3, we present a proposal of a Knowledge Based Recommender System to overcome these drawbacks.

### 2.3. Fuzzy Linguistic Approach

Usually, we work in a quantitative setting, in which the information is expressed by means of numerical values. However, many aspects of different activities in the real world cannot be assessed in a quantitative form, but rather in a qualitative one, i.e., with vague or imprecise knowledge. In that case, a better approach may be to use linguistic assessments instead of numerical values. The fuzzy linguistic approach represents qualitative aspects as linguistic values by means of linguistic variables<sup>24</sup>. This approach is adequate in some situations in which the information may be unquantifiable due to its nature, and thus, it may be stated only in linguistic terms.

In our proposal, the system deals with information that is related to user's tastes, preferences or qualitative features of the items such as *Safety* or *Comfort*. Although, this information could have been modelled by means of numerical values, we have considered using linguistic assessment since this information is vague, imprecise and will be better expressed with the Fuzzy Linguistic Approach.

\*<http://archive.ics.uci.edu/beta/datasets/Entree+Chicago+Recommendation+Data>

One possibility of generating the linguistic term set consists of directly supplying the term set by considering all terms distributed on a scale on which a total order is defined. For example, a set of seven terms  $S$ , could be given as follows:

$$\{s_0 : N, s_1 : VL, s_2 : L, s_3 : M, s_4 : H, \\ s_5 : VH, s_6 : P\}$$

In these cases, it is required that there exists:

- A negation operator  $Neg(s_i) = s_j$  such that  $j = g - i$  ( $g+1$  is the cardinality).
- A min and a max operator in the linguistic term set:  $s_i \leq s_j \Leftrightarrow i \leq j$ .

The semantics of the terms are given by fuzzy numbers defined in the  $[0, 1]$  interval. A way to characterize a fuzzy number is to use a representation based on parameters of its membership function<sup>2</sup>. The linguistic assessments given by the users are just approximate ones, some authors consider that the linear trapezoidal membership functions are good enough to capture the vagueness of those linguistic assessments.

This parametric representation is achieved by the 4-tuple  $(a, b, d, c)$ , in which  $b$  and  $d$  indicate the interval in which the membership value is 1, with  $a$  and  $c$  indicating the left and right limits of the definition domain of the trapezoidal membership function<sup>2</sup>. A particular case of this type of representation are the linguistic assessments whose membership functions are triangular, i.e.,  $b = d$ , so we represent this type of membership function by a 3-tuple  $(a, b, c)$ . An example may be the figure 1:

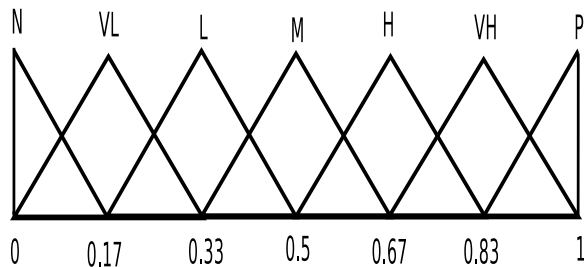


Fig. 1. A linguistic term set and its semantic

Other authors use a non-trapezoidal representation, e.g., Gaussian functions<sup>3</sup>.

## 2.4. Multigranular Linguistic Information

An important aspect of fuzzy linguistic approach is the granularity of the uncertainty, i.e., the level of discrimination among different counts of uncertainty. Therefore, according to the source of information knowledge it can be chosen different counts of uncertainty. The granularity should be small enough so as not to impose useless precision levels on the users but big enough to allow a discrimination of the assessments in a limited number of degrees<sup>13</sup>.

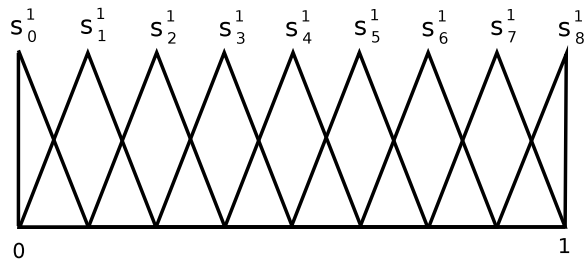
There are two reasons to utilize multigranularity in our proposal:

- 1 Users with different degrees of knowledge about the products. User with more knowledge can discriminate among more counts of uncertainty than other ones with less knowledge.
- 2 Different attributes can need different accuracy degrees. When the products are described in order to fit the product database, each feature can be valued using a different linguistic term set with different granularity.

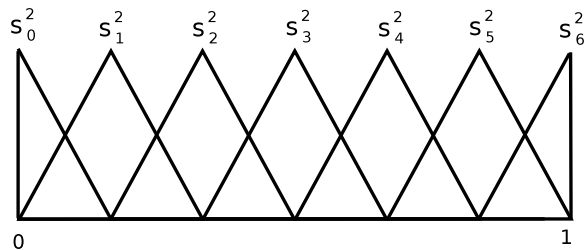
Typical values of cardinality used in the linguistic models are the odd ones, such as 7 or 9<sup>19</sup>, where the mid term represents an assessment of approximately 0.5, and with the rest of the terms being placed symmetrically around it. In this paper, we shall deal with sources of information with different degrees of knowledge, so each one could use different linguistic term sets with different granularity. We call this context as multi-granular linguistic context<sup>11</sup>.

Figures 2 and 3 show two examples of linguistic term sets with different granularity.

$$\begin{aligned} s_0^1 &= \textit{Extremely low} = (0, 0, .125) \\ s_1^1 &= \textit{Very low} = (0, .125, .25) \\ s_2^1 &= \textit{Low} = (.125, .25, .375) \\ s_3^1 &= \textit{A bit low} = (.25, .375, .5) \\ s_4^1 &= \textit{Average} = (.375, .5, .625) \\ s_5^1 &= \textit{A bit high} = (.5, .625, .75) \\ s_6^1 &= \textit{High} = (.625, .75, .875) \\ s_7^1 &= \textit{Very high} = (.75, .875, 1) \\ s_8^1 &= \textit{Extremely high} = (.875, 1, 1) \end{aligned}$$

Fig. 2. The linguistic term set  $S_1$ 

$$\begin{aligned}
 s_0^2 &= \textit{Extremely low} = (0, 0, .16) \\
 s_1^2 &= \textit{Very low} = (0, .16, .33) \\
 s_2^2 &= \textit{Low} = (.16, .33, .5) \\
 s_3^2 &= \textit{Average} = (.33, .5, .66) \\
 s_4^2 &= \textit{High} = (.5, .66, .83) \\
 s_5^2 &= \textit{Very high} = (.66, .83, 1) \\
 s_6^2 &= \textit{Extremely high} = (.83, 1, 1)
 \end{aligned}$$

Fig. 3. The linguistic term set  $S_2$ 

### 3. A Knowledge Based Recommender System Model with multigranular linguistic information

In this section, we present our proposal for a Knowledge Based Recommender System. This model expects users to provide an example of their preferences (for example, Bella Italia restaurant) in order to generate an initial user profile. So, the initial profile match exactly with the description of the selected item. This profile consists of a vector of features in which each feature is described by a *linguistic label*. Each feature describes a different aspect of the user profile, and therefore, it could be assessed with a different linguistic term set according to the available knowledge about this feature.

Sometimes, the given example does not fix exactly what the user wants since one or several features that have been assessed with linguistic terms that do not fulfil user's expectations or needs. So, the user needs to refine his/her profile by changing some of its linguistic assessments. Suppose he/she agrees

with all the features of the given example except one. For example, considering the price as the feature in disagree, next, the user can change the value "low" with the value "very low". In such cases, sometimes it could be suitable to offer the user another linguistic term set closer to him/her knowledge than the one used in the descriptions of the items. With the changes provided by the user, the system will generate the final profile that will be used in the recommendation process.

Therefore, our proposal develops its activity according to the schema (see figure 4).

- 1 *Profiling process*: The system builds the user profile which contains information concerning the necessities of the user. This phase has two steps:

- (a) *Gathering the preferred example from the user*: The user chooses an item as an example of his/her necessities. The description of this item will define the initial user profile.
- (b) *Casual modification of preferences*: Usually the user does not search an item exactly equal to the given example, but a similar one, with some differences in its attributes. So, in such cases, the user must refine his/her profile by using a set of linguistic terms adequate to his/her knowledge level.

- 2 *Recommendation process*: The system calculates the similarity between the user profile and the items, and recommends the most suitable ones. This process is composed of the following steps:

- (a) *Unification of the linguistic information*: Due to the fact we are dealing with multigranular linguistic information, it is necessary to unify it in an unique domain called Basic Linguistic Term Set (BLTS).
- (b) *Calculation of the similarity between the user profile and the items*: In order to recommend an item to the user, we need to know how close the items are to the user profile.
- (c) *Providing the recommendation*: This is the final step in which the closest items

to the user necessities will be recommended.

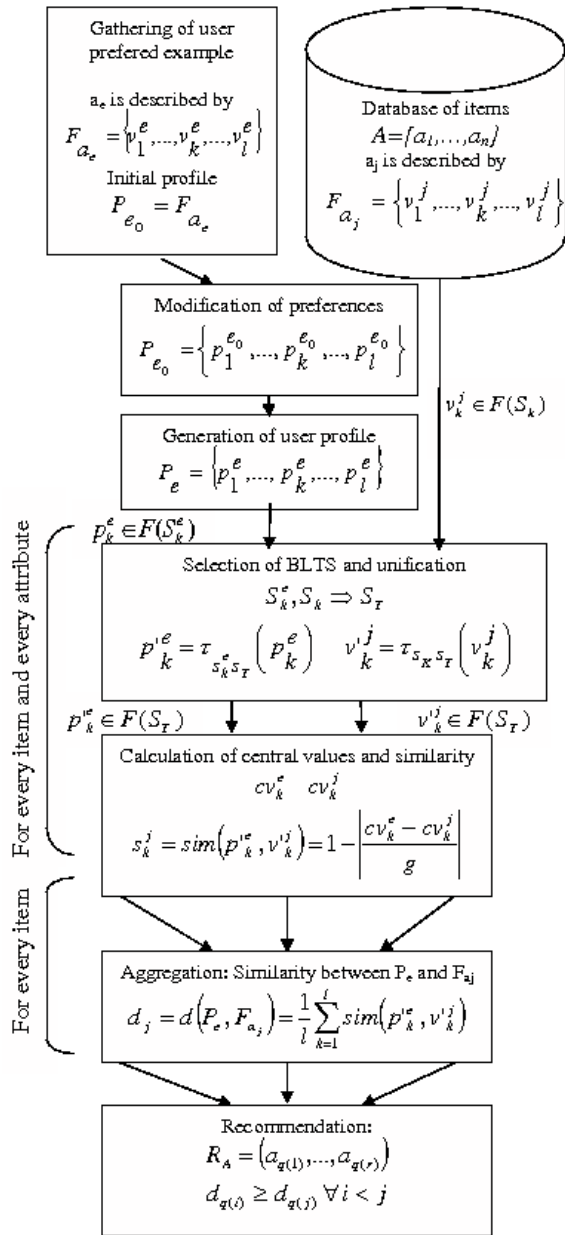


Fig. 4. Recommender System Model

In the following sections we will explain these steps in further detail.

### 3.1. Profiling process

In this phase, the system gathers the user's necessities or preferences in order to know what kind of

item is required by the user  $u_e$ . The Recommender System has a database  $A = \{a_1, \dots, a_n\}$ , with  $n$  items, all of them described by a set of attributes  $\{c_1, \dots, c_l\}$ . Therefore, every item  $a_j$  is described by an utility vector  $F_{a_j} = \{v_1^j, \dots, v_l^j\}$ , in which  $v_k^j$  is the value of the attribute  $c_k$  for the item  $a_j$  assessed in the linguistic term set  $S_k$ . These descriptions are obtained either directly from experts or from surveys realized about the items. In any case, the information about the attributes can be assessed in a multigranular context that allows to use different linguistic term sets. So, every attribute can be assessed in a different label set according to the existing degree of the knowledge about them. Therefore, the description of the items stored in the database conforms a multigranular linguistic space. Once we know how the items are described in the Recommender System, we can study how to build the user profile.

#### 3.1.1. Gathering the preferred example from the user

In our Recommender System, the starting point to define the user necessities is the selection of an example. Let  $a_e$  be the item given as an example by the user  $u_e$ . This item is described in the database by means of an utility vector  $F_e = \{v_1^e, \dots, v_l^e\}$ , in which  $v_k^e \in S_k$  is an assessment for attribute  $c_k$  expressed in terms of  $S_k$ . This selected example defines a initial user profile that we denote as  $P_{e_0} = \{p_1^{e_0}, \dots, p_l^{e_0}\}$ , where  $p_k^{e_0} = v_k^e$ . In this initial user profile, the linguistic terms sets are the same than the ones used in the system database.

#### 3.1.2. Casual modification of preferences

Once the initial user profile is defined, the system offers the user the possibility of changing one or more values of his/her profile in order to refine the recommendation process. Probably, the knowledge the user has about a given attribute  $c_k$  is different to the knowledge that the experts (database's builders) has about the same attribute. So, the linguistic term set used by the experts may not be appropriate for the user. Therefore, the system allows the user to utilize other linguistic term set more suitable to his/her knowledge about the attribute. In this case, for an

attribute  $c_k$ , the user can assign a new value,  $p_k^{e1}$ , expressed in other linguistic term set,  $S'_k$  according to his/her knowledge. Then, we have a final user profile  $P_e = \{p_1^e, \dots, p_l^e\}$  where  $p_k^e \in S_k^e$  are obtained in the following way:

- (a)  $p_k^e = p_k^{e0}, p_k^{e0} \in S_k^e = S_k$  if the attribute  $c_k$  has not been modified
- (b)  $p_k^e = p_k^{e1}, p_k^{e1} \in S_k^e = S'_k$  otherwise.

In order to make easy this task, the system will provide an easy-use interface that allows to select a suitable the linguistic term set for the attribute to change.

### 3.2. Recommendation process

In this phase, the system computes how close the items are to the user profile by means of measure of resemblance or similarity. To accomplish this phase the system will evaluate the similarity between all the items of the database  $A = \{a_1, \dots, a_n\}$  and the user profile following these steps:

- 1 *Unify the linguistic information:* because there is no way to deal directly with information that has been assessed in different linguistic term sets, we need to unify the information in a unique domain.
- 2 *Calculate the similarity between every item and the user profile:* they system computes the similarity degree between the user profile and database items.
- 3 *Providing the recommendations:* finally, the system suggests to the user the most suitable items, i.e., the closest ones to the user profile.

#### 3.2.1. Unification of the linguistic information

In order to manage multigranular information, we must unify it using a unique expression domain<sup>12,13,18</sup>. In this case, we choose as unification domain a Basic Linguistic Term Set (BLTS) that we note as  $S_T$ . The information will be unified by means of fuzzy sets defined in the BLTS,  $F(S_T)$ . The selection of the BLTS is explained in<sup>12</sup>.

Following, the system unifies the multi-granular linguistic terms by means of fuzzy sets defined in the BLTS,  $F(S_T)$ , using the following transformation function:

**Definition 1**<sup>12</sup> Let  $A = \{l_0, \dots, l_p\}$  and  $S_T = \{s_0, \dots, s_g\}$  be two sets of linguistic terms such that  $g \geq p$ . Then, a function of multigranular transformation  $\tau_{AS_T}$ , is defined as:

$$\begin{aligned} \tau_{AS_T} : A &\rightarrow F(S_T) \\ \tau_{AS_T}(l_i) &= \{(s_k, \alpha_k^i) \mid k \in \{0, \dots, g\}\}, \forall l_i \in A \\ \alpha_k^i &= \max_y \min \{\mu_{l_i}(y), \mu_{s_k}(y)\} \end{aligned}$$

where  $F(S_T)$  is the set of all the fuzzy sets defined on  $S_T$ , and  $\mu_{l_i}(y)$  and  $\mu_{s_k}(y)$  are membership functions of the fuzzy sets associated to the terms  $l_i$  and  $s_k$  respectively.

To unify the multigranular linguistic information the system will use the transformation functions  $\tau_{S_k S_T}$  in order to express the user profile and the descriptions of the items by means of fuzzy sets defined into fuzzy sets in the BLTS. For instance, an assessment of the user profile,  $p_k^e \in S_k^e$ , is transformed into a fuzzy set,  $p_k^e$ , in which this fuzzy set is described by a tuple of membership degrees  $(\alpha_{k0}^e, \dots, \alpha_{kg}^e)$ .

In the same manner, the descriptions of the items are also transformed into the BLTS. An assessment,  $v_k^j \in S_k$ , of the item,  $a_j$ , is transformed into a fuzzy set,  $v_k^j$ , and it is also represented in the same way  $(\alpha_{k0}^j, \dots, \alpha_{kg}^j)$ .

Once all the information is expressed in the same expression domain, we can proceed to calculate the similarity between the user profile and item database.

#### 3.2.2. Calculation of the similarity between the user profile and the items

Once the information has been unified, the system will look for which items are closer to the user's necessities. To accomplish this step, we need to calculate the similarity between the user profile,  $P_e$ , and the item,  $a_j$ , of the database using the following function:

$$d_j = d(P_e, a_j) = \frac{1}{l} \sum_{k=1}^l w_i \text{sim}(p_k^e, v_k^j)$$

where  $w_i$  represents the importance of each attribute and  $\sum w_i = 1$ , being  $\text{sim}$  a function that computes the similarity between the values  $P_e$  and  $a_j$



Although initially, we have considered this function, *sim*, to be accomplished by using the Euclidean distance. However, it was discarded because its results are not good (see <sup>14</sup>).

Then, we propose to compute the similarity using a measure based on the use of central values <sup>14</sup> that obtain suitable results according to our aim in this phase of the recommendation process.

**Definition 2** <sup>14</sup> Giving a fuzzy set  $b' = (\alpha_1, \dots, \alpha_g)$  defined on  $S = \{s_h\}$  for  $h = 0, \dots, g$ , we obtain its central value *cv* in the following way:

$$cv = \frac{\sum_{h=0}^g idx(s_h) \alpha_h}{\sum_{h=0}^g \alpha_h}, \text{ where } idx(s_h) = h$$

This value represents the central position or centre of gravity of the information contained in the fuzzy set  $b'$ . The range of this central value is the closed interval  $[0, g]$

Therefore, from this definition, it is defined the following similarity function *sim*:

**Definition 3** <sup>14</sup> Let  $b'_1$  and  $b'_2$  be two fuzzy sets defined on the BLTS,  $S_T = \{s_0, \dots, s_g\}$ , and let  $cv_1$  and  $cv_2$  be the central values of  $b'_1$  and  $b'_2$  respectively, then the similarity between them is calculated as:

$$sim(b'_1, b'_2) = 1 - \left| \frac{cv_1 - cv_2}{g} \right|$$

The final result of this step is a similarity vector  $D = (d_1, \dots, d_n)$  in which the system will keep the similarity between the user profile  $P_e$  and all the items in the database.

### 3.2.3. Recommendation

Here, the system will rank the items according to the similarity values of the vector  $D = (d_1, \dots, d_n)$ . The best ones will be those that are closer the user profile, i.e., those with the greatest score in the similarity. The system will recommend the top-N items that reach a given threshold <sup>7</sup>, i.e., if one of the top-N items is too far from the user profile (its similarity degree is less than the threshold) then this item will not be included in the recommendation.

Let the item set  $A = \{a_1, \dots, a_n\}$ ,  $r$  the maximum number of items to be recommended and  $h$  the

threshold to be reached. Then, the recommendation to the user is given by the recommendation vector,  $R_A$ , where the first element is the top one recommended item, the second is the second closest to the user profile and so on:

$$R_A = (a_{q(1)}, \dots, a_{q(r_1)}) \text{ where } r_1 \leq r$$

where the function  $q$  is defined in the following way:

$$\begin{aligned} q: \{1, 2, \dots, r_1\} &\rightarrow \{1, 2, \dots, n\} \\ q(i) &\neq q(j) \quad \forall i \neq j \\ q(i) &\neq e \quad \forall i = 1, \dots, r_1 \\ &\text{being } a_e \text{ the example given by the user} \\ d_{q(i)} &\geq d_{q(j)} \quad \forall i < j \\ d_{q(i)} &\geq h \quad \forall i = 1, \dots, r_1 \end{aligned}$$

## 4. Example

We shall show an easy example in order to clarify our proposal. Let us suppose we are looking for a new car. The system has a database with the cars that could be recommended. They are described using a set of features. To simplify, in this example we have six items,  $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$  where each item corresponds to a car which is described by a set of four features  $C = \{c_1, c_2, c_3, c_4\}$ , where  $c_1 = Price$ ,  $c_2 = Velocity$ ,  $c_3 = Safety$  and  $c_4 = Comfort$ . In real systems we could find thousand or millions items stored in the database and dozens of attributes are used to describe them.

To describe the attributes we could use different domains, i.e. different linguistic term sets. Here we have used the linguistic term set  $S_{1,2}$  (see figure 5) for the attributes  $c_1$  and  $c_2$ , and the linguistic term set  $S_{3,4}$  (see figure 6) for  $c_3$  and  $c_4$ . These sets are defined by the following membership functions:

$$\begin{aligned} s_0^{1,2} &= \text{Extremely low} = (0, 0, .125) \\ s_1^{1,2} &= \text{Very low} = (0, .125, .25) \\ s_2^{1,2} &= \text{Low} = (.125, .25, .375) \\ s_3^{1,2} &= \text{A bit low} = (.25, .375, .5) \\ s_4^{1,2} &= \text{Average} = (.375, .5, .625) \\ s_5^{1,2} &= \text{A bit high} = (.5, .625, .75) \\ s_6^{1,2} &= \text{High} = (.625, .75, .875) \\ s_7^{1,2} &= \text{Very high} = (.75, .875, 1) \\ s_8^{1,2} &= \text{Extremely high} = (0.875, 1, 1) \end{aligned}$$

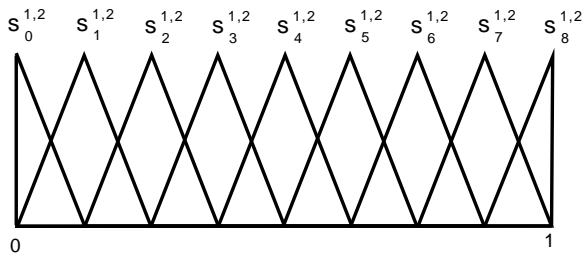


Fig. 5. The linguistic term set  $S_{1,2}$

- $s_0^{3,4} = \text{Negligible} = (0, 0, .16)$
- $s_1^{3,4} = \text{Very inferior} = (0, .16, .33)$
- $s_2^{3,4} = \text{Inferior} = (.16, .33, .5)$
- $s_3^{3,4} = \text{Average} = (.33, .5, .66)$
- $s_4^{3,4} = \text{Superior} = (.5, .66, .83)$
- $s_5^{3,4} = \text{Very superior} = (.66, .83, 1)$
- $s_6^{3,4} = \text{Outstanding} = (.83, 1, 1)$

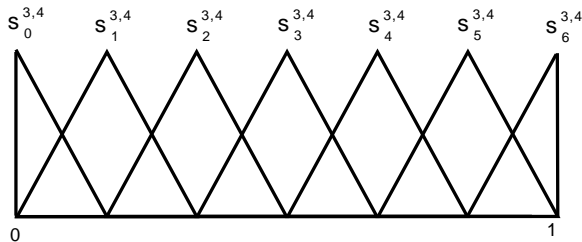


Fig. 6. The linguistic term set  $S_{3,4}$

The descriptions of the items, using these linguistic term sets, can be seen in the table 1.

Table 1. Item database

	$c_1$	$c_2$	$c_3$	$c_4$
$a_1$	$s_0^{1,2}$	$s_3^{1,2}$	$s_3^{3,4}$	$s_3^{3,4}$
$a_2$	$s_1^{1,2}$	$s_2^{1,2}$	$s_1^{3,4}$	$s_4^{3,4}$
$a_3$	$s_7^{1,2}$	$s_4^{1,2}$	$s_0^{3,4}$	$s_5^{3,4}$
$a_4$	$s_5^{1,2}$	$s_1^{1,2}$	$s_2^{3,4}$	$s_3^{3,4}$
$a_5$	$s_1^{1,2}$	$s_6^{1,2}$	$s_3^{3,4}$	$s_6^{3,4}$
$a_6$	$s_8^{1,2}$	$s_0^{1,2}$	$s_4^{3,4}$	$s_6^{3,4}$
	$s_1^{1,2}$	$s_8^{1,2}$	$s_3^{3,4}$	$s_1^{3,4}$

A user,  $u_e$ , wants to receive a recommendation from our system. Firstly, the system must build the user profile:

- 1 *Gathering the preferred example from the user:* the user states an item  $a_1$ , close to what he/she needs. So, this item is the example selected by the user and the initial user profile

will be:

$$P_{e_0} = \{s_0^{1,2}, s_3^{1,2}, s_3^{3,4}, s_2^{3,4}\}$$

- 2 *Casual modification of preferences:* Nevertheless, the user realizes that the attribute  $c_1$  (cars price) of the selected item does not represent what he/she wants. Due to this fact, he/she wants to provide a new value, however, the linguistic term set used to describe  $c_1$  is not suitable for him/her knowledge. He/she doesn't need to discriminate between nine linguistic terms to express his/her necessity about this feature. So he/she would rather use a smaller set. He/she decides to use a linguistic term set with only five elements,  $S_1^{1e}$  that is defined below (see figure 7:)

- $s_0^{1e} = \text{Very low} = (0, 0, .25, )$
- $s_1^{1e} = \text{Low} = (0, .25, .5)$
- $s_2^{1e} = \text{Average} = (.25, .5, .75)$
- $s_3^{1e} = \text{High} = (0.5, 0.75, 1)$
- $s_4^{1e} = \text{Very high} = (.75, 1, 1)$

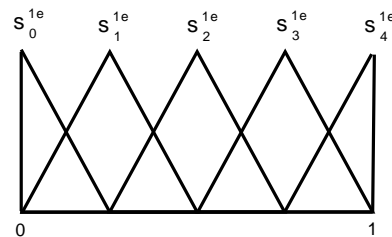


Fig. 7. The linguistic term set  $S_{1e}$

The user assesses this attribute with the value  $s_1^{1e}$  and so, now, the user profile is:

$$P_e = \{s_1^{1e}, s_3^{1,2}, s_3^{3,4}, s_2^{3,4}\}$$

Once the the user profile is available, the system accomplishes the recommendation process:

- 1 *Unification of the linguistic information:* According to <sup>11</sup> the system chooses  $S_{1,2}$  as BLTS ( $S_T$ ), being its higher granularity the main criterion to make such decision, and transforms the user profile and item database into  $S_T$ . So, the item database (see Table 2) and the user

Table 2: Item database expressed into the BLTS

	$c_1$	$c_2$	$c_3$	$c_4$
$a_1$	(1,0,0,0,0,0,0,0)	(0,0,0,1,0,0,0,0)	(0,0,.14,.57,1,.57,.14,0,0)	(.28,.71,.85,.42,0,0,0,0)
$a_2$	(0,0,0,0,0,1,0,0)	(0,0,1,0,0,0,0,0)	(.42,.85,.71,.28,0,0,0,0)	(0,0,0,0,.42,.85,.71,.28,0)
$a_3$	(0,0,0,0,0,0,0,1)	(0,0,0,0,1,0,0,0)	(1,.57,.14,0,0,0,0,0)	(0,0,0,0,0,.28,.71,.85,.42)
$a_4$	(0,0,0,0,0,1,0,0)	(0,0,0,0,0,0,1,0)	(0,.28,.71,.85,.42,0,0,0)	(0,0,0,0,0,0,.14,.57,1)
$a_5$	(0,0,0,0,0,0,0,1)	(1,0,0,0,0,0,0,0)	(0,0,0,0,.42,.85,.71,.28,0)	(0,0,0,0,0,0,.14,.57,1)
$a_6$	(0,1,0,0,0,0,0,0)	(0,0,0,0,0,0,0,1)	(0,0,.14,.57,.1,.57,.14,0,0)	(.42,.85,.71,.28,0,0,0,0)

profile will be expressed in terms that can be compared one each other.

$$P_e = \{(0,0,.33,.66,1,.66,.33,0,0), \\ (0,0,0,1,0,0,0,0,0), \\ (0,0,.14,.57,1,.57,.14,0,0), \\ (0,.28,.71,.85,.42,0,0,0,0)\}$$

2 *Calculation of the similarity between the user profile and the items:* In order to obtain this measurement the system calculates the central values of the fuzzy sets of every item in the database (see table 3) and the fuzzy sets of the user profile:

$$P_e^{CV} = \{4,3,4,2.62\}$$

Table 3. Central values of the item database

	$c_1$	$c_2$	$c_3$	$c_4$
$a_1$	0	3	4	2.62
$a_2$	5	2	1.37	5.37
$a_3$	7	4	1.19	7.05
$a_4$	5	6	2.62	7.5
$a_5$	8	0	5.37	7.5
$a_6$	1	8	4	1.37

Finally, the system computes the similarity between the user profile and each item of the the database using the similarity function presented in 3 (see table 4) considering that all the features have the same importance.

Table 4. Similarity between the user profile and the items

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
0.87	0.77	0.64	0.67	0.58	0.71

3 *Providing the recommendations:* This is the final step of the recommendation process. The system will sort out the items according to the similarity to the user profile and obtains

$$R_A = (a_1, a_2, a_6, a_4, a_3, a_5)$$

The first item,  $a_1$ , cannot be recommended since it was chosen as an example of the user's necessities. Let's suppose that the system recommends the two items closest to the user profile, therefore the final recommendations will be:

$$\{a_2, a_6\}$$

## 5. Conclusions

Recommender Systems support users to find the most suitable products in e-shops within a huge amount of products according to their necessities and preferences. There exists different types of Recommendations Systems, as the Content-based or the Collaborative Recommendation Systems, that provide good recommendations but they present some problems, overall, related to the amount of information necessary to make recommendations. Hybrid and Knowledge-based recommender systems face these problems from different points of view.

In this contribution we have presented a Knowledge Based Recommender System that deals with multigranular linguistic information instead of numerical values. The advantage of this representation is that we are able to gather the user's information, that is usually related to perceptions or tastes, without losing expressiveness or accuracy. Moreover, we have defined a flexible model to deal with the information in which each attribute can be assessed with the most suitable linguistic term set and the users can use linguistic term sets according to their knowledge or preferences.

Besides, we have proposed the use of a similarity measure based on central values of fuzzy sets which is able to compute the similarity between items. In this manner, we haven't just taken into account if the assessments are the same, but also we have computed how different are they from each other.

## 6. Acknowledgements

This work is partially supported by the Research Projects TIN-2006-02121, 2007/BA020 and FEDER funds

## References

1. 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.
2. P.P. Bonissone and K.S. Decker. *Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-Off Precision and Complexity*, chapter Uncertainty in Artificial Intelligence, pages 217–247. North-Holland, 1986.
3. G. Bordogna and G. Passi. A fuzzy linguistic approach generalizing boolean information retrieval: A model and its evaluation. *Journal of the American Society for Information Science*, (44):70–82, 1993.
4. R. Burke. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69(32), 2000.
5. R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
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. Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.
8. 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.
9. Robert H. Guttman. Merchant differentiation through integrative negotiation in agent-mediated electronic commerce. *Master's Thesis, School of Architecture and Planning, Program in Media Arts and Sciences, Massachusetts Institute of Technology*, 1998.
10. C. Hayes and P. Cunningham. Smart radio - community based music radio. *Knowledge-Based Systems*, 14(3-4):197–201, 2001.
11. F. Herrera and E. Herrera-Viedma. Linguistic decision analysis: Steps for solving decision problems under linguistic information. *Fuzzy Sets and Systems*, (115):67–82, 2000.
12. F. Herrera, E. Herrera-Viedma, and L. Martínez. A fusion approach for managing multi-granularity linguistic term sets in decision making. *Fuzzy Sets and Systems*, (114):43–58, 2000.
13. E. Herrera-Viedma, L. Martínez, F. Mata, and F. Chiclana. A consensus support system model for group decision-making problems with multi-granular linguistic preference relations. *IEEE Transactions on Fuzzy Systems*, 13(5):644–658, 2005.
14. E. Herrera-Viedma, F. Mata, L. Martínez, F. Chiclana, and L.G. Pérez. Measurements of consensus in multi-granular linguistic group decision making. *Modeling Decisions for Artificial Intelligence, Proceedings Lecture Notes in Computer Science*, (3131):194–204, 2004.
15. J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.
16. B. Krulwich. Lifestyle finder: intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.
17. E. Levrat, A. Voisin, S. Bombardier, and J. Bremont. Subjective evaluation of car seat comfort with fuzzy set techniques. *International Journal of Intelligent Systems*, (12):891–913, 1997.
18. L. Martínez, J. Liu, J.B. Yang, and F. Herrera. A multi-granular hierarchical linguistic model for design evaluation based on safety and cost analysis. *International Journal of Intelligent Systems*, 20(12):1161–1194, 2005.
19. G. Miller. The magical number seven or minus two: Some limits on our capacity of processing information. *Psychological Review*, (63):81–97, 1956.
20. M. J. Pazzani, J. Muramatsu, and D. Billsus. Syskill webert: Identifying interesting web sites. In *AAAI/IAAI, Vol. 1*, pages 54–61, 1996.
21. Saverio Perugini, Marcos Andre Goncalves, and Edward A. Fox. Recommender system research: A connection-centric survey. *Journal of Intelligent Information Systems*, 23(2):107–143, 2004.
22. P. Resnick and H.R. Varian. Recommender systems. *Association for Computing Machinery. Communications of the ACM.*, 40(3):56, Mar 1997.
23. J.B. Schafer, J.A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, (5):115–153, 2001.
24. L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning-i, ii, iii. *Information Sciences*, 8-9:199–249, 301–357, 43–80, 1975.
25. L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996.