Exploiting Linguistic Preference Relations in Knowledge Based Recommendation Systems

L.G. Pérez
University of Jaén
lgonzaga@ujaen.es

M.J. Barranco
University of Jaén
barranco@ujaen.es

L. Martínez
University of Jaén
martin@ujaen.es

M. Espinilla
University of Jaén
mestevez@ujaen.es

Abstract

Many methods and tools have been developed by e-commerce companies in order to personalize their web sites and services according to the users' necessities and tastes. The most well-known tools are the Recommendation Systems. These systems lead people to interesting items through recommendations. In this contribution we present a Knowledge Based Recommendation System that uses linguistic preference relations about users' necessities to assist them in their searches.

Keywords: Recommendation Systems, e-commerce, knowledge-based

1 Introduction

Internet has become one of the most important tools people use to communicate, find information or buy items. When people visit an e-commerce website and they state their necessities, they expect to find a variety of interesting items. However, sometimes these feedbacks are not accurate and they also include many items that do not match with his/her real expectations. In these cases, many customers could feel disappointed and could decide to visit another web site.

In order to resolve this problem, some methods and tools have been developed. The most successful have been the Recommendation Systems. The aim of these systems is to assist people to find out the best items that satisfy their necessities using recommendations, leading them to interesting items or hiding those that are unattractive. The initial types of Recommendation Systems were the Collaborative [7] and the Content-based [11] ones. The first type uses the ratings of the items of many users to filter and recommend items to a specific user. In the second one they learn a profile from user's interests based on the features presented in the items that the user has rated. Then, they use this profile to find similar items that the user could like. In both cases to obtain a suitable recommendation it is required that the user has assessed a minimum number of items. However sometimes there is no historical information about the user's tastes or necessities (for instance, when a new user visit the web site we do not know anything about him/her) and neither the Collaborative nor the Content-based systems are able to make any recommendation. Other types of recommendation systems try to overcome the problems of lack of information. For instance, the Hybrid ones [1] or the Knowledge Based Recommendation Systems [2].

We focus on Knowledge based recommendation systems that infer the recommendations about which items satisfy the users from the information provided by the user regarding his/her knowledge about items that can be recommended. The case-based reasoning [8] is one of the methods that has shown successful in this type of Recommendation Systems [4]. These systems manage three types of knowledge [3]:

1. *Catalog Knowledge:* Knowledge about the items that can be recommended and their features. In our proposal the features are described by means of numbers that belong to the interval $[0, 1]$. In future works our system will manage more complex information.

2. *Functional knowledge:* Knowledge about how to map between the user's necessities and the item that could satisfy these necessities.

3. *User knowledge:* Knowledge about the user, his/her necessities, tastes, and so on.

The User Knowledge plays a key role in the recommendation process. The more useful knowledge the system has about the user, the better recommendation it generates. To gather the information needed to obtain the User Knowledge, typical Knowledge Based Recommendation System [2] follows a two-step process:

1. Users provide a start point, usually a known item that represents their necessities. This item or example will define a initial user profile. This user profile will gather the main features of the given example.

2. Users refine interactively their profile stating new features or modifying some of them until the system achieves an acceptable knowledge about the users' necessities.

If this interactive refinament step is not well-design, the features are difficult to assess or understand or the items are described with many features, then users can find this refinement step very tedious and they could not be willing or trained to do so.

In this contribution we present a Knowledge Based Recommendation System that makes easier and improves the gathering process of the User Knowledge. Our system acquires knowledge about the users' necessities from an incomplete linguistic preference relation that the user has provided in which he/she expresses his/her preference over different items and then, using the consistency property of preference relation, the system builds a consistent relation from which it is infered the User Knowledge that is used to define the user profile.

This contribution is structured as follows, in section 2 we shall review some preliminaries. In section 3 we shall present our proposal and in section 4 some conclusions are point out.

2 preliminaries

In this section we shall review the Linguistic Computational Methods. Besides, we shall go over the Linguistic Preference Relations and we shall present a method to fill a linguistic preference relation using the consistency property.

2.1 Linguistic Computational Methods

Although the most usual representacion of information in computer sciences is by means of numbers, 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 variables instead of numerical ones. To perform computations over linguistic variables there are three models in the literature. (i) The model based on the Extension Principle [5], (ii) the symbolic one [6] and (iii) the model based on the 2-tuple linguistic representation [9]. Here we shall review the last one because it is used in this proposal.

Definition 1. *The Symbolic Translation of a linguistic term $s_i \in S = \{s_0, ..., s_g\}$ is a numerical value assessed in $[-.5, .5)$ that supports the "difference of information" between a counting of information β assessed in $[0, g]$ obtained after a symbolic aggregation operation (acting on the order index of the labels) and the closest value in $\{0, ..., g\}$ that indicates the index of the closest linguistic term in S (s_i).* it

From this concept we develop a linguistic representation model which represents the linguistic information by means of 2-tuples, (s_i, α_i), $s_i \in S$ and $\alpha_i \in [-.5, .5)$. s_i represents the linguistic label center of the information and α_i is the Symbolic Translation.

This linguistic representation model defines a set of functions to make transformations among linguistic terms, 2-tuples and numerical values.

Definition 2. *Let $s_i \in S$ be a linguistic term, then its equivalent 2-tuple representation is obtained by means of the function θ as:*

$$\theta : S \longrightarrow (S \; x \; [-0.5, 0.5))$$
$$\theta(s_i) = (s_i, 0)/s_i \in S$$

Definition 3. *Let $S = \{s_0, ..., s_g\}$ be a linguistic term set and $\beta \in [0, g]$ a value supporting the result of a symbolic operation, then the 2-tuple that expresses the equivalent information to β is obtained with the following function:*

$$\Delta : [0, g] \longrightarrow Sx[-0.5, 0.5)$$

$$\Delta(\beta) = \begin{cases} s_i & i = round(\beta) \\ \alpha = \beta - i & \alpha \in [-.5, .5) \end{cases}$$

where round is the usual round *operation, s_i has the closest index label to "β" and "α" is the value of the symbolic translation.*

Definition 4.*Let $S = \{s_0, ..., s_g\}$ be a linguistic term set and (s_i, α) be a linguistic 2-tuple. There is always a function Δ^{-1}, such that, from a 2-tuple it returns its equivalent numerical value $\beta \in [0, g]$.*

$$\Delta^{-1} : Sx[-.5, .5) \longrightarrow [0, g]$$

$$\Delta^{-1}(s_i, \alpha) = i + \alpha = \beta$$

The 2-tuple linguistic representation model has a computation model based on the 2-tuples that was presented in [9].

2.2 Linguistic preference relations and their reconstruction

In many situations the preference information is expressed in the form of preference relation. While in the literature the most common preference relation is a complete, consistent and a numerical relation, in real situations this representation may not be not suitable. For instance, users could be under time pressure and lack of information and cannot assess the alternatives with accurate values, some alternatives could be unknown by the users, or their opinions about the alternatives could be subjective. In these situations a more suitable and realistic representation could be the use of linguistic asssessments to model the subjectivity and let that some of the elements of the relation could not be provided.

However, there is no way to exploit directly incomplete preference relations and they need to be filled. Due to the fact we are dealing with linguistic information, In order to represent it we shall use the 2-tuple linguistic representation model [9] to accomplish processes of computing with words.

Hereafter, we shall present some definitions and a method for reconstructing a consistent and complete linguistic preference relation from an incomplete one. First of all, in order to be able to apply the reconstruction method we need to transform the domain of the linguistic term $S = \{s_0, ..., s_g\}$ into a linguistic term set $S' = \{s_{-g'}, ...s_0, ..., s_{g'}\}$ where the indexes of the terms has been translated so that the mid term has the index 0 and the rest of terms are placed symmetrically around it and where g' is $g/2$. We use the function Υ to transform a value in S into S':

Definition 5. Let $s_i \in S = \{s_0, ..., s_g\}$ a linguistic label, the function Υ that translates s_i into a term in $S' = \{s_{-g'}, ...s_0, ..., s_{g'}\}$ is defined as follows:

$$\Upsilon : S \longrightarrow S'$$

$$\Upsilon(s_i) = s'_{i-g/2}$$

Proposition 1. Let $s'_i \in S' = \{s_{-g'}, ...s_0, ..., s_{g'}\}$ a linguistic label, there is always a Υ^{-1} function, such that from s'_i it returns its equivalent $s_i \in S = \{s_0, ..., s_g\}$

Proof. It is trivial, we consider the function:

$$\Upsilon^{-1} : S' \longrightarrow S$$

$$\Upsilon^{-1}(s'_i) = s_{i+g/2}$$

Definition 6. Let (s_i, α_i) be a linguistic 2-tuple and let s_i be a linguistic term such that $s_i \in S = \{s_0, ..., s_g\}$, then a function ϖ that transform this 2-tuple into a 2-tuple in which its label belongs to $S' = \{s_{-g'}, ...s_0, ..., s_{g'}\}$ is defined as follows:

$$\varpi : S \times [-.5, .5) \longrightarrow S' \times [-.5, .5)$$

$$\varpi((s_i, \alpha_i)) = (\Upsilon(s_i), \alpha_i)$$

Proposition 2. Let (s'_i, α_i) be a linguistic 2-tuple and let s'_i be a linguistic term such that $s'_i \in S' = \{s_{-g'}, ...s_0, ..., s_{g'}\}$, then a function ϖ^{-1} that transform this 2-tuple into a 2-tuple in which its label belongs to $S = \{s_0, ..., s_g\}$

Proof. It is trivial, we consider the function:

$$\varpi^{-1} : S' \times [-.5, .5) \longrightarrow S \times [-.5, .5)$$

$$\varpi^{-1}((s'_i, \alpha_i)) = (\Upsilon^{-1}(s'_i), \alpha_i)$$

In order to apply the reconstruction method we need to introduce a sum operator, $\oplus$, that is closed in a 2-tuple domain.

Definition 7 Let (s_i, α_i) and (s_j, α_j) be two linguistic 2-tuple, the operation $\oplus$ is defined as follows:

$$\oplus : (S\times[-.5, .5))\times(S\times[-.5, .5)) \longrightarrow S\times[-.5, .5)$$

$$(s_i, \alpha_i) \oplus (s_j, \alpha_j) = max$$

$$\left\{(s_0, 0), \Delta\left(min\left\{\Delta^{-1}(s_i, \alpha_i) + \Delta^{-1}(s_j, \alpha_j), g\right\}\right)\right\}$$

Definition 8. [12] Let $P_{S'} = (p_{ij})_{n\times n}$ be a linguistic preference relation, then $P_{S'}$ is called a complete linguistic relation, if

$$p_{ij} \in S, p_{ij} \oplus p_{ji} = s_0, p_{ii} = s_0, \; for\ all\ i, j$$

Definition 9. [12] Let $P_{S'} = (p_{ij})_{n\times n}$ be a linguistic preference relation, then $P_{S'}$ is called a consistent complete linguistic preference relation, if

$$p_{ij} = p_{ik} \oplus p_{kj}, \; for\ all\ i, j, k$$

We must realize that this consistent property is a type of additive transitivity.

Definition 10. [12] Let $P_{S'} = (p_{ij})_{n\times n}$ be a linguistic relation, then $P_{S'}$ is called an incomplete linguistic relation, if some of its elements can not be given by the expert.

In [12] are presented the necessary properties to ensure that is possible to complete a preference relation: an incomplete preference relation can be filled if there exists at least one known element (except diagonal elements) in each line or each column of $P_{S'}$.

Definition 11. [12] Let $P_{S'} = (p_{ij})_{n\times n}$ be an incomplete linguistic preference relation, if

$$p_{ij} = p_{ik} \oplus p_{kj} \; for\ all\ p_{ik}, p_{kj}, p_{ij} \in \Omega$$

where Ω is the set of all the known elements in $P_{S'}$, then $P_{S'}$ is called a consistent incomplete linguistic preference relation

Our proposal for constructing a complete linguistic 2-tuple preference relation is:

Algorithm for constructing a complete relation provided a row or a column of the preference relation:

Step 1. Let $X = \{x_1, ..., x_n\}$ be a discrete set of alternatives. The expert must provide a row (or a column) of the relation, P, using the linguistic terms in S. We shall obtain a preference relation P where every known element will be expressed by means of a 2-tuple.

Step 2. The preference relation P is transformed into $P_{S'}$ by means of the function ϖ.

Step 3. Utilize the known elements in $P_{S'}$ to determine all the unknown elements, and get a consistent preference relation, $P'_{S'}$, using Definition 11.

Step 4. Transform the 2-tuples of preference relation $P'_{S'}$ into 2-tuples in S by means of the function ϖ^{-1} obtaining a consistent preference relation, P'.

Step 5. End.

3 A Knowledge Based Recommender System model

The main advantage of our model is that it only requires a little amount of information from the user consisting of a few examples of his/her necessities and a preference relation between them. With this information a user profile is built and it is used to find the most suitable item(s). Besides, compared to typical knowledge based recommendation systems [4] users do not need to accomplish a refinement step because the system is able to define the user profile from the preference information provided by the user.

The model is structured as follows (see fig. 1):

1. *Obtaining the user profile:* We must gather the preference information from the user, expressed by means of the closest items to his/her necessities, normally, four or five items. Then, we need to know which ones are closer to his/her real necessities. This information will be expressed by means of a relation where the user will express his/her preferences using linguistic assessments. With this information and the description of these examples, the user profile is defined.

2. *Recommendation:* Once we have the user profile, the system will find those items that cover his necessities, tastes or preferences.

In the next subsections we explain in detail these steps.

3.1 Obtaining the user profile

Here we shall present in detail the steps of our model to obtain the user profile.

3.1.1 Gathering the preference information

First of all, the user must provide a set of items (four or five) closed to his/her necessities. Let be $X = \{x_1, x_2, ..., x_m\}$ the set of items that can be recommended, and $X^u = \{x_1^u, ..., x_n^u\}$ the set of examples chosen by the user according to his preferences.

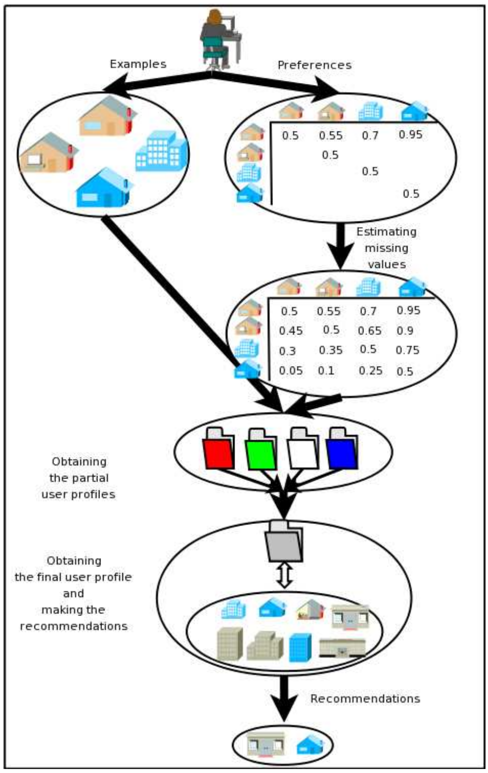

Figure 1: Recommendation Model

Then, the user must give an incomplete preference relation over the selected items. This model asks the user to provide a complete row (or column) of this relation.

3.1.2 Completing the preference relation

Before computing the user profile we need to fill the linguistic relation provided by the user using the algorithm presented in section 2 and we shall obtain the following relation:

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

Where each member of the relation will be expressed with linguistic term of S. p_{ij} is the assessment provided by the user about the preference of the item x_i^u over x_j^u and p'_{ij} is the element that has been computed in order to obtain a complete and consistent relation.

3.1.3 Obtaining the partial user profiles

For every column of the preference relation, which represents the preference of each item over a specific item x_j^u, we can obtain a partial user profile concerning the item x_j^u. For building this partial profile we propose the use of the IOWA operator (Induced OWA operator) proposed by Yager [14]. With this operator we will aggregate the vectors of characteristics of the other items different to x_j^u using the known elements of the preference relation $(p_{1j}, p_{2j}, \ldots, p_{nj})$ as order inducing values.

The IOWA operator is used to aggregate tuples of the form (v_i, a_i) where v_i is called the order inducing value and a_i is called the argument value.

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

$B_v = (b_1, \ldots, b_l)$ is the result of ordering the vector $A = (a_1, \ldots, a_l)$ according to the value of the order inducing values, indeed of the values of the elements a_i, and W^T is the column vector of weighs with the next conditions:

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

In our case, we use this operator for aggregating a set of tuples $\{(c_i^1, \ldots, c_i^t), \forall i \neq j\}$, which describe the items $\{x_i^u, \forall i \neq j\}$, according to the order of the inducing values $(p_{1j}, p_{2j}, \ldots, p_{nj})$

The result will be the partial profile, pp_j, for the item x_j^u 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 k we apply the following function:

$$c_{pp_j}^k = F_W\left(\left\langle p'_{1j}, c_1^k\right\rangle, \ldots, \left\langle p'_{nj}, c_n^k\right\rangle\right) = W^T B_v$$

where the vector $B_v = (b_1, \ldots, b_{n-1})$ is given by an ordering process, from highest to lowest value, of the elements of the set $\left\{c_i^k, \forall i \neq j\right\}$ according to order inducing values, $\left(p'_{1j}, \ldots, p'_{nj}\right)$ $\{\forall i \neq j\}$.

There are different methods to build the weighting vector $W = (w_1, \ldots, w_{n-1})$. We could associate it with a linguistic quantifier [13] or resolve a mathematical problem such as in [10].

3.1.4 Obtaining the final user profile

The following step is to combine the partial user profiles for obtaining the final user profile that describes the tastes or necessities of the user. In this phase we shall utilize the same operator than we have used in the step before, that is, the IOWA operator. So, we shall aggregate every partial profile, $\left(c_{pp_j}^1, \ldots, c_{pp_j}^t\right)$, obtained for every item x_j^u, achieving a unique and final user profile. For every attribute we shall apply the next function:

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

where the vector $B'_v = (b'_1, \ldots, b'_{n-1})$ is given by an ordering, from highest to lowest value, of the elements of the set $\left\{c_{pp_i}^k, \forall i \neq j\right\}$ according to the order inducing variables, $(p_1, \ldots, p_n)$, and the weighting vector $W' = (w'_1, \ldots, w'_n)$ that could be obtained applying a linguistic quantifier.

In order to compute p_i we shall extend the linguistic term set S to $\bar{S}$ and we shall use the following function that calculates the dominance degree for the alternative p_i over the rest of alternatives:

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

Where $\beta_{ij} = \Delta^{-1}(p_{ij})$ being p_{ij} a linguistic 2-tuple representing the preference of the alternative i over the alternative j.

Then, the closest alternative to the user's needs (the preferred one) has the highest value, the second closer one, the second higher value, and so forth.

Here we see the shape of the he final user profile that it is used in the next phase:

$$FP_u = \left\{c_{fp}^1, \ldots, c_{fp}^t\right\}$$

3.2 Recommendation

This is the final step of our model. The aim of this phase is to find the closest item to the final user profile. As we have mentioned above, we have an item database $X = \{x_1, x_2, ..., x_m\}$ which contains all the items we can recommend, where each item x_i is described by a set of features $x_i = \{c_i^1, \ldots, c_i^t\}$. In the previous step we have obtained a final user profile $FP_u = \left\{c_{fp}^1, \ldots, c_{fp}^t\right\}$. Now, we shall compute the similarity of the description for each product, x_i with the user profile:

$$\nu\left(x_i, FP_u\right) = \phi\left(\nu_1\left(c_i^1, c_{fp}^1\right), \ldots, \nu_t\left(c_i^t, c_{fp}^t\right)\right) \in [0, t]$$

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:

$$\nu_j\left(c_i^j, c_{fp}^j\right) = \gamma\left(1 - |c_i^j - c_{cp}^j|\right)$$

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

The final recommendation(s) will be those items that are closer to the final profile, i.e its overall similarity is greater, but that have not been chosen yet as examples of what the user needed.

4 Conclusions

Our knowledge based recommender system model represents an alternative for situations where the most well-known models, that is, the content-based and collaborative ones, can not be applied, because we have only a limited knowledge over the tastes and preferences of the user. We talk about situations where we have no historical information of the choices the user has made in the past, and neither have a database of the decisions taken by other users in similar circumstances.

On the other hand, this proposal presents some advantages over other knowledge-based recommender systems such as we have made easier the process of gathering information about the user's necessities or preferences. We ask for just a few examples and preferences between them, and in this manner, users do not need to know, review or understand how the items are described to be able to refine their preferences.

5 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 contentbased information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 714–720, 1998.

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

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

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

[5] R. Degani and G. Bortolan. The problem of linguistic approximation in clinical decision making. *International Journal*

of Approximate Reasoning, (2):143–162, 1988.

[6] M. Delgado, J. Verdegay, and M. Vila. On aggregation operations of linguistic labels. *International Journal of Intelligent Systems*, (8):351–370, 1993.

[7] 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.

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

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

[10] 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.

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

[12] Z. Xu. Incomplete linguistic preference relations and their fusion. *Information Fusion*, 7(3):331–337, 2006.

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

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