Proceedings of the 2007 International Conference on Intelligent Systems and Knowledge Engineering, Chengdu, 15-16 October 2007.

A Knowledge Based Recommender System with Multigranular Linguistic Information

L. Martínez, M.J. Barranco, L.G. Pérez, M. Espinilla, F. Siles

Department of Computer Science, University of Jaén, Jaén 23071, Spain

Abstract

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 to satisfy the necessities of a great variety of users. Nevertheless, searching in such a wide range of items could be a very difficult and tedious task. Recommender Systems assist users to find items by means of recommendations based on information provided from different sources such as: other users, experts, etc. Most of the recommender systems force users to provide their preferences or necessities using an unique numerical scale of information fixed in advance. Normally, this information is usually related to opinions, tastes and perceptions, and therefore, it means that it is usually better expressed in a qualitative way, with linguistic terms, than in a quantitative way, with precise numbers. In this contribution, we propose a Knowledge Based Recommender System that uses the fuzzy linguistic approach to define a flexible framework that captures the uncertainty of the user's preferences. Thus, this framework will allow users to express their necessities in a different scale, closer to their knowledge, from the scale used to describe the items.

Keywords: E-Commerce, E-Services, Recommender Systems, Fuzzy Linguistic Approach, Multigranular Linguistic Information

1. Introduction

One of the main problems users face surfing in Internet is the vast quantity of information they find, being most of it useless. For instance, in the e-commerce area, some e-shop websites offer all the items that are related to users' queries even if they do not meet their real expectations. In such cases, users can feel disappointed because they do not find what they want among so huge amount of alternatives or because they waste a lot of time to find it. Different e-services have risen to help them to reach easily and quickly their necessities. In this paper, we focus in Recommender Systems, a class of software [18] that has emerged in the last years within E-Commerce area [19]. Their aim is to assists users to find out the most suitable items according to their preferences, necessities or tastes, hiding or removing the useless information.

These systems gather preference information from users, experts, etc., related to their preferences, tastes, and opinions about a given set of items (books, music, etc.) in such a way that using this information they rank the items and make recommendations about which items are the most attractive for them. The techniques utilized to achieve this aim are different from each other, both in the required information and in the necessary processes to compute the recommendations. Depending on these techniques, we can classify the Recommender Systems in Demographic Recommender Systems [13, 17], Content-based Recommender Systems [16, 15], Collaborative Filtering Recommender Systems [6, 7], Knowledge Based Recommender Systems [4, 20] and Hybrid Recommender Systems [1, 5].

The information provided by the users to these systems is usually vague and incomplete because it is related to customers' 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 [8] . This obligation implies a lack of expressiveness and hence a lack of precision in the suggested recommendations. Our proposal for a Knowledge Based Recommender System will offer the users the posibility 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) [14, 22]. In addition, our model 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. To deal with linguistic information we shall use the fuzzy linguistic approach [21] and to rank the items we shall use a similarity measurement [11].

In this contribution, we present a Knowledge based recommender system that will filter and recommend the closest items to the user's necessities by measuring the similarity among the descriptions of the items and the user profile. The system accomplishes the following steps to make the recommendations:

1. *Gathering the user profile:* the user profile is an information structure that gathers the information provided by the user about his/her necessities, tastes, etc.
2. *Calculation of the similarity between the user profile and the items:* To find out the most suitable items for the customers, the model will measure the distance between the user profile and the items of the item database.
3. *Making a recommendation:* To recommend the most suitable items for a customer, the system ranks these items by means of their similarity to the user profile. The most suitable items will be closer to the user profile than the less suitable ones.

This contribution is structured as follows. In section 2 we shall make a brief review of the fuzzy linguistic approach. In section 3 we present our multigranular knowledge based recommender model. In section 4 we show, by means of an example, how this model works. Finally, in section 5 some conclusions are point out.

2. 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 [21]. 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.

We have to choose the appropriate linguistic descriptors for the term set and their semantics. In order to accomplish this objective, an important aspect to analyze 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 choose different counts of uncertainty. Typical values of cardinality used in the linguistic models are the odd ones, such as 7 or 9, 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 contribution, 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 multigranular linguistic context [9].

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 [2]. 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 [2]. 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:

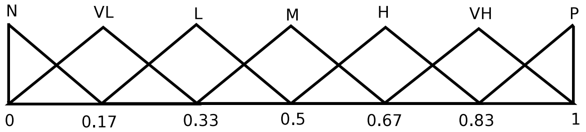


Fig. 1: A linguistic term set and its semantic.

Other authors use a non-trapezoidal representation, e.g., Gaussian functions [3].

3. A knowledge based recommender systems model with multigranular linguistic information

In this section, we present our proposal for a Knowledge Based Recommender System. This system expects users to provide an example of his/her preferences (for example, Bella Italia restaurant) in order to be able to define an initial user profile. This profile consists in a vector of features in which each feature is described by a linguistic label from a linguistic term set. 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 nature of this aspect.

Sometimes, the given example does not represent exactly what the user wants and the user needs to refine his/her profile by changing some of his/her assessments (for example, considering the price, the user can change the value "low" with the value "very low"). In such cases, it would be more suitable to use different linguistic term sets closer to the user's knowledge than the linguistic term sets used in the descriptions of the items. With these changes provided by the user, the system will define the final profile that will be used in the recommendation process.

This model develops its activity according to the schema of the figure 2.

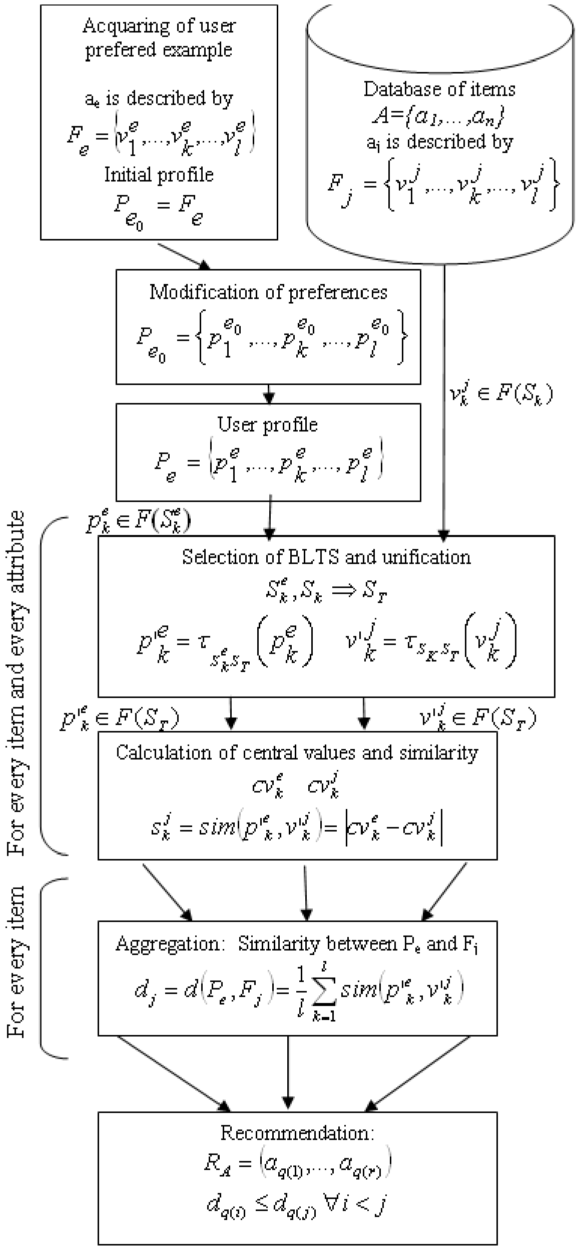
1. *Gathering the user profile:* The system builds the user profile which contains information concerning the necessities of the user. This phase has two steps:

   a) *Acquiring a prefered 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. *Calculation of the similarity between the user profile and the items:* The system calculates the satisfaction degree concerning the user necessities for each database item.

   a) *Unification of the linguistic information:* Due to the fact we are dealing with multigranular linguistic information, it is necessary to unify



Acquaring of user prefered example
a_e is described by
$F_e = \{v_1^e, \ldots, v_k^e, \ldots, v_l^e\}$
Initial profile
$P_{e_0} = F_e$
Database of items
$A=\{a_1,\ldots,a_n\}$
a_j is described by
$F_j = \{v_1^j, \ldots, v_k^j, \ldots, v_l^j\}$
Modification of preferences
$P_{e_0} = \{p_1^{e_0}, \ldots, p_k^{e_0}, \ldots, p_l^{e_0}\}$
$v_k^j \in F(S_k)$
User profile
$P_e = \{p_1^e, \ldots, p_k^e, \ldots, p_l^e\}$
$p_k^e \in F(S_k^e)$
Selection of BLTS and unification
$S_k^e, S_k \Rightarrow S_T$
$p'^e_k = \tau_{S_k^e S_T}(p_k^e)$ $v'^j_k = \tau_{S_k S_T}(v_k^j)$
$p'^e_k \in F(S_T)$
$v'^j_k \in F(S_T)$
Calculation of central values and similarity
cv_k^e cv_k^j
$s_k^j = sim(p'^e_k, v'^j_k) = |cv_k^e - cv_k^j|$
For every item and every attribute
For every item
Aggregation: Similarity between P_e and F_j
$d_j = d(P_e, F_j) = \frac{1}{l}\sum_{k=1}^{l} sim(p'^e_k, v'^j_k)$
Recommendation:
$R_A = (a_{q(1)}, \ldots, a_{q(r)})$
$d_{q(i)} \le d_{q(j)}\ \forall i < j$

Fig. 2: Recommender system model.

it in an unique domain called Basic Linguistic Term Set (BLTS). Each linguistic label is transformed into a fuzzy set defined in such 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 item is to the user profile. To accomplish this calculation the system will compute a similarity degree between the user profile and every item of the item database.

3. *Recommendation:* This is the final phase in which the closest items to the user necessities will be recommended

In the following sections we will explain this steps in detail.

3.1. Gathering the user profile

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 means of a set of attributes $C = \{c_1, \dots, c_l\}$. Therefore, every item a_j is described by an utility vector $F_j = \left\{v_1^j, \dots, v_l^j\right\}$, 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. Every attribute can be assessed using different label sets depending on the knowledge available for them. Once we know how the items are described in the Recommender System, we can study how to build the user profile

3.1.1. *Acquiring a prefered example from the user*

In this kind of Recommender Systems, 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 as the ones used in the system database.

3.1.2. *Casual modification of preferences*

Once the initial user profile is defined, we offer the user the possibility of changing one or more values of his/her profile in order to refine the recommendation process. In this case, for an attribute c_k, the user can assign a new value, $p_k^{e_1}$, expressed in other linguistic terms 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) $p_k^e = p_k^{e_0}, p_k^e \in S_k^e = S_k$ if the attribute c_k has not been modified

b) $p_k^e = p_k^{e_1}, p_k^e \in S_k^e = S'_k$ otherwise.

3.2. Calculation of the similarity between the user profile and the items

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 distance 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 distance between every item and the user profile:* now, they system can evaluate the similarity degree between the user profile and database items.

3.2.1. *Unification of the linguistic information*

In order to manage multigranular information, we must unify it using a unique expression domain [10]. In this case, we choose as unification domain a Basic Linguistic Term Set (BLTS) that we note S_T. The information will be unified by means of fuzzy sets defined in the BLTS, $F(S_T)$. The chosen BLTS must meet the conditions stated in [10].

Following, we must express the linguistic terms of the different sets by means of fuzzy sets defined on the BLTS, $F(S_T)$, using the following transformation function:

Definition 3.1. *[10] 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 τ_{AS_T}, is defined as:*

$$\begin{array}{l} \tau_{AS_T} : A \rightarrow F(S_T) \\ \tau_{AS_T}(l_i) = \left\{ \left(s_k, \alpha_k^i\right) | k \in \{0, \dots, g\} \right\}, \forall l_i \in A \\ \alpha_k^i = \underset{y}{max}\ min \{\mu_{l_i}(y), \mu_{s_k}(y)\} \end{array}$$

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 context 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 over fuzzy sets defined into the BLTS. For instance, an assessment of the user profile, p^e_k, is transformed into a fuzzy set, p'^e_k, in which this fuzzy set is described by a tuple of membership degrees $\left(\alpha^e_{k0}, \ldots, \alpha^e_{kg}\right)$.

In the same manner, the descriptions of the items are also transformed into the BLTS. An assessment, v^j_k, of the item, a_j, is transformed into a fuzzy set, v'^j_k, and it is also represented in the same way $\left(\alpha^j_{k0}, \ldots, \alpha^j_{kg}\right)$.

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

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

After the information has been unified, the system finds out which items are the closest to the user's necessities. To accomplish this step, we need to calculate the similarity between the user profile, P_e, and an item, a_j, of the database by using the following function:

$$d_j = d(P_e, a_j) = \frac{1}{l}\sum_{k=1}^{l} w_i sim\left(p'^e_k, v'^j_k\right)$$

where w_i represents the importance of each attribute and $\sum w_i = 1$, being 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 it computes the fuzzy sets as vector of membership degrees, but without taking into account its position in it, and therefore, it involves undesirable results (see [12]). In [12] a central value measurement is proposed to overcome the before problem by using it in order to define distance function.

Definition 3.2. *Giving a fuzzy set $b' = (\alpha_1, \ldots, \alpha_g)$ defined on $S = \{s_h\}$ for $h = 0, \ldots, g$, we obtain its central value cv in the following way:*

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

This value represents the central position or gravity centre 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 was defined the following function sim: [12]

Definition 3.3. *Let b'_1 and b'_2 be two fuzzy sets defined on the BLTS, $S_T = \{s_0, \ldots, s_g\}$, 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 $D = (d_1, \ldots, d_n)$ in which the system will keep the similarity between the user profile P_e and the items.

3.3. Recommendation

In this phase the system will rank the items according to the similarity values of the vector $D = (d_1, \ldots, d_n)$. The best ones will be those that are the closest to the user profile, i.e., those with the greatest score in the similarity. So, giving the item set $A = \{a_1, \ldots, a_n\}$ and giving a number r of items to recommend, 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:

$$R_A = \left(a_{q(1)}, \ldots, a_{q(r)}\right)$$

where the function q is defined in the following way:

$q : \{1, 2, \ldots, r\} \rightarrow \{1, 2, \ldots, n\}$
$q(i) \neq q(j)\ \forall i \neq j$
$q(i) \neq e\ \forall i = 1, \ldots, r$
being a_e the example given by the user
$d_{q(i)} \geq d_{q(j)}\ \ \forall i < j$

4. Example

In this example, we use a simple database composed by six items, $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$. Each item is descibed by a set of features, $C = \{c_1, c_2, c_3, c_4\}$ in order to show the working of our recommender system. In real systems we could find thousand or millions items stored in the database and more attributes are used to describe them. To describe the attributes c_1 and c_2 we have used the linguistic term set $S_{1,2}$ (see figure 3) and for c_3 and c_4 (see figure 4) the linguistic term set $S_{3,4}$. These sets are defined by the following membership functions:

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

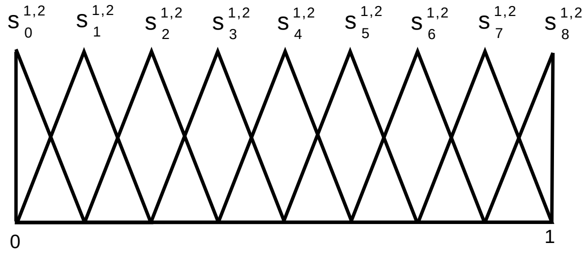


Fig. 3: The linguistic term set $S_{1,2}$.

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

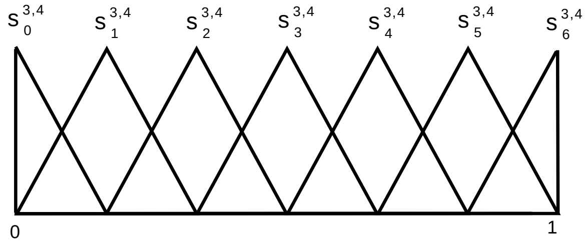


Fig. 4: The linguistic term set $S_{3,4}$.

The descriptions of the items, using these linguistic term sets, can be seen in the table 4.

A user, u_e, wants to receive a recommendation from our system. The steps the system accomplishes are the following:

1. *Acquiring a prefered example from the user:* the user states that the item a_1 is quite close to what he/she needs. From this information, the system define the initial user profile:

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

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

Table 2: Item database.

2. *Casual modification of preferences:* Nevertheless, the user realizes that attribute c_1 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 quite complex for him/her and he/she would rather use a smaller set. He/she decides to use the linguistic term set S_1^{1e} that is defined below (see figure 5:)

$$\begin{aligned}
s_0^{1e} &= Very\,low = (0, 0, .25,)\\
s_1^{1e} &= Low = (0, .25, .5)\\
s_2^{1e} &= Average = (.25, .5, .75)\\
s_3^{1e} &= High = (0.5, 0.75, 1)\\
s_6^{1e} &= Very\,high = (.75, 1, 1)
\end{aligned}$$

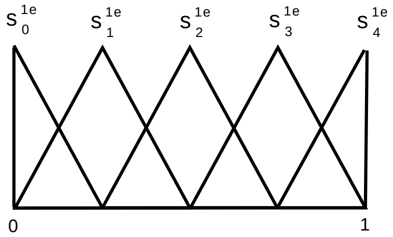


Fig. 5: The linguistic term set S_{1e}.

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

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

To compute the similarity between the user profile and items we need to achieve the following steps:

1. *Unification of the linguistic information:* The system chooses $S_{1,2}$ as BLTS (S_T). The system transforms the user profile and item database into S_T

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

Table 1: Item database expressed into the BLTS.

and the following item database (see Table 4) and user profile are obtained:

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

2. *Calculation of the similarity between the user profile and the items:* In this step the system computes the distance between the user profile and the items. First of all, the system calculates the central values of the fuzzy sets of the user profile and the fuzzy sets of every item in the database (see table 3):

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

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

And finally, we compute the distance between the user profile and each item of the the item database (see table 4).

Table 4: Distance between the user profile and the items.

a_1	a_2	a_3	a_4	a_5	a_6
1	1.84	2.81	2.57	3.31	2.31

The last step of our model is the recommendation phase. If we sort out the items according to the distance to the user profile, the system will obtain

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

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

In this contribution we have presented a Knowledge Based Recommender System that deal 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 loosing 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 to compute how close two items are. In this manner, not only have we taken into account if the assessments are the same, but we have also computed how different they are from each other.

Acknowledgements

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

References

[1] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based 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] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*, 1999.

[6] 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.

[7] X. Guo and J. Lu. Intelligent e-government services with recommendation techniques. *International Journal of Intelligent Systems on the special issue on E-Service intelligence*, 22(5):1–17, 2007.

[8] C. Hayes and P. Cunningham. Smart radio - community based music radio. *Knowledge-Based Systems*, 14(3-4):197–201, 2001.

[9] F. Herrera and E. Herrera-Viedma. Linguistic decision analisys: Steps for solving decision problems under linguistic information. *Fuzzy Sets and Systems*, (115):67–82, 2000.

[10] F. Herrera and L. Martínez. A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE-FS*, 8(6):746, December 2000.

[11] 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.

[12] 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 Intelliegence, Proceedings Lecture Notes in Computer Science*, (3131):194–204, 2004.

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

[14] 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.

[15] R.J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of DL-00, 5th ACM Conference on Digital Libraries*, pages 195–204, San Antonio, US, 2000. ACM Press, New York, US.

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

[17] M.J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

[18] P. Resnick and H.R. Varian. Recommender systems. *Association for Computing Machinery. Communications of the ACM.*, 40(3):56, Mar 1997.

[19] J.B. Schafer, J.A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, (5):115–153, 2001.

[20] B. Towle and C. Quinn. Knowledge based recommender systems using explicit user models. In *Papers from the AAAI Workshop, AAAI Technical Report WS-00-04*, pages 74–77, 2000.

[21] 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.

[22] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996.