Information Sciences 311 (2015) 102–118

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling

C. Martinez-Cruz [a,*], C. Porcel [a,*], J. Bernabé-Moreno [b], E. Herrera-Viedma [b,*]

[a] University of Jaén, Department of Computer Science, Jaén, Spain
[b] University of Granada, Department of Computer Science and Artificial Intelligence, Granada, Spain

ARTICLE INFO

Article history:
Received 25 November 2014
Received in revised form 6 February 2015
Accepted 9 March 2015
Available online 20 March 2015

Keywords:
Recommender system
Ontology
User profile
Fuzzy linguistic modeling
Trust-network based system

ABSTRACT

Recommender systems evaluate and filter the vast amount of information available on the Web, so they can be used to assist users in the process of accessing to relevant information. In the literature we can find countless approaches for generating personalized recommendations and all of them make use of different users' and/or items' features. In this sense, building accurate profiles plays an essential role in this context making the system's success depend to a large extent on the ability of the learned profiles to represent the user's preferences and needs. An ontology works very well to characterize the users profiles involved in the process of generating recommendations. In this paper we develop an ontology to characterize the trust between users using the fuzzy linguistic modeling, so that in the recommendation generation process we do not take into account users with similar ratings history but users in which each user can trust. We present our ontology and provide a method to aggregate the trust information captured in the trust-ontology and to update the user profiles based on the feedback.

1. Introduction

Nowadays we live in the so called Information Society, in which we are bombarded with a lot of information in all fronts of our lives. This great amount of information introduces noise in the way we access to information, which makes difficult for us to find relevant information and also affects the way we make decisions. For this reason, the Internet users increasingly need fast and effective automated systems that assist them in an easy and effective manner to access to information relevant for them according to their preferences or needs [33].

Recommender systems are examples of this kind of automated systems [10,28,53]. A recommender system attempts to discover information items (movies, music, books, research resources, images, web pages, papers, etc.) that are likely to be of interest to the user. These tools separate relevant from irrelevant information and deliver it then to users who demand it, which makes them very useful for commercial organizations too. Recommender systems are broadly used for knowledge discovery and to provide personalized items in many activities as e-commerce, digital libraries, e-marketing and so on. The delivery of personalized recommendations, requires the system to have some information available about every user, such as the ratings provided by the users about the viewed or purchased items. This need for information introduces the requirement for the system to maintain users' profiles containing the users' preferences or needs. Another aspect to take into

* Corresponding authors.
E-mail addresses: cmcruz@ujaen.es (C. Martinez-Cruz), cporcel@ujaen.es (C. Porcel), viedma@decsai.ugr.es (E. Herrera-Viedma).

http://dx.doi.org/10.1016/j.ins.2015.03.013

consideration is which additional information is required by the system, and how this information is processed and managed to generate a list of personalized recommendations. One of the mostly used method to generate recommendations is the collaborative approach [47] in which the recommendations to a particular user are based upon other users' recommendations with similar profiles, taking into account the ratings provided by those users.

One key disadvantage of this approach consists of the need for a lot of ratings to obtain a good performance, which is usually difficult to achieve [42]. In fact, users typically provide just a few ratings, so the systems have difficulties to compute the similarity between two users. Therefore, collaborative approaches tend to fail in generating recommendations since they usually fail at obtaining groups of users with similar preferences. Thus, some improvements need to be introduced to overcome this situation and one promising direction is to focus on *trust*, which plays a crucial role in on-line social networks [17,55], so widespread and popular today. Trust networks are social networks in which users can assign trust scores to each other. In fact people tend to rely upon recommendations received from trusted users, such as friends, more than those generated by automatic systems [51]. Although the recommendations provided by a recommender system might be more precise, users we trust could make more useful recommendations for us. In the literature, we can find some proposals about the incorporation of trust models in recommendation systems [30,37,54,59]. In these systems, the recommendation engine uses in one way or another the trusted network between users. A key aspect to obtain good results in the generated recommendations is precisely storing the trusted network. For that reason, and because of the success they have demonstrated in similar scenarios, we propose to use ontologies as an efficient way to represent and exploit the information, especially trusted network.

Ontologies are very suitable to model different aspect of the world we live in. In the last decade, ontologies have been increasingly used within the field of recommender systems, allowing knowledge-based techniques to supplement classical machine learning and statistical approaches [6,57]. Ontologies have been introduced in recommender systems with different goals (see Section 2 for further details). In the collaborative approach, domain ontologies are mainly used to analyze the user behavior according to this knowledge structure, building user profiles [34,50]. There are even proposals including fuzzy logic in the ontological representations to allow for some uncertainty in them. For instance, fuzzy ontologies can be used to represent user profiles [8,16,27,40]. In the same way that fuzzy ontologies have been used to represent user profiles, we consider them suitable for modeling the trust between users, extracted from a trusted network.

In this sense, our proposal relies on a combination of these approaches to improve the recommendation process, namely trust networks along with trust propagation mechanisms, and user profiles based on ontologies. Our proposal is therefore, a new recommender system whose main novelties are listed below:

- We define an ontology that represents the degree of trust between users based on the evaluations provided according to their experiences. We are going to implement a multi-granular fuzzy linguistic modeling [32,36], to keep the maximum flexibility to manage the information by representing the different concepts of the system with different linguistic label sets.
- We use a domain ontology to establish the relationships between users and their preferences about the recommendation subject.
- We present a method to estimate the trust score between two users, because the trusted network can be huge and most users do not know each other.
- We propose a new recommendation approach in which the recommendations are taken from trustworthy users, i.e., we do not consider users with similar ratings history but users in which each user can trust.

The rest of this paper is set out as follows: Section 2 contains background information about the basics of recommender systems, the fuzzy linguistic modeling and the basics of ontologies. In Section 3 we present the new proposal. Section 4 addresses the validation of the system and in Section 5 we throw our conclusions based on the study findings.

2. Background

2.1. Basis of recommender systems

Recommender systems help users in the effective identification of items suiting their wishes, needs or preferences. They have the effect of guiding the users in a personalized way to relevant or useful objects in a large space of possible options [9]. In a recommender system, the users' preferences about research resources can be used to define user profiles that are applied as filters to streams of documents. In this sense, building accurate profiles plays an essential role in this context making the system's success depend to a large extent on the ability of the learned profiles to represent the user's preferences and needs. Then, in order to generate personalized recommendations that are tailored to the user's preferences or needs, recommender systems must collect personal preference information, such as user's history of purchase, items which were previously interesting for the user, click-stream data, and demographic information. Two different ways to obtain information about user preferences are distinguished [22], although many systems adopt a hybrid approach: the *implicit approach* is implemented by inference from some kind of observation, and the *explicit approach* which interacts with the users by acquiring feedback on information that is filtered.

Another key aspect to consider when designing the system is the approach used to generate the recommendations. Taking into account the knowledge source, different methods can be distinguished [10,19,22,41,44], mainly the *Content-based approach* that generate the recommendations based on the characteristics used to represent the items and the ratings that a user has given to them, and the *Collaborative approach* that generate recommendations using explicit or implicit preferences from many users, ignoring the items representation. Other proposed approaches are: *Demographic*, *Knowledge-based* and *Utility-based* [10]. Each approach has certain advantages and, of course, disadvantages, depending on the scope settings. One solution is to use a *hybrid strategy* combining different approaches in order to reduce the disadvantages of each one of them and to exploit their benefits [4,10,12,15]. For this reason, in this paper we propose the use of a hybrid approach.

2.2. Fuzzy linguistic modeling

In some situations, the information cannot be precisely assessed in a quantitative manner but can be qualitatively evaluated. Very good results have been obtained by using Fuzzy Sets Theory to model qualitative information [58]. The two main fuzzy methods for managing qualitative information are [3,23,31]: The *classical* linguistic approach, based on the use of labels whose semantics is represented by means of fuzzy sets and their associated membership functions and the *ordinal* linguistic approach, based on the use of labels whose semantics is established on an ordered structure.

2.2.1. The 2-tuple fuzzy linguistic approach

This model was introduced in [24] to avoid the loss of information that occurs when an approximation function (e.g., rounding operation) is used. Let $S = \{s_0, \ldots, s_g\}$ be a linguistic term set with odd cardinality, where the mid-term represents an indifference value to which the remaining terms are symmetric. For instance, we could use the following set of terms with 7 labels: $S = \{N, VL, L, M, H, VH, P\}$, where $s_0 = N = None$, $s_1 = VL = Very\_Low$, $s_2 = L = Low$, $s_3 = M = Medium$, $s_4 = H = High$, $s_5 = VH = Very\_High$, and $s_6 = P = Perfect$.

We assume that the semantics of labels is given by means of fuzzy subsets defined in the $[0,1]$ interval, which are described by their membership functions $\mu_{s_i} : [0,1] \to [0,1]$, and we consider all terms distributed on a scale in which a total order is defined, i.e., $s_i \leqslant s_j \iff i \leqslant j$. We consider linear triangular membership functions to be adequate to capture the vagueness of these linguistic assessments. This representation is achieved by three elements (a, b, c), where a is the point at which the membership is 1 and b and c are the left and right limits of the definition domain of the triangular membership function.

Definition 1 [24]. Let $S = \{s_0, \ldots, s_g\}$ be a linguistic term set and $\beta \in [0, g]$ a value representing the result of a symbolic aggregation operation, then the 2-tuple that expresses the equivalent information to β is obtained with the following function:

$$\Delta : [0, g] \longrightarrow S \times [-0.5, 0.5) \tag{1}$$

$$\Delta(\beta) = (s_i, \alpha), with \begin{cases} s_i & i = round(\beta) \\ \alpha = \beta - i & \alpha \in [-.5, .5) \end{cases} \tag{2}$$

where $round(\cdot)$ is the usual rounding operation, s_i has the closest index label to "β" and "α" is the value of the symbolic translation. For all Δ there exists Δ^{-1}, defined as $\Delta^{-1}(s_i, \alpha) = i + \alpha$.

Example. Let $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$ be the linguistic term set, and let $\beta = 2.8$ be the result of a symbolic aggregation operation. The 2-tuple that expresses the equivalent information to β is $\Delta(\beta) = \Delta(2.8) = (s_3, -0.2)$, because $round(\beta) = 3$ and $\beta - i = -0.2$.

In order to establish the computational model negation, comparison and aggregation operators are defined. Using functions Δ and Δ^{-1}, any of the existing aggregation operators can be easily be extended for dealing with linguistic 2-tuples without loss of information [24]. Some examples are the following:

Definition 2 (*Arithmetic mean*). Let $x = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ be a set of linguistic 2-tuples, the 2-tuple arithmetic mean $\bar{x}^e$ is computed as:

$$\bar{x}^e[(r_1, \alpha_1), \ldots, (r_n, \alpha_n)] = \Delta\left(\sum_{i=1}^{n} \frac{1}{n} \Delta^{-1}(r_i, \alpha_i)\right) = \Delta\left(\frac{1}{n} \sum_{i=1}^{n} \beta_i\right). \tag{3}$$

Definition 3 (*Weighted average operator*). Let $x = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ be a set of linguistic 2-tuples and $W = \{w_1, \ldots, w_n\}$ be their associated weights. The 2-tuple weighted average $\bar{x}^w$ is:

$$\bar{x}^w[(r_1, \alpha_1), \ldots, (r_n, \alpha_n)] = \Delta\left(\frac{\sum_{i=1}^{n} \Delta^{-1}(r_i, \alpha_i) \cdot w_i}{\sum_{i=1}^{n} w_i}\right) = \Delta\left(\frac{\sum_{i=1}^{n} \beta_i \cdot w_i}{\sum_{i=1}^{n} w_i}\right). \tag{4}$$

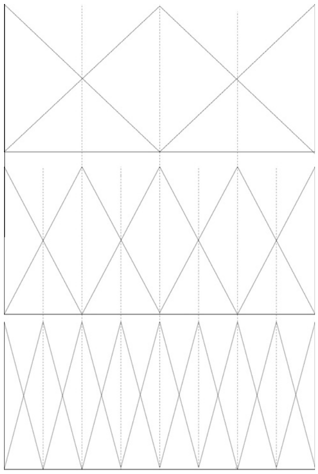

Fig. 1. Linguistic hierarchy of 3, 5 and 9 labels.

Definition 4 (*Linguistic Weighted Average Operator*). Let $x = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ be a set of linguistic 2-tuples and $W = \{(w_1, \alpha_1^w), \ldots, (w_n, \alpha_n^w)\}$ be their linguistic 2-tuple associated weights. The 2-tuple linguistic weighted average $\bar{x}_l^w$ is:

$$\bar{x}_l^w[((r_1, \alpha_1), (w_1, \alpha_1^w)) \ldots ((r_n, \alpha_n), (w_n, \alpha_n^w))] = \Delta\left(\frac{\sum_{i=1}^n \beta_i \cdot \beta_{W_i}}{\sum_{i=1}^n \beta_{W_i}}\right), \quad (5)$$

with $\beta_i = \Delta^{-1}(r_i, \alpha_i)$ and $\beta_{W_i} = \Delta^{-1}(w_i, \alpha_i^w)$.

2.2.2. Linguistic hierarchies to model multi-granular linguistic information

When different experts have varied degrees of uncertainty in relation to a phenomenon, several linguistic term sets with a different cardinality (granularity of uncertainty) are necessary. The use of different label sets to assess information is also necessary when an expert has to evaluate different concepts. In these situations, we need tools to manage multi-granular linguistic information [25,32,36].

Multi-granular fuzzy linguistic modeling based on a 2-tuple fuzzy linguistic approach and the concept of linguistic hierarchy were proposed in [25]. A *Linguistic Hierarchy, LH,* is a set of levels *l(t,n(t))*, i.e., $LH = \bigcup_t l(t, n(t))$, where each level t is a linguistic term set with different granularity $n(t)$ from the remaining levels of the hierarchy. A graphical example of a linguistic hierarchy is shown in Fig. 1. Using this *LH*, the linguistic terms in each level are the following:

- $S^3 = \{a_0 = Null = N, a_1 = Medium = M, a_2 = Total = T\}$.
- $S^5 = \{b_0 = Null = N, b_1 = Low = L, b_2 = Medium = M, b_3 = High = H, b_4 = Total = T\}$.
- $S^9 = \{c_0 = Null = N, c_1 = Very\_Low = VL, c_2 = Low = L, c_3 = More\_Less\_Low = MLL,\ c_4 = Medium = M, c_5 = More\_Less\_High = MLH, c_6 = High = H, c_7 = Very\_High = VH,\ c_8 = Total = T\}$.

A family of transformation functions among labels from different levels was defined in [25] to combine multi-granular linguistic information with no loss of information:

Definition 5. Let $LH = \bigcup_t l(t, n(t))$ be a linguistic hierarchy whose linguistic term sets are denoted as $S^{n(t)} = \{s_0^{n(t)}, \ldots, s_{n(t)-1}^{n(t)}\}$. The *transformation function* between a 2-tuple that belongs to level t and another 2-tuple in level $t' \neq t$ is defined as:

$$TF_{t'}^t : l(t, n(t)) \longrightarrow l(t', n(t')) \quad (6)$$

$$TF_{t'}^t(s_i^{n(t)}, \alpha^{n(t)}) = \Delta\left(\frac{\Delta^{-1}(s_i^{n(t)}, \alpha^{n(t)}) \cdot (n(t') - 1)}{n(t) - 1}\right) \quad (7)$$

To define the computational model, a level is selected that makes the information uniform (e.g., the highest granularity level), allowing use of the operators defined in the 2-tuple fuzzy linguistic approach.

2.3. Ontologies and their relationship with recommender systems

After the Ontology concept was first defined by philosophers to describe reality, several disciplines also interested in information modeling such as Computer Science have redefined it [20,21,52]. One of the simpler and more recent visions

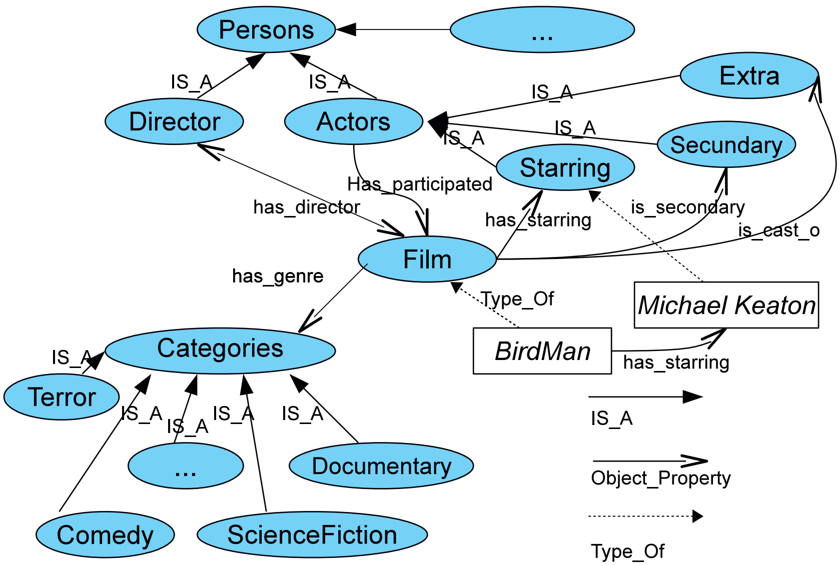

Fig. 2. Example of an ontology.

of this concept is proposed by Agarwal [2] who states that *an ontology is the manifestation of a shared understanding of a domain that is agreed between a number of agents and such agreement facilitates accurate and effective communications of meaning, which in turn leads to other benefits such as inter-operability, reuse and sharing* [29]. This knowledge representation model has appeared together with the Semantic Web [5], to provide a semantic structure and meaning to the unstructured data of Web. Then, for example, meanwhile in XML or other specifications defines concepts like *Egg* and *Muffin* separately, in the Semantic Web we know that a *Muffin is made of Egg*.

In Fig. 2, we can see a brief example of an ontology about *movies*. As we can see, an ontology is defined by classes such as: *Categories, Persons, Actors, and Films*. These classes are defined by *datatype* properties (similar to attributes in databases), i.e. *year*, *title* or *duration* of a film, and object properties which are used to stablish relationships between classes, i.e. *has_genre, has_starring*, etc. There are instances, which represent that an element is a *type_of* a class, like *Birdman* which is a *Film*. Constrains play an important role, in order to stablish restrictions such as: cardinality (number of values of a property), value constraints, if a property is *transitive*, *functional*, etc. Finally, axioms allows to infer knowledge from an ontology which has not explicitly modeled or which has not been discovered yet. There are some reasoners like Racer, Pellet or Fact++.

Thanks to their popular use, nowadays we can find many services, methodologies and languages related to ontologies [18,29], and what is more important, there are many repositories of ontologies available to users.[1]

Ontologies have been playing a relevant role in recommender systems. They are being widely used both at content-based and to represent the user profiles because they provide a common vocabulary with terms interrelated semantically [50]. Those content-based recommender systems that use ontologies have been called Ontology-based recommender systems [45] and they use ontologies to complement other methods for representing the system knowledge. There are recommender systems that use ontologies to expand the user interests in the items which are identified in the ontology. In general, most of the developed recommender systems proposals that involve domain ontologies use them to measure the preferences of users to the items of the content [35,50,12,49]. Other proposal, such as Blanco-Fernández et al.'s [7] uses ontologies in a hybrid approach where not only all kind of relationships are included but also ontology inference is performed to recommend tourist routes. A good study about the use of ontologies in content-based and hybrid recommender systems can be found in Ruotsal, dissertation [46].

The use of Fuzzy Ontologies in recommender systems is shaping as an important research topic. A Fuzzy Ontology, in a non-formal definition can be considered as a set of concepts (classes), relationships and axioms where each relation is measured with an uncertainly degree. We can find some formal definitions of a fuzzy ontology, fuzzy relationships and fuzzy inference process in [11,8]. There are many argues in the literature about the definition of fuzzy ontologies which consider that real world applications or knowledge include imprecision and vagueness. Despite this, there are alternative approaches, that combines fuzzy logic and ontology modeling techniques to develop recommender systems [13] which recommend personalized recipes to patients using a domain recipe ontology and fuzzy logic as a guide to infer recommendations. In this proposal we also deal with fuzzy knowledge in the ontology.

[1] http://www.w3.org/wiki/Ontology_repositories.

2.4. Trust networks

Trust plays an essential role in many online social networks. An example is Epinions,[2] suggesting recommendations about product reviews written by reliable (trusted) users. Trust networks are social networks in which users can explicitly assign trust scores to rate other users, i.e., the users can express their level of trust in another users. But in the actual information society, trust networks are usually very large and therefore a lot of users do not even know the vast majority of other users. For this reason, we need to use a method to estimate the trust degree between two users. The idea is to search for a path between the two users and propagate the trust degrees found along the path. But usually, we can find several paths between two users, so we may select the most relevant and aggregate (using the corresponding operators) the propagated trust degrees into the trust degree estimation. Due to computational complexity and/or in order to eliminate less informative paths or to select the most relevant paths, an upper path length limit is typically imposed. It is called *horizon*, H, and typical values for H are 2 or 3.

To aggregate the propagated trust degrees of the paths found there are several alternatives [54]. Some of them are based in Ordered Weighted Average (OWA) operators [56]. To overcome the problems of the majority guided OWA operators, in [39] the authors propose a majority guided induced OWA (IOWA) operator [14]. In this case, the reordering of the set of values to be aggregated is not induced by themselves (like in OWA operator) but this reordering is induced by means of a variable.

Using this operator in [26] is defined a majority guided linguistic IOWA operator, MLIOWA, that we adopt in our system, because it allows us to work with linguistic information and overcome the previous problem with the majority guided. It is defined according to the following expression:

$$\Phi_Q((u_1, p_1), \ldots, (u_n, p_n)) = s_k \in S \tag{8}$$

$$\text{with } k = round\left(\sum_{i=1}^{n} w_i \cdot ind(p_{\sigma(i)})\right)$$

such that:

- $(u_{\sigma(i)}, p_{\sigma(i)})$ is the pair with $u_{\sigma(i)}$ the i-th lowest value in the set $u_1, \ldots, u_n$.
- $u_i = sup_i$, where sup_i is the overall support of value p_i, obtained as:

  $$sup_i = \sum_{j=1}^{n} sup_{ij} | sup_{ij} = \begin{cases} 1 & if |ind(p_i) - ind(p_j)| < \alpha \\ 0 & otherwise \end{cases}$$

  with $\alpha \in \{0, 1, \ldots, \tau\}$ and sup_{ij} a binary support function that expresses the support from p_j for p_i or the similarity between both values.
- $ind(s_i) = i$.
- Q is a linguistic quantifier representing the concept of fuzzy majority in the aggregation. It is used to compute the weighting vector $W = (w_1, \ldots, w_n)$, such that $\forall i \in 1, \ldots, n, w_i\ in[0, 1], \sum_{i=1}^{n} = 1$, and:

  $$w_i = Q\left(\frac{sup_{\sigma(i)}}{n}\right) \Big/ \sum_{j=1}^{n} Q\left(\frac{sup_{\sigma(j)}}{n}\right)$$

  with $Q(sup_{\sigma(i)}/n)$ denoting the degree to which $p_{\sigma(i)}$ represents the majority.

With respect to the variable inducing the reordering of the set of values to be aggregated, the importance degree of a particular topic is usually chosen, because there are topics more important than others. However, in a trust network we work with information about reliability, so that we use as variable inducing the reordering the average global trust of all users of each of the founded path. To compute the global trust of a user we use PageRank [38] because is one of the most widely used global trust metric.

3. A new recommender system based on trust and ontologies

In this section we present the new model of recommender systems based on trust and ontologies, designed using a multi-granular linguistic modeling. We work with a multi-granular fuzzy linguistic approach [25], in order to allow for higher flexibility in the communication processes of the system. Different label sets $(S_1, S_2, \ldots)$ are used to represent the different concepts required for the system operation. These label sets, S_i, are selected from among those that compose a LH, i.e., $S_i \in LH$. The number of different label sets used is limited by the number of LH levels. In many cases, therefore, the label sets S_i and S_j can be associated with the same LH label set but with different interpretations according to the concept to be modeled. The different concepts assessed in the system are the following:

[2] http://www.epinions.com.

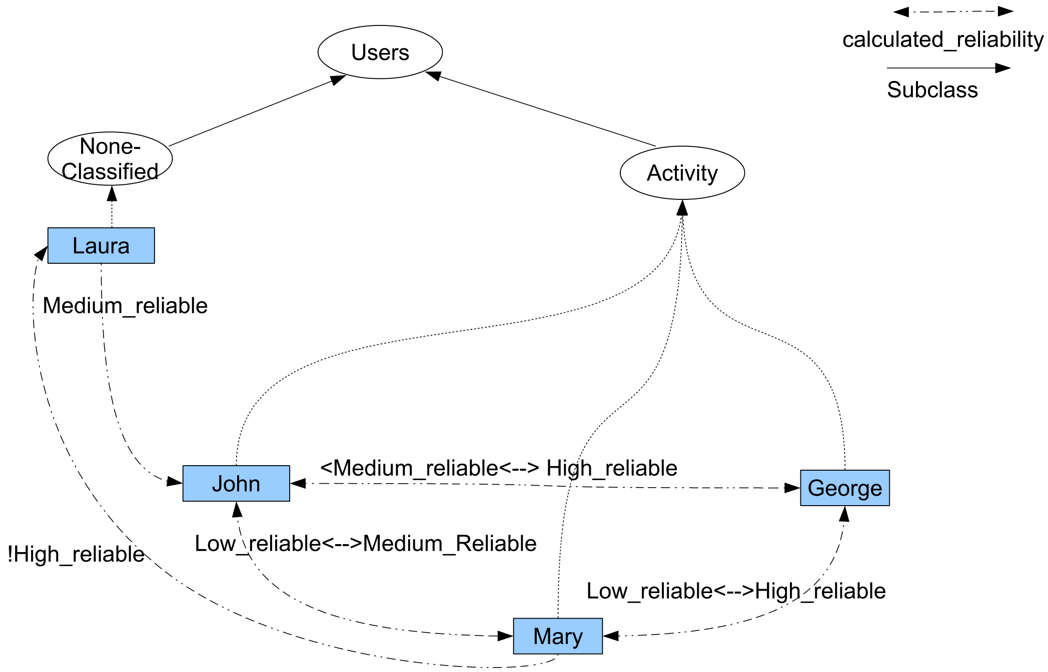

Fig. 3. Ontology for representing users reliability.

- *Degree of trust* of a user relative to another, which is labelled in S_1.
- *Membership degree* of item scope with respect to each category used in the domain ontology, which is labelled in S_2.
- Predicted *degree of relevance* of item for an user, which is labelled in S_3.
- *Degree of satisfaction* with a recommended item expressed by an user, which is labelled in S_4.

Following the *LH* depicted in Fig. 1, level 2 (5 labels) is used to represent the degrees of trust, membership and satisfaction ($S_1 = S^5$, $S_2 = S^5$ and $S_4 = S^5$) and level 3 (9 labels) is used to represent the predicted relevance degrees ($S_4 = S^9$). As the degrees of trust, membership and satisfaction are provided initially by the users, we use a set of 5 labels to facilitate them the characterization of resource items or user interest topics. On the other hand, as the predicted relevance degrees are computed automatically by the system we use the set of 9 labels which presents an adequate granularity level to represent the results. Using this *LH*, the linguistic terms in each level are the following ones:

- $S^5 = \{b_0 = None = N, b_1 = Low = L, b_2 = Medium = M, b_3 = High = H, b_4 = Total = T\}$.
- $S^9 = \{c_0 = None = N, c_1 = Very\_Low = VL, c_2 = Low = L, c_3 = More\_Less\_Low = MLL,\ c_4 = Medium = M,$ $c_5 = More\_Less\_High = MLH, c_6 = High = H, c_7 = Very\_High = VH,\ c_8 = Total = T\}$.

The system uses this information to produce new recommendations that better suits the user preferences. In the following subsections all components are described in detail.

3.1. Knowledge base representation using ontologies

The concepts of trust, membership, predicted relevance and satisfaction, together with users, items and contextual information constitute the knowledge base of our proposal. The system architecture consists of the following elements:

- *On2Trust*: an ontology that models the trust between users. In this ontology we manage fuzzy data by defining fuzzy properties and classes explicitly.
- Domain ontology: the ontology that classifies semantically the system items. This ontology allows to establish the relationship between users and the items.
- Database: an ordinary database where the ontology and data are stored. Ontologies represent the system semantic structure whereas data is stored in databases to preserve the system performance.
- Recommendation engine represents the computing process that classifies information in the ontology and generates recommendations. It is widely described in Section 3.2.

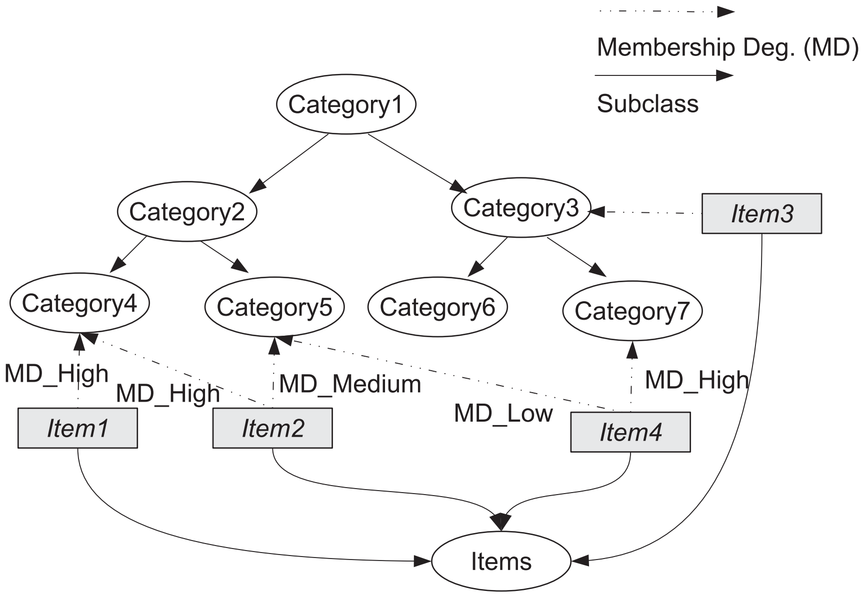

Fig. 4. Ordinary domain ontology.

3.1.1. On2Trust: Trust network ontology

On2Trust is an ontology that holds users in the system according with their degree of trust. This concept is modeled in the ontology as an object property, whose existence represents the reliability relationship between two users. This property has an asymmetric value, because the reliability degree of one user to with respect to another particular user does not imply the opposite meaning. Values of this concept are represented using the level S_1 of the *LH*, i.e., we use 5 labels (S^5). In the ontology, these properties have been called *None_Reliable*, *Low_Reliable*, *Medium_Reliable*, *High_Reliable* and *Total_Reliable*.

In Fig. 3 we can see an example of behavior of the *On2Trust* ontology. In this figure, users are represented in a gray rectangle, and these ontology instances are classified. Users convey their trust to other users using the labels previously described as trust object properties. The direction of the property is also included in the figure. For example, we can see how *Laura* has a *Medium* trust in *John* and *Mary* has *High* trust in *Laura*.

3.1.2. Domain ontology

A domain ontology is included in the system to keep the items semantically organized. This ontology can be a "light weight" one because we only require a simply categorization of the items. In Fig. 4, we show an example of generic ontology that satisfies this simply classification of terms and instances. In this ontology, we represent items as instances of a generic class, whose characterization would depend of the particular problem. The items establish a relationship with the domain ontology using a predefined object property that represents the concept of membership degree, assessed in S_2 (with 5 labels) which is defined in the ontology as *MD_None*, *MD_Low*, *MD_Medium*, *MD_High* and *MD_Total*.

3.1.3. Knowledge base: Domain and On2Trust ontologies

Domain and *On2Trust* ontologies are used together to establish the relationship between users and items in order to represent the degree of satisfaction and predicted degree of relevance. The former is explicitly defined by an user and stored in the ontology using these properties: *None_Satisfaction*, *Low_Satisfaction*, *Medium_Satisfaction*, *High_Satisfaction* and *Total_Satisfaction*. The later is described in Section 3.2 and refreshed each time the system computes the recommendations, using these properties: *None_Predicted_Relevance*, *Very_Low_Predicted_Relevance*, *Low_Predicted_Relevance*, *More_Less_Low_Predicted_Relevance*, *Medium_Predicted_Relevance*, *More_Less_High_Predicted_Relevance*, *High_Predicted_Relevance*, *Very_High_Predicted_Relevance* and *Total_Predicted_Relevance*.

An example about how the information is represented using both ontologies, is shown in Fig. 5. Here, each pair item-user is related using two properties, the degree of satisfaction and the predicted degree of relevance. For example, *Item 2* has *Low Satisfaction* to *Mary* but the recommendation engine predicts that it would have a *Medium Relevance* for her.

As described in the previous sections, fuzziness in this ontology has been modeled as object properties. This way of modeling responds to efficiency reasons due to the recommendations calculation process results are categorized for each concept using the specifications of a *LH* in the answer, which are defined in the first part of Section 3. The ontology only keeps the information semantically stored.

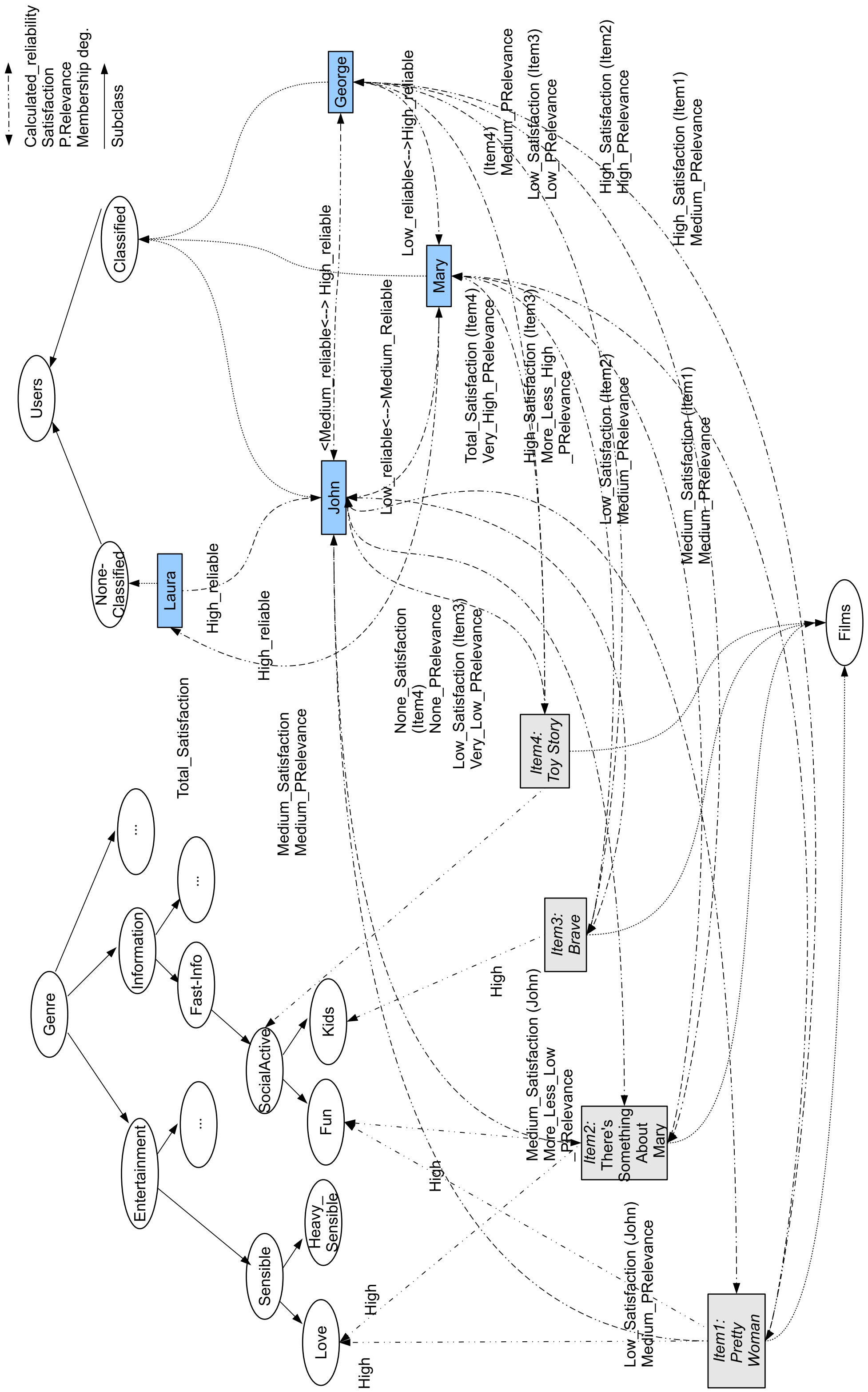

Fig. 5. Complete vision of the system ontologies: *On2Trust* and domain ontology.

3.2. Recommendation approach

To generate the recommendations, we need to infer knowledge from the ontologies presented in the previous subsection. In this manner our recommendation approach does not take into account the users similarities, but the knowledge gained from the ontologies *On2Trust*. That is, we exploit trust information obtained from *On2Trust*. The users explicitly express trust degrees to other users, and this knowledge is organized in *On2Trust*. However, a large number of users have not supplied the trust degrees to many other users. Therefore, to generate recommendations, we need a method to estimate the trust degree between users. Then, to estimate the level in which a user u trust in other user $v, \tau_{u,v}$, we aggregate the propagated trust degrees by all paths found, applying the MLIOWA operator in this manner (see Section 2.4):

$$\tau_{u,v} = \Phi_Q((AT_1^{u,v}, t_1^{u,v}), \ldots, (AT_n^{u,v}, t_n^{u,v})) \tag{9}$$

where $AT_i^{u,v}$ is the average global trust of all users found in the path i between u and v, $t_i^{u,v}$ is the propagated trust in that path between the two users and according to the majority represented by the fuzzy linguistic quantifier Q. To apply this operator we follow the configuration established in [26], i.e., we assume that the linguistic quantifier $Q_1 = most_of$ defined by the parameters $(0.3, 0.8)$ and $\alpha = 1$. In that paper a working example is available.

As we mentioned at the beginning of this section, users similarities are not taken into account in our model, i.e., in the procedure to estimate the rating a user is going to give to an unknown item, similarities are replaced by the values of trust. These values are explicitly supplied by the users or estimated with the described procedure. So, if we wish to estimate (if no evaluation is stored yet) or upgrade the relevance of a item i for a user u:

1. Identify the set of trusted users of u, Γ_u. To do that, we estimate the trust between u and all other users (see Eq. (9)) taking into account the selected horizon, i.e. $\tau_{u,v}\ \forall v \in \Upsilon$ with $v \neq u$ and Υ the set of users. As $S_1 = S^5$, we consider that the user v is a trusted user of u if $\tau_{u,v} > (s_2^5, 0)$, i.e., if the linguistic similarity degree is higher than the mid linguistic label.
2. To recovery the assessments provided by the trusted users of u over the item i, i.e., the linguistic satisfaction assessments $sat(y, i) \in S_4,\ \forall y \in \Gamma_u$.
3. The item i is recommended to u with a predicted relevance degree $p_rel(u, i) \in S_3 \times [-0.5, 0.5]$ which is calculated as follows:

$$p_rel(u,i) = \bar{x}_l^w((TF_{S_4}^{S_5}(sat(y_1,i),0), TF_{S_4}^{S_1}(\tau_{u,y_1})), \ldots, (TF_{S_4}^{S_5}(sat(y_n,i),0), TF_{S_4}^{S_1}(\tau_{u,y_n}))), \tag{10}$$

where $y_1, \ldots, y_n \in \Gamma_u$, $\bar{x}_l^w$ is the linguistic weighted average operator (see Definition 4) and $TF_{t'}^{t}$ is the transformation function between a 2-tuple that belongs to level t and another 2-tuple in level $t' \neq t$ (Definition 5).

Finally, when users have received the recommended items, they are asked to assess the relevance of these recommendations in order to update their profiles. Users communicate their linguistic evaluation judgements to the system, $rc \in S_4$, indicating their satisfaction with the recommendations (higher values of rc = greater satisfaction). This information is updated in the ontology to be taken into account in future recommendations.

3.3. Overview of system operation

To operate with the proposed system, we have used the widely known *MovieLens* database [1] which has been extended to include data about user trust. The structure of our database is shown in Fig. 6 and consists of the following information:

- A description of films and their categories obtained from MovieLens.
- A description of users inserted into the system.

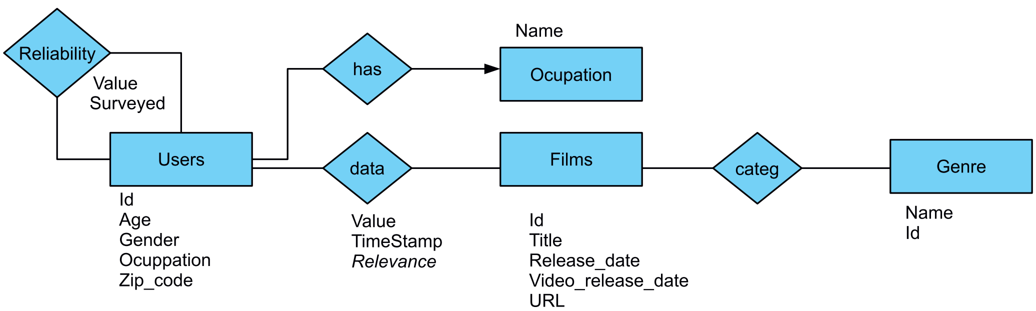

Fig. 6. Entity relationship diagram of the extended MovieLen database.

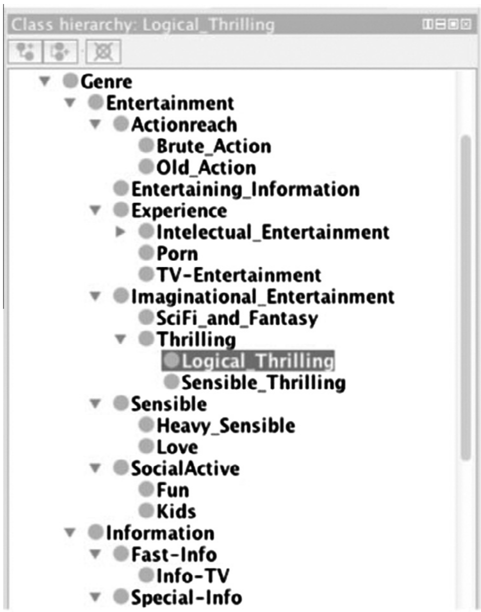

Fig. 7. Movie ontology (only genre classification).

- A set of ratings of a film by an user, obtained from MovieLens and the feedback of the users. In this manner, we avoid the cold-star problem. However, if we implemented the system without MovieLens information, we would need other technique to solve this problem, such as initially adopted the content-based approach [28] until a enough number of ratings is available.
- Information about reliability among users. Notice that trust has been included in the database randomly due to this database extension has been created explicitly to test this proposal. In a real situation, to start working the system requires that users provide some trust degrees in others. For this reason, when the users are inserted into the system, they must provide explicitly at least five degrees of trust in other users.

This database is used as the storage back-end of the ontology, where recommendation system data is organized semantically. To do that, we have used the OWL API[3] (OWLDB), a framework that implements the management of OWL ontologies. Besides, we have used *MovieOntology*[4] (see Fig. 7) to establish a correspondence between classes in the ontology and database genres. Then, the ontology extends the initial number of genres to 27, allowing to increase the vocabulary of our system and the connections among items thanks to ontology relationships.

The process consists of the stages shown in Fig. 8. Initially, the ontology is populated and data is exported to a relational database. This population is performed from different data sources. Users are defined in the *On2Trust* ontology and initial user trust relationships are defined in it, items are categorized by genre in the domain ontology and each user-item pair with satisfaction information is represented linking both ontologies. Once the recommendation process starts, users get out of the *Unclassified Users* class and trust relationships are established between every user. The reliability relationship between two users are calculated following the method described in 3.2. When this process ends the system obtains a prediction about the relevance degree that a user will assign to a item. All this information is also represented in the ontology.

4. Experiments and evaluation of the system

In this section we present experiments we have conducted to evaluate the performance of the proposed system. In the domain of recommender systems, the evaluation of different approaches is carried out according to two types of tests, on-line and off-line tests. On-line tests require the system to be fully operative to gather a vast amount of data, which makes this kind of validation very time consuming and cost intensive. Therefore, the majority of the recommender systems in the research domain are validated with off-line tests, where a predefined set of data is employed. This led us to adopt the off-line tests approach. In the subsequent sections we describe each and every test performed and discuss the results obtained in each case.

[3] OWL API is a framework that manages OWL ontologies developed by Manchester University: http://owlapi.sourceforge.net/.
[4] http://www.movieontology.org/.

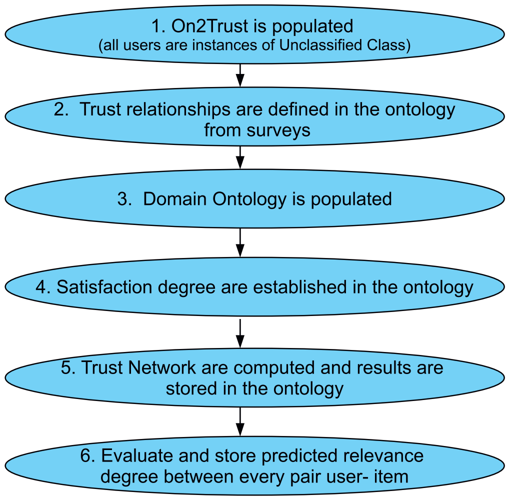

Fig. 8. Overview of system operation.

4.1. Dataset

To perform these experiments we have used a variation of the Epinions dataset[5] [30]. In our case we could not use the MovieLens dataset because it does not contain information about trust between users, and the modified dataset used in Section 3.3 contains information about trust but randomly generated and that is not significant. The Epinions dataset was collected by Paolo Massa in a 5 week crawl (November/December 2003) from the Epinions.com Web site. The dataset contains the following information: 49,290 users who rated a total of 139,738 different items at least once, writing 664,824 reviews; the total number of issued trust statements is 487,181. Users and items are represented by anonymized numeric identifiers. We have used the ratings data and trust data files. The former has the following format ⟨*userID itemID ratingValue*⟩. The second has the format ⟨*sourceUserID targetUserID trustStatementValue*⟩.

In order to get a better understanding about the data set we are going to use for the evaluation, we extracted some background information. In the histogram represented on the left side of Fig. 9, we can see that the vast majority of the users have rather a small amount of links. To understand the rating behavior, we also produced a histogram (see the right side of Fig. 9): for the users who rated items, higher ratings are predominant (especially 4 and 5), whereas lower ratings are pretty uncommon (1 and 2). Finally, we combined both histograms first segmenting the users set according to the number of links they have and then analyzing the distribution of ratings for each segment (see Fig. 10); as we can see, user with a larger number of links to other users (> 20) give higher rankings in comparison to users with an average number of links.

Due to performance reasons, for the execution of the experiments we have slightly modified the original dataset. We have taken a reduced Epinion dataset containing a sample of the first 1000 users and 2000 items. The ratings subset for this reduced subset still contains 7841 ratings and 52,548 trust statement values. The modified data set is loaded into the ontology, as we have explained previously, to enable the experimentation according to the proposed approach.

4.2. Experiments description

To develop the experiments we implemented the approach proposed in this paper, considering $H = 2,\ H = 3$ and $H = 4$ (called *Trust-H2*, *Trust-H3* and *Trust-H4*). In terms of algorithmic complexity our approach is exponential, although we have chosen a multi-threaded solution that significantly reduces execution times. In order to compare the results with other techniques, we have implemented a set of collaborative approaches, the most commonly used in recommendation systems. In that case, instead of generating the recommendations just by considering the users' trust extracted from *On2Trust*, these approaches take into consideration the similarity of their ratings. Unlike our suggested ontology proposed to facilitate knowledge representation on trust between users, collaborative approaches cannot use *On2Trust*. We implemented both *item-based* and *user-based* approaches [10] with different configurations. We have also analyzed the impact of having a different number of neighbors on the similarity computation, achieved by using different values for the variable K, which represents the most similar k-users. Specifically, we used following values for $K : 10,\ 20$ and 50. In addition, we combined these values with two different similarity metrics, such as *Cosine* and *Pearson*. We name *Col-It-* or *Col-Us-* the algorithms *item* and *user* based respectively; likewise we added to the algorithm name the suffixes *Cs* or *Ps* for Cosine or Pearson metrics; the last suffix corresponds to the number of neighbors ($K10,\ K20$ or $K50$). It might well be considered that these approaches have a linear complexity.

[5] http://www.trustlet.org/wiki/Epinions.

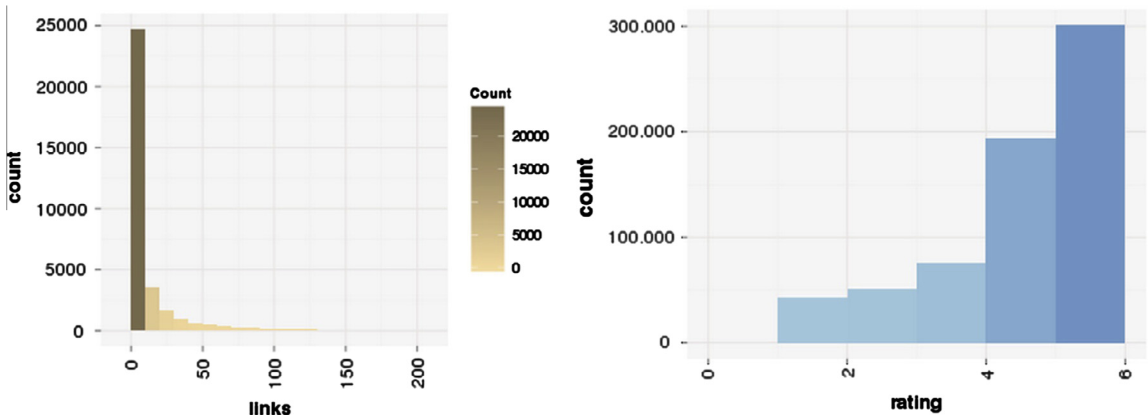

Fig. 9. Histogram showing the amount of links per user (left) and item ratings (right) in the Epinion data set.

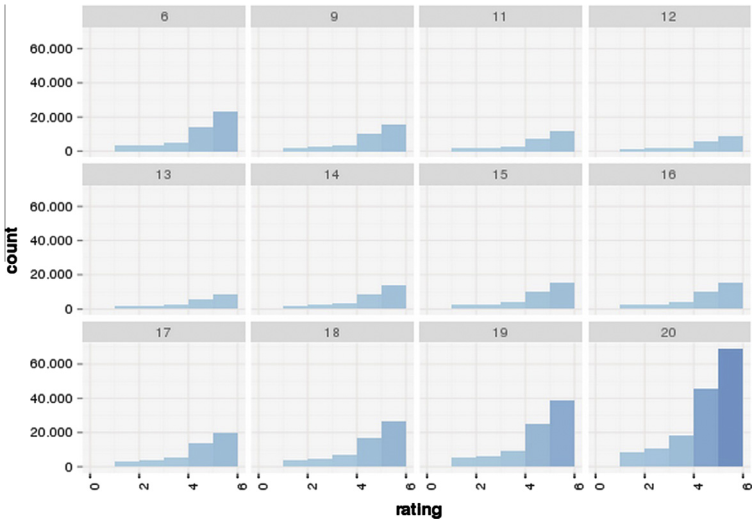

Fig. 10. Ratings distribution for the users grouped by the number of links to other users in the Epinion data set.

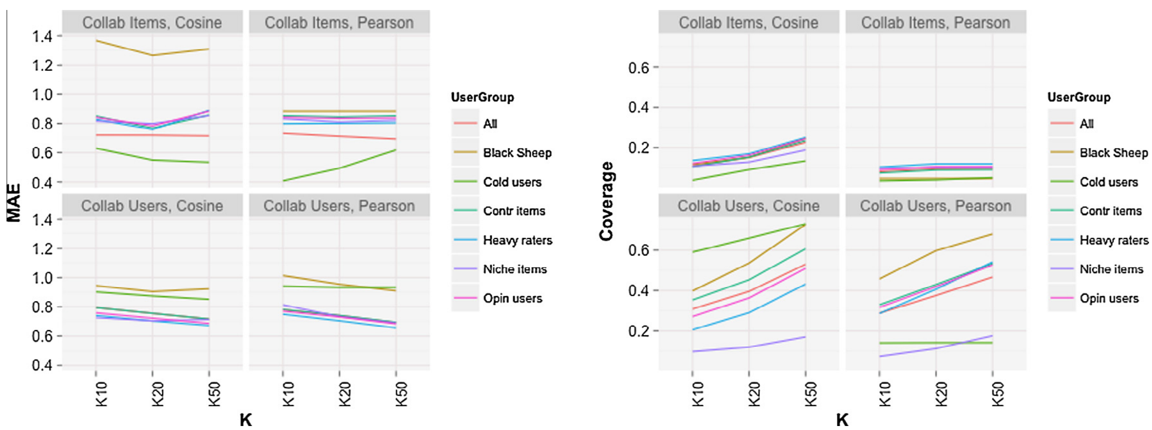

Fig. 11. MAE and Coverage obtained with differents configurations of collaborative approach.

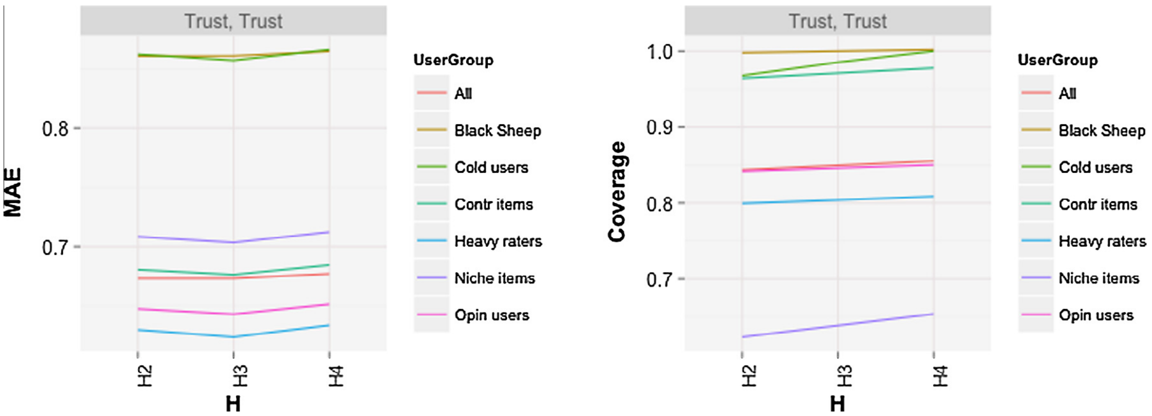

Fig. 12. MAE and Coverage obtained with our suggested approach for differents horizons.

Once the aforementioned data set has been configured and adapted to our platform, and the different approaches to compare have been implemented, we carried out a cross-validation process [43]. We took the original data set and partitioned it into two subsets, one for training and the other one for validation. For designing our experiments, we partitioned the data set into a training set with 80% of the data and a validation set with the additional 20%. Thus, we first performed the analysis on the training set and then we tested against the validation set, in both cases using the ontology. In order to minimize the bias introduce by the way of choosing these training and validation subsets, the process is performed several times but with different partitions. It is called *k-fold cross validation*, because the original set is randomly divided into *k* partitions, and the process is run *k* times, so that each partition is used at least once as a test set. In each iteration, the partition to be used as the test set is taken to determine the error of our model and the additional $k - 1$ partitions are used as train set. As part of the

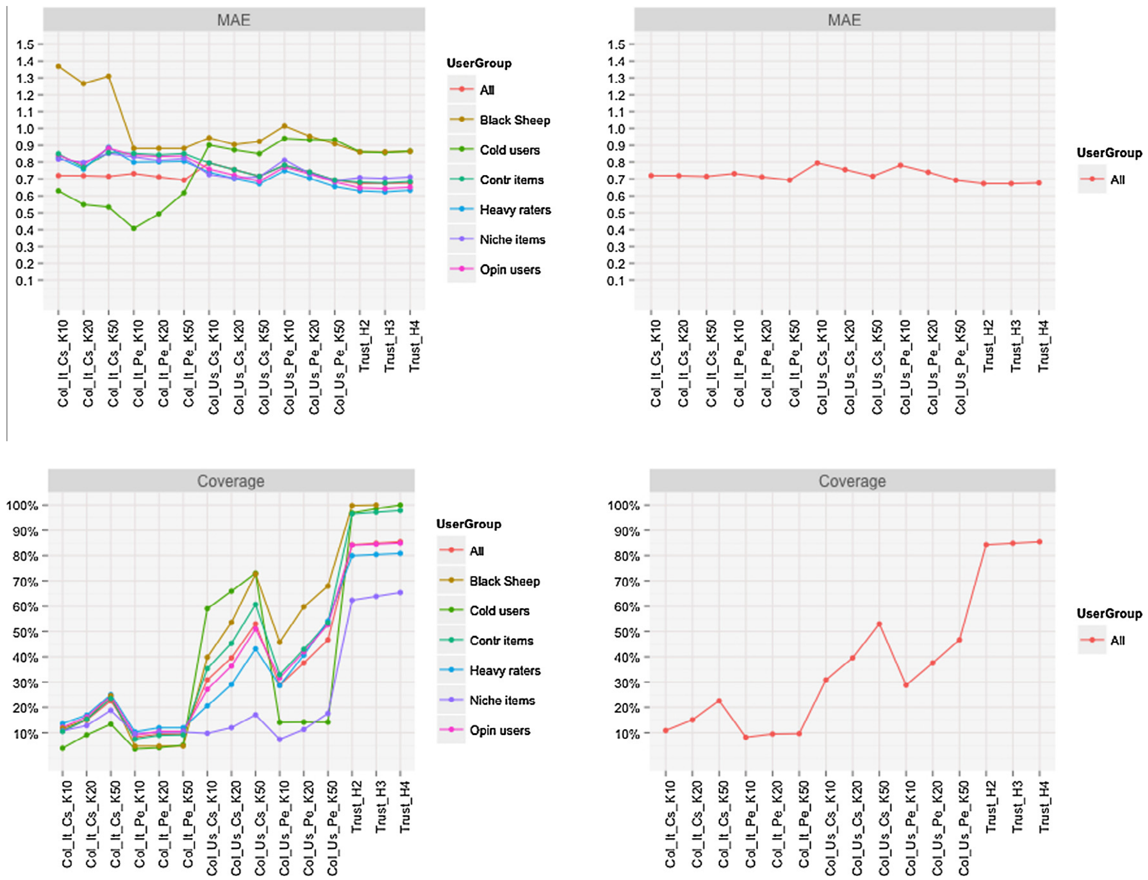

Fig. 13. Comparison of MAE and coverage between *On2trust* and the other schemes.

process, the average error per iteration is computed, which helps us understanding the deviation from the predictions generated by the model and the expected real values. Typically k is chosen to be 4, 5 or 10. For our validation we also relied on a standard value and took the value $k = 5$.

The purpose of carrying out this cross validation is obtaining a value to measure the accuracy of the different algorithms for predicting ratings and therefore having a metric to compare their performance. To measure the accuracy, we adopted the *Mean Absolute Error (MAE)*, which is typically used in the evaluation of recommender systems [48] and whose purpose is to quantify the degree of accuracy for the predictions generated by the recommender system vs. the real value created by the users. It is computed as follows: $MAE = \frac{\sum_{i=1}^{n}|p_i - r_i|}{n}$, where r represents the set of ratings generated by the users and p the predicted ones by the system. The lower the MAE, the more we can trust the predictions generated by the recommender system.

On the other hand, we also analyzed the *coverage* achieved with each approach [48]. This measure refers to the proportion of ratings of the validation set the system can generate a prediction for.

To complete our experiments, we studied the performance and coverage of different configurations on different portions of the input data, classifying users and items into different types [30]. The users are classified in these types: *cold start users* who provided from 1 to 4 ratings, *heavy raters* who provided more than 10 ratings, *opinionated users* who provided more than 4 ratings and whose standard deviation is greater than 1.5, *black sheep users* who provided more than 4 ratings and for which the average distance of their rating on corresponding item with respect to mean rating of the item is greater than 1. The items are classified in: *niche items* which received less than 5 ratings and *controversial items* which received ratings whose standard deviation is greater than 1.5.

4.3. Experiments results

First, we focus on collaborative approaches. Fig. 11 shows respectively the MAE and coverage for cosine/Pearson similarity measure for item-based and users-based collaborative approaches for the different user groups. The MAE obtained is similar for all combinations (except for small differences), but the coverage is better with the user-based collaborative approach. In both cases, the higher the number of neighbors, the better the results, especially in coverage.

Now we analyze the results obtained with our approach based on trust and ontologies. Fig. 12 shows respectively the MAE and coverage obtained with our proposal for different horizons. These figures show that a higher horizon value does not guarantee better results in MAE terms (in fact, the better MAE is obtained with $h = 3$), but in coverage. Moreover, a higher horizon value penalizes the time to results as it implies much higher execution time.

To closure up the system evaluation, we present the comparison of our proposal with the different recommendation approaches analyzed. Fig. 13 show the results of this comparison: on the left side the result with all the groups of users are presented, while the right side shows the results for all users, not classified. We can see that the better MAE is obtained with item-based collaborative implementation for cold users, but only for very specific situations. However, in general terms, we see that *On2trust* clearly outperforms the other approaches; specifically, we have achieved an improvement of 2.42%. Beyond that achieved improvement in MAE, the best results of our proposal manifest in terms of coverage. In Fig. 13 clearly see to which extent the coverage obtained with our proposal outperforms the other methods.

5. Concluding remarks

In this paper, we have presented a recommender system incorporating ontologies to improve the representation of user profiles. We have developed *On2Trust*, an ontology to efficiently characterize the trust network between users, using the fuzzy linguistic modeling to facilitate the representation of different concepts. The system also incorporates a domain ontology to represent the relationships between users and their preferences about the items.

The main idea of our new recommendation approach consists of not taking into account users with similar ratings history in the recommendation generation process, but rather trustworthy users – or the ones each user particularly can trust. To achieve this, we have proposed a method to estimate the trust score between a pair of users. This method finds all possible paths between the two users, exploring *On2Trust*. Finally, it aggregates the trust information represented in the most relevant paths found between the pair of users.

To evaluate the performance of our system, we have developed several experiments based on off-line tests. Specifically, we have compared different configurations of the system parameters, and we also compare the new recommendation approach with the collaborative ones. The obtained results reveals an improvement over previous proposals.

Further research is warranted to explore other techniques to improve the recommendation approach. For instance, automatic techniques to establish the user profiles in a more efficient manner could be analyzed. Other proposal might be to focus on applying specific measures of the social networks analysis, exploiting the information represented in the developed ontologies, incorporating these measures into a new recommendation approach.

Acknowledgments

This paper has been developed with the financing of Projects UJA2013/08/41, TIN2013-40658-P, TIC5299, TIC-5991, TIN2012-36951 co-financed by FEDER and TIC6109.

References

[1] Movielens – Movie Recommendations. <http://movielens.umn.edu/login>.
[2] P. Agarwal, Ontological considerations in GIScience, Int. J. Geograph. Inform. Sci. 19 (5) (2005) 501–536.
[3] S. Alonso, I. Pérez, F. Cabrerizo, E. Herrera-Viedma, A linguistic consensus model for web 2.0 communities, Appl. Soft Comput. 13 (1) (2013) 149–157.
[4] A. Barragáns-Martínez, E. Costa-Montenegro, J. Burguillo, M. Rey-López, F. Mikic-Fonte, A. Peleteiro, A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition, Inform. Sci. 180 (22) (2010) 4290–4311.
[5] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Sci. Am. 284 (5) (2001) 28–37.
[6] Y. Blanco-Fernández, M. López-Nores, J. Pazos-Arias, J. Garca-Duque, An improvement for semantics-based recommender systems grounded on attaching temporal information to ontologies and user profiles, Eng. Appl. Artif. Intell. 24 (8) (2011) 1385–1397.
[7] Y. Blanco-Fernández, M. López-Nores, J.J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, Exploiting digital tv users' preferences in a tourism recommender system based on semantic reasoning, IEEE Trans. Consum. Electron. 56 (2) (2010) 904–912.
[8] F. Bobillo, U. Straccia, Fuzzy ontology representation using owl 2, Int. J. Approx. Reason. 52 (7) (2011) 1073–1094.
[9] R. Burke, Hybrid recommender systems: survey and experiments, User Model. User-Adapt. Interact. 12 (2002) 331–370.
[10] R. Burke, Hybrid web recommender systems, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, LNCS, vol. 4321, 2007, pp. 377–408.
[11] S. Calegari, D. Ciucci, Towards a fuzzy ontology definition and a fuzzy extension of an ontology editor, in: Y. Manolopoulos, J. Filipe, P. Constantopoulos, J. Cordeiro (Eds.), Enterprise Information Systems, 8th International Conference, ICEIS 2006, Paphos, Cyprus, May 23–27, 2006, Revised Selected Papers, Lecture Notes in Business Information Processing, vol. 3, Springer, 2006, pp. 147–158.
[12] I. Cantador, A. Bellogín, P. Castells, A multilayer ontology-based hybrid recommendation model, AI Commun. 21 (2–3) (2008) 203–210.
[13] S.-M. Chen, Y.-H. Huang, R.-C. Chen, S.-W. Yang, T.-W. Sheu, Using fuzzy reasoning techniques and the domain ontology for anti-diabetic drugs recommendation, in: Proceedings of the 4th Asian Conference on Intelligent Information and Database Systems, ACIIDS'12, vol. Part I, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 125–135. <http://dx.doi.org/10.1007/978-3-642-28487-8_13>.
[14] F. Chiclana, E. Herrera-Viedma, F. Herrera, S. Alonso, Some induced ordered weighted averaging operators and their use for solving group decision-making problems based on fuzzy preference relations, Euro. J. Oper. Res. 182 (1) (2007) 383–399.
[15] L. Duen-Ren, L. Chin-Hui, L. Wang-Jung, A hybrid of sequential rules and collaborative filtering for product recommendation, Inform. Sci. 179 (2009) 3505–3519.
[16] M. Ferreira-Satler, F. Romero, V. Menendez-Dominguez, A. Zapata, M. Prieto, Fuzzy ontologies-based user profiles applied to enhance e-learning activities, Soft Comput. 16 (2012) 1129–1141.
[17] J. Golbeck, Computing and Applying Trust in Web-Based Social Networks, Ph.D. thesis, 2005.
[18] A. Gómez-Pérez, M. Férnandez-López, O. Corcho-García, Ontological Engineering, Springer-Verlag, Inc., New York, 2003.
[19] N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, J. Riedl, Combining collaborative filtering with personal agents for better recommendations, in: Proceedings of the Sixteenth National Conference on Artificial Intelligence, 1999, pp. 439–446.
[20] T.R. Gruber, Toward Principles for the Design of Ontologies used for Knowledge Sharing, Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University.
[21] N. Guarino, Formal ontology, concept analysis and knowledge representation, Int. J. Human Comput. Stud. 43 (5/6) (1995) 625–640.
[22] U. Hanani, B. Shapira, P. Shoval, Information filtering: overview of issues, research and systems, User Model. User-Adap. Interact. 11 (2001) 203–259.
[23] F. Herrera, S. Alonso, F. Chiclana, E. Herrera-Viedma, Computing with words in decision making: foundations, trends and prospects, Fuzzy Optim. Decis. Making 8 (4) (2009) 337–364.
[24] F. Herrera, L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, IEEE Trans. Fuzzy Syst. 8 (6) (2000) 746–752.
[25] F. Herrera, L. Martínez, A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in multiexpert decision-making, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 31 (2) (2001) 227–234.
[26] E. Herrera-Viedma, G. Pasi, A. López-Herrera, C. Porcel, Evaluating the information quality of Web sites: a qualitative methodology based on fuzzy computing with words, J. Am. Soc. Inform. Sci. Technol. 57 (4) (2006) 538–549.
[27] C. Lee, M. Wang, H. Hangras, A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation, IEEE Trans. Fuzzy Syst. 18 (2) (2010) 374–395.
[28] P. Lops, M. Gemmis, G. Semeraro, Recommender systems handbook, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Content-based Recommender Systems: State of the Art and Trends, Springer, 2011, pp. 73–106.
[29] C. Martinez-Cruz, I. Blanco, M. Vila, Ontologies versus relational databases: are they so different? A comparison, Artif. Intell. Rev. 38 (2012) 271–290.
[30] P. Massa, P. Avesani, Computing with social trust, in: Trust Metrics in Recommender Systems, Springer, 2009, pp. 259–285.
[31] S. Massanet, J. Riera, J. Torrens, E. Herrera-Viedma, A new linguistic computational model based on discrete fuzzy numbers for computing with words, Inform. Sci. 258 (2014) 277–290.
[32] F. Mata, L. Martínez, E. Herrera-Viedma, An adaptive consensus support model for group decision making problems in a multi-granular fuzzy linguistic context, IEEE Trans. Fuzzy Syst. 17 (2) (2009) 279–290.
[33] G. Meghabghab, A. Kandel, Search Engines, Link Analysis, and User's Web Behavior, Springer-Verlag, Berlin, Heidelberg, 2008.
[34] S. Middleton, D. Roure, N. Shadbolt, Handbook on ontologies, in: Ontology-Based Recommender Systems, International Handbooks on Information Systems, 2009, pp. 779–796.
[35] S.E. Middleton, D.D. Roure, N.R. Shadbolt, Ontology-based recommender systems, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, International Handbooks on Information Systems, Springer, Berlin, Heidelberg, 2009, pp. 779–796. chap. 35.
[36] J. Morente-Molinera, I. Pérez, R. Ureña, E. Herrera-Viedma, On multi-granular fuzzy linguistic modelling in group decision making problems: a systematic review and future trends, Knowl. Based Syst. 74 (2015) 49–60.
[37] J. O'Donovan, B. Smyth, Trust in recommender systems, in: Proceedings of the 10th International Conference on Intelligent User Interfaces, 2005, pp. 167–174.
[38] L. Page, S. Brin, R. Motwani, T. Winograd, The Pagerank Citation Ranking: Bringing Order to the Web, Technical report, Stanford, USA, 1998.
[39] G. Pasi, R. Yager, Modeling the concept of majority opinion in group decision making, Inform. Sci. 176 (2006) 390–414.
[40] I. Pérez, R. Wikstrm, J. Mezei, C. Carlsson, E. Herrera-Viedma, A new consensus model for group decision making using fuzzy ontology, Soft Comput. 17 (9) (2013) 1617–1627.
[41] A. Popescul, L. Ungar, D. Pennock, S. Lawrence, Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, in: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI), San Francisco, 2001, pp. 437–444.
[42] C. Porcel, E. Herrera-Viedma, Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries, Knowl.-Based Syst. 23 (2010) 32–39.
[43] P. Refaeilzadeh, L. Tang, H. Liu, Cross-Validation, 2008.
[44] P. Reisnick, H. Varian, Recommender systems, Commun. ACM 40 (3) (1997) 56–59.
[45] T. Ruotsal, Methods and Applications for Ontology-Based Recommender Systems, Ph.D. thesis, Aalto University School of Science and Technology, Finland, 2010.
[46] T. Ruotsal, Methods and Applications for Ontology-Based Recommender Systems, Ph.D. thesis, Aalto University School of Science and Technology, Finland, 2010. <http://lib.tkk.fi/Diss/2010/isbn9789526031514/isbn9789526031514.pdf>.
[47] J. Schafer, D. Frankowski, J. Herlocker, S. Sen, The adaptive web, in: Collaborative Filtering Recommender Systems, Springer, Berlin, Heidelberg, 2007, pp. 291–324.

[48] G. Shani, A. Gunawardana, Recommender systems handbook, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Evaluating Recommendation Systems, Springer, 2011, pp. 257–298.
[49] A. Sieg, B. Mobasher, R. Burke, Improving the effectiveness of collaborative recommendation with ontology-based user profiles, in: Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec '10, ACM, New York, NY, USA, 2010, pp. 39–46. <http://doi.acm.org/10.1145/1869446.1869452>.
[50] V. Sieg, P. Shoval, B. Shapira, M.T. Maimon, Evaluation of an ontology-content based filtering method for a personalized newspaper, in: RecSys '08: Proceedings of the 2008 ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2008, pp. 91–98.
[51] R. Sinha, K. Swearingen, Comparing recommendations made by online systems and friends, in: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries, 2001, pp. -1–1.
[52] R. Studer, V. Benjamins, D. Fensel, Knowledge engineering: principles and methods, IEEE Trans. Data Knowl. Eng. 25 (1–2) (1998) 161–197.
[53] A. Tejeda-Lorente, C. Porcel, E. Peis, R. Sanz, E. Herrera-Viedma, A quality based recommender system to disseminate information in a university digital library, Inform. Sci. 261 (2014) 52–69.
[54] P. Victor, C. Cornelis, M. DeCock, P. Pinheiro da Silva, Gradual trust and distrust in recommender systems, Fuzzy Sets Syst. 160 (10) (2009) 1367–1382.
[55] J. Wu, F. Chiclana, A social network analysis trust-consensus based approach to group decision-making problems with interval-valued fuzzy reciprocal preference relations, Knowl.-Based Syst. 59 (2014) 97–107.
[56] R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Trans. Syst. Man Cybernet. 18 (1) (1988) 183–190.
[57] S. Yang, Developing an ontology-supported information integration and recommendation system for scholars, Exp. Syst. Appl. 37 (2010) 7065–7079.
[58] L. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Part I, Inform. Sci. 8 (1975) 199–249;
L. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Part II, Inform. Sci. 8 (1975) 301–357;
L. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Part III, Information Sciences 9 (1975) 43–80.
[59] C. Ziegler, Towards Decentralized Recommender Systems, Ph.D. thesis, 2005.