Information Sciences 181 (2011) 4823–4846

Contents lists available at ScienceDirect

Information Sciences

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

Exploring synergies between content-based filtering and Spreading Activation techniques in knowledge-based recommender systems ☆

Yolanda Blanco-Fernández *, Martín López-Nores, Alberto Gil-Solla, Manuel Ramos-Cabrer, José J. Pazos-Arias

ETSE Telecomunicación, Campus Universitario, Vigo 36310, Spain

ARTICLE INFO

Article history:
Received 21 September 2009
Received in revised form 9 June 2011
Accepted 10 June 2011
Available online 23 June 2011

Keywords:
Personalization
Content-based filtering
Semantic reasoning
Spreading Activation techniques

ABSTRACT

Recommender systems fight information overload by selecting automatically items that match the personal preferences of each user. The so-called content-based recommenders suggest items similar to those the user liked in the past, using syntactic matching mechanisms. The rigid nature of such mechanisms leads to recommending only items that bear strong resemblance to those the user already knows. Traditional collaborative approaches face up to overspecialization by considering the preferences of other users, which causes other severe limitations. In this paper, we avoid the intrinsic pitfalls of collaborative solutions and diversify the recommendations by reasoning about the semantics of the user's preferences. Specifically, we present a novel content-based recommendation strategy that resorts to semantic reasoning mechanisms adopted in the Semantic Web, such as Spreading Activation techniques and semantic associations. We have adopted these mechanisms to fulfill the personalization requirements of recommender systems, enabling to discover extra knowledge about the user's preferences and leading to more accurate and diverse suggestions. Our approach is generic enough to be used in a wide variety of domains and recommender systems. The proposal has been preliminary evaluated by statistics-driven tests involving real users in the recommendation of Digital TV contents. The results reveal the users' satisfaction regarding the accuracy and diversity of the reasoning-driven content-based recommendations.

1. Introduction

Recommender systems provide personalized advice to users about items they might be interested in. These tools are already helping people efficiently manage content overload and reduce complexity when searching for relevant information. To fulfill these personalization needs, three main components are required: (i) a database that stores characterizations of the available items, (ii) profiles that model the users' preferences, and (iii) recommendation strategies that make personalized suggestions to each individual.

The first recommendation strategy was *content-based filtering* [41,30], which consists of suggesting items similar to those the user liked in the past. In spite of its accuracy, this technique is limited due to the similarity metrics employed, which are based on rigid syntactic approaches that can only detect similarity between items that share all or some of their attributes [1]. Consequently, traditional content-based approaches lead to *overspecialized suggestions* including only items that bear strong resemblance to those the user already knows (i.e. items bound to the attributes defined in his/her profile).

☆ Work funded by the Ministerio de Educación y Ciencia (Gobierno de España) Research Project TIN2010-20797.
* Corresponding author.
E-mail address: yolanda@det.uvigo.es (Y. Blanco-Fernández).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.06.016

In order to fight overspecialization, researchers devised *collaborative filtering* [36,25,29] – whose basic idea is to move beyond the experience of an individual user's profile and instead draw on the experiences of a community of like-minded users (his/her neighbors), and even they combined content-based and collaborative filtering in *hybrid approaches* [6,22,33,13,40]. Even though collaborative (and hybrid) approaches mitigate the effects of *overspecialization* by considering the interests of other users, they bring in new limitations, such as the *sparsity problem* (related to difficulties to select each individual's neighborhood when the knowledge about the users' preferences is sparse), privacy concerns bound to the confidentiality of the users' personal data, and scalability problems stemmed from the management of many user profiles (instead of just one profile like in content-based approaches).

The contribution of our paper is a *content-based strategy* that diversifies the recommendations by exploiting semantic reasoning about the user's interests, instead of considering other individuals' preferences. This way, we overcome the *overspecialization* effects without suffering from the intrinsic limitations of collaborative and hybrid solutions. Specifically, our reasoning mechanisms have been borrowed from the area of the Semantic Web, an initiative that is based on (i) annotating Web resources by semantic annotations (metadata), (ii) formalizing this knowledge in a domain ontology that represents concepts and relationships by classes and properties, respectively, and (iii) carrying out reasoning processes about the ontology in order to infer semantic relationships among the annotated resources.

Broadly speaking, our content-based strategy suggests items which are semantically related to the user's preferences, instead of offering items with *the same* attributes that appear in his/her profile. For example, in the TV field, a viewer who has enjoyed documentaries about *traveling* and *archeology* might receive as recommendations programs about *potholing* (a hobby strongly related to the study of ancient graves) or about *Greece* (a country of deep-rooted archeological tradition). Our domain-independent strategy consists of two stages that adopt *semantic associations* [4] and *Spreading Activation techniques* (henceforth, *SA techniques*) [14] as reasoning mechanisms:

(1) Firstly, the **pre-filtering phase** selects an excerpt from the domain ontology that comprises only instances of classes and properties that are significant for the user (because they are closely related to his/her preferences). For that reason, this excerpt is named the user's *Ontology of Interest*. Then, we infer hidden semantic associations among the items included in the user's Ontology of Interest, starting from the hierarchical relationships and properties formalized in it.
(2) Next, the **recommendation phase** processes the discovered knowledge and provides the personalized recommendations. To this aim, we emphasize the use of SA techniques as computational mechanisms able to explore efficiently a generic network with nodes interconnected by links, and to detect concepts that are strongly related to each other. In our approach, the considered network corresponds to the user's Ontology of Interest, while the strongly related nodes are his/her preferences and the items to be suggested.

The filtering criteria employed to delimit the user's Ontology of Interest have been described in detail in [9]. For that reason, here we focus on the second phase of our strategy. Specifically, our main research contribution consists of extending traditional SA techniques so that the personalization requirements of a recommender system can be considered. To this aim, our improved SA techniques must fulfill the following requirements:

- Firstly, our SA mechanisms must enable our strategy to discover useful knowledge for the recommendation process by reasoning about the semantics of the user's Ontology of Interest.
- Secondly, the knowledge inferred by the SA mechanisms must serve to increase the diversity of the offered content-based recommendations.
- Lastly, our SA approach must learn automatically the user's preferences from the feedback provided after recommendations, and thereafter update conveniently his/her personal profile. This way, our reasoning-based suggestions evolve as the user's preferences change over time, thus reinforcing his/her confidence in our personalization strategy.

This paper is organized as follows. The next two sections provide necessary background to understand our approach: Section 2 explains internals of semantic associations and highlights the limitations of traditional SA techniques for personalization purposes, while Section 3 presents the two essential components of our reasoning framework: the domain ontology and the user profiles. Next, Section 4 details the internals of our two-phase recommendation strategy, exploring synergies between our improved SA techniques and content-based filtering in the selection of *diverse* recommendations. Afterwards, Section 5 provides an example of our strategy in the scope of Digital TV, where we highlight how to exploit our reasoning capabilities to select TV programs among the myriad available in the digital stream. Next, Section 6 presents the experimental evaluation of our approach and discusses scalability and computational feasibility concerns. Finally, Section 7 summarizes the conclusions from our work and motivates possible lines of further research.

2. Background on semantic reasoning

In this section, we describe the internals of the semantic reasoning mechanisms exploited in our recommendation strategy: semantic associations and SA techniques. Very briefly, the associations allow to interrelate the items available in the

recommender system, whereas the SA techniques serve to discover new knowledge about the users' preferences from the inferred associations and the concepts formalized in the domain ontology.

2.1. Semantic associations

The semantic associations employed in our reasoning approach have been borrowed from [4], where Anyanwu and Sheth defined the relationships that can be established between two specific class instances in an ontology. In order to categorize these associations, they resorted to a structure named *property sequence*, which consists of a set of class instances linked to each other by means of properties. The first class instance defined in the sequence is the *origin*, the last one is the *terminus*, and the *length* of the sequence is the number of properties included in it. The semantic associations defined in [4] are defined next with the aid of Fig. 1:

- *ρ-path association*. Two class instances i_1 and i_5 are *ρ-pathAssociated* in an ontology if it is possible to find a property sequence whose origin is i_1 and whose terminus is i_5 (or vice versa). Obviously, the longer the property sequence linking both class instances, the less significant the relationship between them, due to the presence of many intermediate nodes.
- *ρ-join association*. Two class instances are *ρ-joinAssociated* if both are origins (e.g. i_1 and i_6 in Fig. 1) or terminus (i_5 and i_8) of two property sequences containing instances belonging to a common class C (named the *union class*).

2.2. Spreading Activation techniques

SA techniques are computational mechanisms able to efficiently explore huge generic networks of nodes interconnected by links. According to the guidelines established in [14], these techniques work as follows:

- Each node is associated to a weight (called the *activation level*) that grows with its relevance in the network: the more relevant the node, the higher its activation level. Besides, each link joining two nodes has a *weight* whose value is proportional to the strength of the relationship existing between both nodes.
- Initially, a set of nodes are selected and the nodes connected with them by links (named *neighbor nodes*) are activated. In this process, the activation levels of the initially selected nodes are spread until reaching their neighbors in the network.
- The activation level of a reached node is typically computed by considering the levels of its neighbors and the weights assigned to the links that join them to each other. Consequently, the more relevant the neighbors of a given node (i.e. the higher their activation levels) and the stronger the relationship between the node and its neighbors (i.e. the higher the weights of the links between them), the more relevant the node will be in the network.
- The spreading process is repeated until reaching all the nodes of the network. In the end, the highest activation levels correspond to the nodes that are most closely related to the initially selected ones.

Since the spreading process permits to reach nodes that are not directly joined to the initially selected ones, SA techniques carry out inference processes where new knowledge is learned. To harness these inferential capabilities, several algorithms have been proposed for exploration and extraction of the most significant concepts formalized in a knowledge network. In

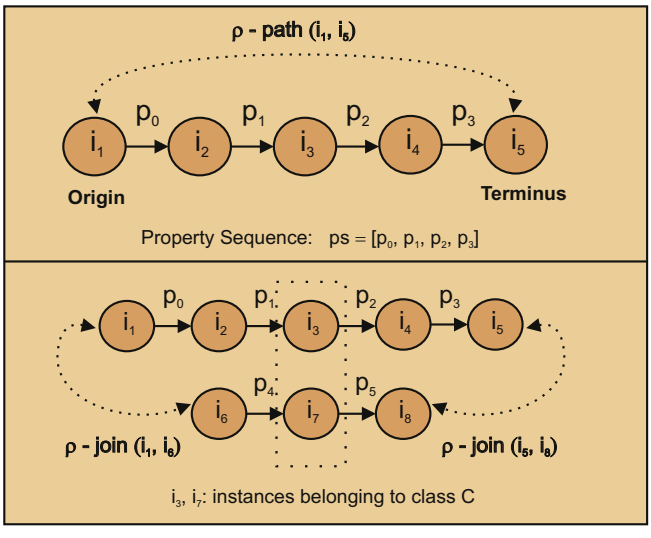

Fig. 1. Semantic associations adopted in our reasoning-driven approach.

literature, many applications resort to the so-called *Hopfield Net* algorithm due to its beneficial properties of search and knowledge discovery, as explained in [23].

2.2.1. The Hopfield Net algorithm

The Hopfield Net algorithm is based on a neural network that provides two capabilities especially relevant for the spreading process: parallel search and convergence (see [12] for details). On the one hand, the search capabilities allow the algorithm to activate in each iteration all the nodes of the network in parallel, computing their activation levels according to the levels of the remaining nodes in the network. On the other, the algorithm Hopfield Net traverses successively the nodes (iteration by iteration) until their activation levels converge to a stable value. The internals are as follows:

- Firstly, a value 1 is assigned as the activation level for the initially activated nodes, and a value 0 is established for the remaining nodes of the network.
- Next, the initial activation levels are spread through the network, and the levels corresponding to all the nodes are computed by using the sigmoid function (f_S included in Eq. (1)):

$$A_j(t+1) = f_S\left(\sum_{i=0}^{n-1} A_i(t)\cdot w_{ij}\right), \quad 0 \leqslant j \leqslant n-1 \tag{1}$$

In this expression:

- $A_j(t+1)$ is the activation level of the node j in iteration $t+1$,
- $A_i(t)$ is the activation level of the node i in iteration t,
- n is the number of nodes in the network,
- w_{ij} is the weight of the link between the nodes i and j, being $w_{ij} = 0$ if there does not exist a link between them in the network,
- $f_S(x) = \frac{1}{1+exp\left[\frac{\theta_1 - x}{\theta_2}\right]}$, where θ_1 is a configurable threshold, and θ_2 is a parameter used to modify the shape of the sigmoid function $f_S(x)$.

- The spreading process is repeated until the activation level of all the nodes reach a stable value, as indicated by Eq. (2), where ξ is a configurable parameter taking very low values

$$\sum_{j=0}^{n-1} |A_j(t+1) - A_j(t)| \leqslant \xi \tag{2}$$

2.2.2. Limitations of traditional SA techniques in personalization field

We have identified two severe drawbacks that prevent us from exploiting the inferential capabilities of traditional SA techniques in our reasoning-driven recommendation strategy. These drawbacks lie within (i) the kind of links modeled in the considered network and (ii) the weighting processes of those links.

- On the one hand, the kind of the modeled links is closely related to the richness of the reasoning processes carried out during the spreading process. These links establish paths to propagate the relevance of the initially activated nodes to other nodes closely related to them. For that reason, it is possible that some significant nodes never be detected, due to the absence of links reaching them in the network. Existing SA techniques (see examples in [32,35,23,37]) model very simple relationships, which lead to poor inferences and prevent from discovering the knowledge hidden behind more complex associations.
- The second limitation of traditional SA approaches is related to the weighting processes of the links modeled in the network. According to the guidelines described in Section 2.2, these weights remain invariable over time, because their values depend either on the existence of a relationship between the two linked nodes or on the strength of this relationship. This static weighting process is not appropriate for our personalization process, where it is necessary that the weights assigned to the links of the user's network enable to: (i) learn automatically his/her preferences from the feedback provided after recommendations and (ii) adapt dynamically the spread-based inference process as these preferences evolve.

In Section 4, we will explain how our reasoning-driven approach fights above limitations by extending traditional SA techniques so that they can be adopted in a content-based recommender system. Prior to that, the next section describes the procedures we have followed to formalize the domain ontology and to model the user profiles.

3. Background on our reasoning-driven personalization framework

3.1. The domain ontology

In the field of the Semantic Web, an ontology characterizes the concepts typical in a domain and their relationships by means of classes and properties, respectively, which are organized hierarchically [8]. Besides, the ontology is populated

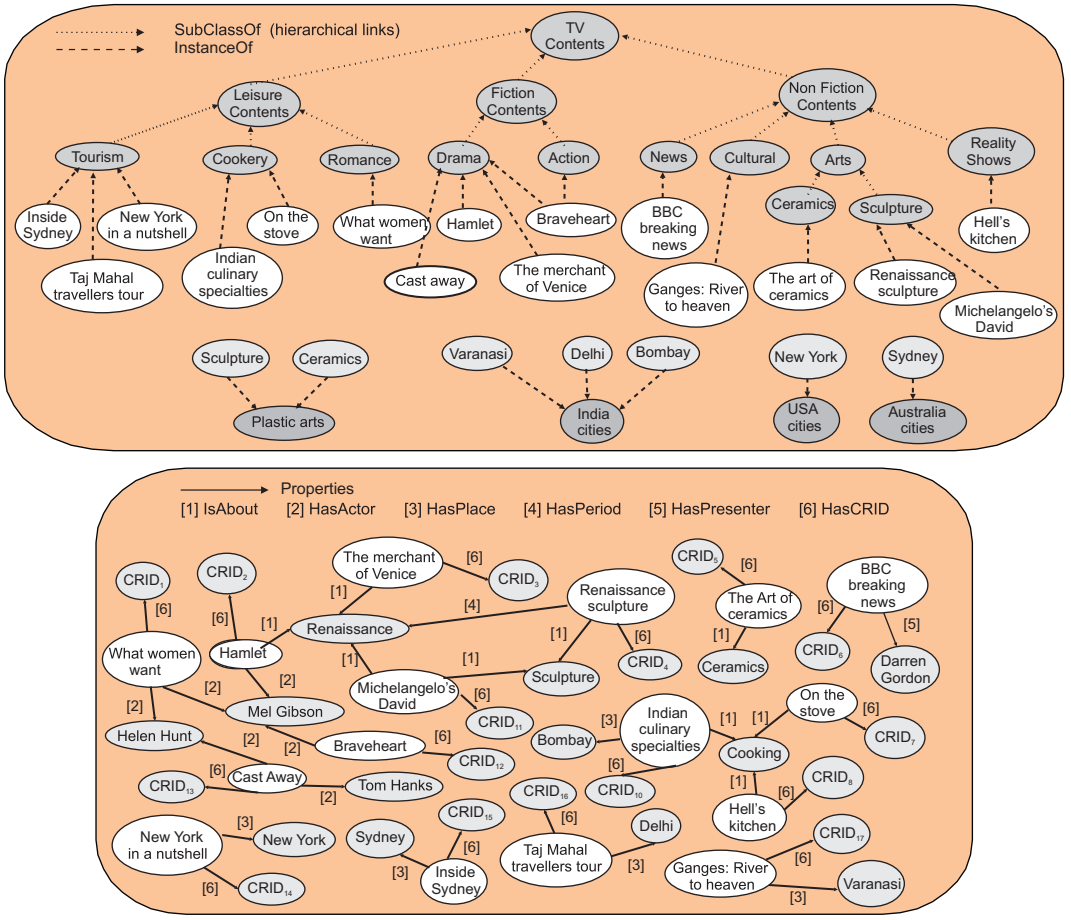

Fig. 2. Subset of classes (top), properties and specific instances (bottom) defined in an ontology about the TV domain.

by including specific instances of classes and properties. In the context of a recommender system, class instances represent the available items and their attributes, whereas property instances link items and attributes to each other. We depict in Fig. 2 a brief excerpt from an ontology for the TV domain, defined from the TV-Anytime specification (a collection of metadata providing detailed descriptions about generic audiovisual contents [38]). In this figure, it is possible to identify several class instances referred to specific TV programs, which belong to a hierarchy of genres (e.g. *Fiction*, *Sports*, *Music*, *Leisure*). The attributes of these TV contents (e.g. cast, intented audience, topics) are also identified by hierarchically-organized classes, and related to each program by means of labeled properties (e.g. *hasActor*, *hasIntendedAudience*, *isAbout*).

Ontologies have become the cornerstone of the Semantic Web due to two reasons. On the one hand, formal conceptualizations enable inference processes to discover new knowledge from the represented information. On the other, ontologies facilitate automated knowledge sharing, by allowing easy reuse between users and software agents. This feature facilitates the development of ontologies, which would be a tedious task otherwise. Nowadays, there exist repositories containing multiple and very diverse ontologies (e.g. *SchemaWeb*[1]), as well as numerous management tools providing useful functionalities for development tasks (e.g. merging of multiple ontologies, consistency checking, discovery of equivalent classes, reuse of concept descriptions, automatic categorization of instances in the appropriate classes via logics-based reasoners [3,17], etc.). In sum, by reusing the concepts and relationships formalized in publicly available ontologies and resorting to the existing management tools, it is possible to create a domain ontology for reasoning-purposes with acceptable effort.

There exist several standard implementation languages for ontology development. The first proposals were RDF [7] and RDFS [10], which added a formal semantics to the purely syntactic specifications provided in XML. Next, DAML [15] and OIL [18] arose, which have been finally fused and standardized by W3C as OWL [26], the most expressive language nowadays including three sub-levels (Lite, DL and Full). The language to use in the application of our reasoning-driven approach depends on the knowledge and expressiveness necessities of the domain considered and the recommender system.

[1] Available in http://www.schemaweb.info/schema/BrowseSchema.aspx.

3.2. User modeling technique

Reasoning about a user's preferences requires a formal representation including semantic descriptions of the items that are appealing or unappealing to him/her (named *positive* and *negative preferences*, respectively). These descriptions permit a recommender system to learn new knowledge about the user's interests, which is not possible with many of the existing user modeling techniques:

- Some existing works define too simple user models, containing only flat lists of key words (e.g. attributes) or ratings referred to each item defined in the user's profile [11,24,39]. These proposals provide little knowledge about the user's preferences, and therefore hamper the application of advanced reasoning processes.
- Other more sophisticated proposals take advantage of the hierarchical structures defined in an ontology to model the user's preferences [27,42,19]. In these works, profiles do not contain the specific items the user (dis)liked in the past, but the classes under which these items are categorized in a hierarchy. The main drawback of this approach is that it only explores the hierarchical structure of the domain and misses the semantic descriptions of the items, which are especially useful for user modeling tasks and for subsequent reasoning processes, as we will describe through the paper.

Bearing in mind that the descriptions required in our reasoning mechanisms are already defined in the domain ontology, we propose to model the user's preferences by reusing the knowledge formalized in it. The resulting models are named *ontology-profiles* and store the interest of the user in: (i) the attributes of the items which are (un)interesting for him/her and (ii) the hierarchy of classes under which these items are categorized. This approach has two main advantages for a recommender system:

- On the one hand, the formal representation of the user's profile allows the system to reason and compare effectively his/her preferences against the available items, thus favoring more accurate personalization processes.
- On the other hand, we provide the system with a very detailed model of the user's interests, while not requiring that the classes, properties and instances that identify these preferences be stored in each profile. Thus, we significantly reduce the storage capabilities needed in the reasoning-driven recommender system. To this aim, we use the domain ontology as a common knowledge repository, keeping only two elements in the user's profile: unique references (denoted by IDs)

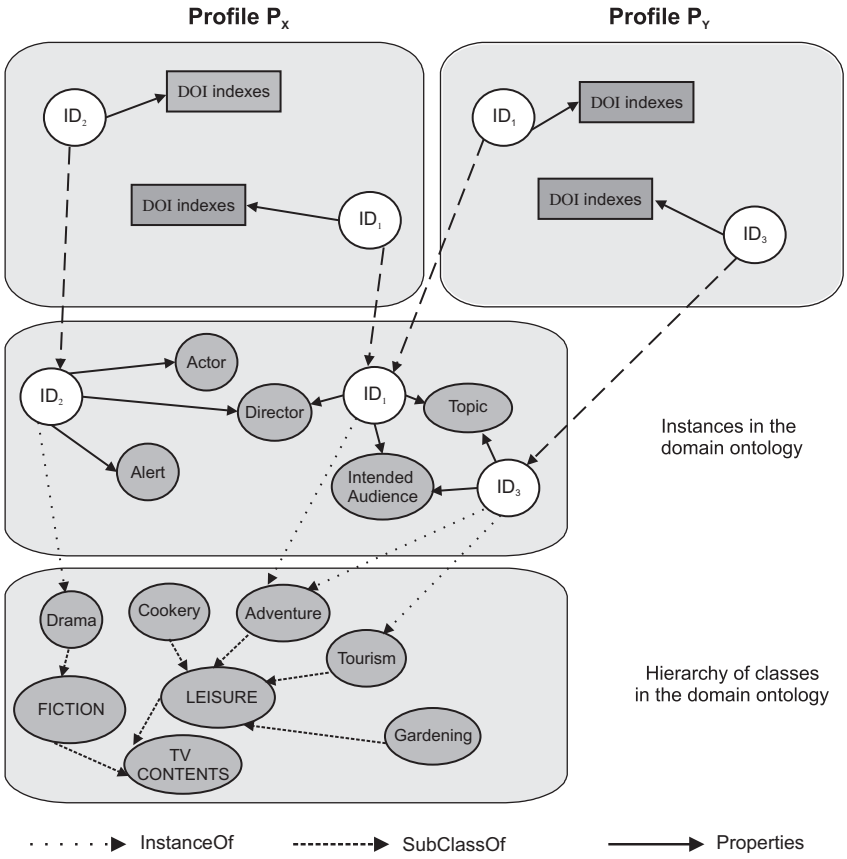

Fig. 3. Our ontology-based approach for modeling user in a TV recommender system.

that identify the items the user (dis)liked, and his/her specific level of interest in each one of them. These references permit to locate in the ontology the items defined in the user's profile and to query their semantic descriptions (i.e. attributes and hierarchical classes) over the conceptualization, as shown in Fig. 3 for a recommender system in TV domain.

Note that our modeling technique does not consider a flat list of attributes referred to the user's preferences, but rather it exploits the structure of the domain ontology and the relationships existing among these attributes in order to learn knowledge about his/her interests and exploit it during the personalization process.

Obviously, recommender systems require the users to define some initial preferences to start working. Considering the users' involvement, the goal is to provide a user-friendly interface to alleviate their initialization burden. Our user modeling technique exploits the hierarchical structure of the underlying ontology for that purpose. Specifically, a list of classes/subclasses and specific instances referred to the items to be recommended (e.g. programs in the TV domain) is shown to the user, who can identify his/her positive and negative preferences by assigning ratings to each specific item. The hierarchy of classes displayed is self-explanatory (see bottom of Fig. 3), so that the users can easily browse it and feel free to rate as items as they want.

After the profile initialization, it is necessary to measure the user's level of interest in each item included in his/her profile. To this aim, we have defined the so-called DOI indexes (*Degree Of Interest*) in the range $[-1,1]$, with -1 representing the greatest disliking and 1 the greatest liking. These indexes can be either explicitly entered by the user or inferred automatically by the recommender system from the relevance feedback provided after recommendations. The DOI index computed for each item is also used to set the ratings corresponding to its attributes and to the classes under which the item is categorized in the ontology. Specifically, the DOI of an attribute is taken as the average of the DOIs of the items it is linked to. Similarly, the DOIs of the most specific classes are computed as the average of the DOIs of the items classified under them. Then, we propagate these values upwards in the hierarchy until reaching its root class. For that purpose, we adopt the approach proposed in [42], which leads to higher DOI indexes for the superclasses closer to the leaf class whose value is being propagated, and lower ones for the classes which are closer to the root of the hierarchy. Besides, the higher the DOI of a given class and the lower its number of siblings, the higher the value propagated to its superclass.

4. Using content-based filtering in tandem with SA techniques

As we mentioned in the introduction, our content-based strategy is divided into two phases named *pre-filtering* and *recommendation phase*. Even though the pre-filtering phase has been detailed in [9], in this section we summarize the main aspects of this process with the goal of clarifying how the user's Ontology of Interest is selected (Section 4.1) and how it is processed by SA techniques in the recommendation phase of the strategy (Sections 4.2, 4.3, 4.4). Regarding SA techniques, we extend traditional approaches by overcoming the limitations pointed out in Section 2.2.2, which hamper their adoption in a recommender system where the focus must be put on the user's preferences:

- On the one hand, our approach extends the simple relationships considered by traditional SA techniques by considering both the properties defined in the ontology and the semantic associations inferred from them. This rich variety of relationships permit to establish links that propagate the relevance of the items selected by the pre-filtering phase, leading to diverse enhanced recommendations.
- On the other hand, to fulfill the personalization requirements of a recommender system, our link weighting process does not depend only on the two nodes joined by the considered link, but also on (the strength of) their relationship to the items defined in the user's profile. This way, the links of the network created for the user are updated as our strategy learns new knowledge about his/her preferences, thus leading to tailor-made recommendations after the spreading process.

Once the principles of our SA approach have been sketched, we focus on the processes required for its use in our content-based strategy: (i) selection of the user's Ontology of Interest, (ii) creation of the user's SA network, (ii) weighting of its links, (iii) processing of the network by SA techniques, and (iv) selection of our reasoning-based recommendations.

4.1. Pre-filtering phase: creating the user's Ontology of Interest

Our pre-filtering phase decides which instances of classes and properties from the domain ontology must be included in the user's Ontology of Interest because they are relevant for him/her. For that purpose, we firstly locate in the domain ontology the items that are (un)appealing to the user (defined in his/her profile). Next, we traverse successively the properties bound to these items until reaching new class instances in the ontology, referred to other items and their attributes. In order to guarantee computational feasibility, we have developed a controlled inference mechanism that progressively filters the instances of classes and properties that do not provide useful knowledge for the personalization process:

- As new nodes are reached from a given instance, we firstly quantify their relevance for the user by an index named *semantic intensity* (denoted by $\lambda_{Sem}(n)$ for node n), whose computation process will be described in this section.

- Next, the nodes whose intensity indexes are not greater than a specific threshold are disregarded, so that our inference mechanism continues traversing only the properties that permit to reach new nodes from those that are relevant for the user.

In order to measure the semantic intensity of a node *n*, we take into account various ontology-dependent *pre-filtering criteria*, so that the more significant the relationship between a given node and the user's preferences, the higher the resulting value. Some of these criteria (described in detail in [9]) are summarized next:

(1) *Length of the property sequence that enables to reach the node starting from the user's preferences*. The longer this sequence, the lower the semantic intensity of the node because its relationship to the user's preferences is less significant due to the presence of many intermediate nodes.
(2) *Existence of hierarchical relationships between the node and the user's preferences*. The intensity of a node increases when it is possible to find a common ancestor between it and the user's preferences in the hierarchies defined in the ontology.
(3) *Existence of implicit relationships between the node and the user's preferences detected by graph theory betweenness*. In graph theory [16], the betweenness among three nodes is high when in the most of paths from the first node to the second one, the third node is also included. Therefore, from a high value of betweenness, it follows that the involved nodes are strongly related. In our approach, these nodes are the user's preferences and the class instance whose relevance is being measured.

Once the nodes related to the user's preferences (and also the properties linking them to each other) have been selected, our strategy infers semantic associations between the instances referred to items that can be recommended. As per the categorization of semantic associations described in Section 2.1, we detect the following relationships between the items defined in the user's Ontology of Interest:

- First, ρ-*path* associations between the items that are joined by a property sequence in the Ontology of Interest, as it happens with the programs *Hell's kitchen* and *Indian culinary specialties* in Fig. 2, which are linked by the instance *cooking* in the ontology.
- Second, ρ-*join* associations between, for instance, the items whose attributes belong to a union class in the ontology. As an example, the programs *Renaissance sculpture* and *The Art of ceramics* in Fig. 2 are associated because both are about plastic arts strongly related to each other (as shown in the class hierarchy of the figure, *sculpture* and *ceramics* belong to the union class *Plastic arts*).

Starting from the user's Ontology of Interest and the semantic associations inferred among its nodes, we create the *user's SA network*, whose knowledge is explored during the second phase of the strategy by exploiting the inference capabilities provided by SA techniques.

4.2. Creation of the user's SA network

The user's SA network can be easily built starting from his/her Ontology of Interest. Specifically, the nodes of this network are the class instances selected by the pre-filtering phase of our strategy. The knowledge learned in this first phase also helps to identify the links that relate the nodes to each other, which permit to carry out the inference processes toward recommendations. In this regard, our SA approach defines two kind of links:

- *Real links*. These links model the knowledge that is explicitly represented in the user's Ontology of Interest. Specifically, we consider a real link in the user's SA network for each one of the property instances included in his/her Ontology.
- *Virtual links*. These links refer to relationships inferred from the Ontology of Interest. In this group, we include both simple hierarchical relationships and the complex semantic associations discovered from the properties and hierarchical links of the user's Ontology of Interest. According to the nature of both relationships, we identify two kind of virtual links:
  - *Associative virtual links*. We consider an associative virtual link between each pair of items related by ρ-*path* or ρ-*join* associations. For instance, from the associations depicted in Fig. 1, we define three associative virtual links: between items i_1 and i_5, due to the ρ-*path* association; between items i_1 and i_6, due to ρ-*join*; and between items i_5 and i_8, again due to ρ-*join*.
  - *Hierarchical virtual links*. We consider a hierarchical virtual link between the two instances belonging to the union class that causes ρ-*join* associations. For instance, in Fig. 1 it is possible to establish a virtual link between items i_3 and i_7, which are classified under the union class *C*.

We define a new type of structure (named *virtual path*) starting from ρ-*join* associations existing between two specific items. This structure permits to go from one item to the other by crossing a minimum number of real links and the hierarchical link that originates the ρ-*join* association between the two items. The length of the virtual path is defined as the number of real links contained in it. As an example, in Fig. 1 it is possible to find a virtual path (of length 3) between items i_1 and i_6, which

consists of the real links i_1–i_2, i_2–i_3, i_7–i_6 and the hierarchical link i_3–i_7. Analogously to what happens with property sequences, the shorter a virtual path between two items, the more relevant their relationship will be, due to the presence of few intermediate nodes between them.

4.3. Weighing links in the user's SA network

As we explained previously, incorporating the personalization requirements of our recommendation strategy into classic SA techniques requires to adapt the weighing process of the links modeled in the user's SA network. Instead of considering that the weight of a link between two nodes depends only on the strength of their mutual relationship, our approach imposes two constraints on the links to be weighed:

- First, given two nodes joined by a link, we consider that the stronger the (semantic) relationship between the two linked nodes and the user's preferences, the higher the weight of the link.
- Second, the weights are dynamically adjusted as the user's preferences evolve over time, thus offering permanently updated content-based recommendations.

In our approach, the weight of the links are assigned by combining two parameters: (i) the contribution of the two linked nodes, measured by their respective *relevance functions* and (ii) the type of link considered.

4.3.1. Relevance function of a node

The aim of the relevance function of a node in the SA network is to quantify its importance for the user, by considering his/her personal preferences and the knowledge learned from his/her Ontology of Interest. Eq. (3) shows how we compute the value of the relevance function for the node i which is linked to the node j (denoted by $f_j(i)$):

$$f_j(i) = \begin{cases} DOI_U(i) & \text{if } i \text{ is defined in } U\text{'s profile} \\ \lambda_{Sem}(i) & \text{otherwise} \end{cases} \tag{3}$$

- If the node i is defined in the user U's profile, the value of its relevance function $f_j(i)$ is its level of interest $DOI_U(i)$, since this is the most appropriate indicator to measure how relevant the node i is for the target user.
- Otherwise, $f_j(i)$ equals the value of the semantic intensity of the node i, so that the higher $\lambda_{Sem}(i)$, the most significant the relationship between i and the user's preferences, and therefore, the more relevant the node i is for him/her (remember Section 4.1).

4.3.2. Type of link to be weighed

The weights assigned to the virtual links are lower than those set for the real links. The intuition behind this idea is that the relationship existing between two nodes joined by a real link is explicitly represented in the user's Ontology of Interest by means of properties, while the relationship between two nodes joined by a virtual link has been inferred by a reasoning-driven prediction process. Thus, as established by Eq. (4), the weight of the link between nodes i and j is computed by combining an attenuation factor $\mu_{ij} \in [0,1)$ with the relevance values of both nodes.

$$w_{ij} = w_{ji} = \begin{cases} 0.5 \cdot (f_i(j) + f_j(i)) & \text{in case of real link} \\ 0.5 \cdot \mu_{ij} \cdot (f_i(j) + f_j(i)) & \text{in case of virtual link} \end{cases} \tag{4}$$

As shown in Eq. (5), the value of the factor μ_{ij} depends on the kind of virtual link established between nodes i and j. Specifically, the weights of the hierarchical virtual links are reduced by a factor 0.85 that prevents the contribution of these links from suffering an excessive decrease.[2] On the contrary, the value of μ_{ij} for the associative virtual links depends on the relevance of the semantic association inferred between the two linked nodes, so that the stronger the relationship between i and j, the higher the value of μ_{ij}. Specifically, we consider that the closer two nodes in the user's SA network, the stronger the association between them. The distance metric defined in our approach depends on the type of association inferred between the two linked items (i and j):

- In case of a ρ-*path* association, we use the length of the property sequence between i and j ($length(ps)$ in Eq. (5)) in order to measure the strength of the relationship. The higher the length of the sequence, the less significant the association and, therefore, the more severe the attenuation of the weight corresponding to the associative link between the two joined nodes.
- In case of a ρ-*join* association, we use the length of the virtual path existing between nodes i and j ($length(vpath)$), as shown in Eq. (5).

[2] The value 0.85 has been empirically adjusted after numerous experiments.

$$\mu_{ij} = \begin{cases} 0.85 & \text{for hierarchical virtual link} \\ \frac{1}{\hat{\mu}_{ij}} & \text{for associative virtual link} \end{cases}$$

$$\hat{\mu}_{ij} = \begin{cases} length(ps) & \text{if } \rho - path(i,j) \\ length(vpath) & \text{if } \rho - join(i,j) \end{cases} \quad (5)$$

From our weighting process, it follows that the weight of an associative virtual link between two nodes depends both on their relevance for the user (just like in real links and hierarchical virtual links) and on the kind of semantic associations inferred between both nodes. These two contributions permit to differentiate our personalization approach from other existing SA proposals where there is no place for complex relationships.

4.4. The processes of spreading activation and selection of recommendations

Once the knowledge learned about the user's preferences has been modeled in his/her SA network, we process the semantics of its nodes and links by an improved spreading activation mechanism:

- Firstly, we activate in the user's SA network the nodes referred to the items defined in his/her profile, considering both his/her positive and negative preferences. The positive preferences permit the spreading process to identify items that are significant for the user, because they are related to items he/she enjoyed in the past. The negative preferences lead to detect items that must not be suggested due to their relationships to unappealing items.
- Secondly, we assign the activation levels of all the nodes in the network. We use the *DOI* indexes defined in the user's profile for the nodes initially activated, and a value 0 for the remaining nodes.
- Next, the activation levels of the user's preferences are propagated through the SA network by using the Hopfield Net algorithm, which is in charge of selecting the items with high levels to be recommended to the user. The principle of parallel search of this algorithm is especially beneficial for our personalization approach, because the capability of activating in parallel all the nodes in the user's SA network permits to carry out the spreading process in an efficient way. Specifically, the algorithm computes the activation level of each node in the user's SA network by adding the contribution from *all* of its neighbor nodes. This contribution considers both the activation level of each neighbor node and the weight of the link (real or virtual) joining it to the considered node. For that reason, the more relevant the neighbors of a node (i.e. the higher activation levels) and the stronger the relationships among them and the considered node (i.e. the greater weights of links), the more significant this node will be for the user. This contribution is incorporated as an argument into the sigmoid function used by Hopfield Net (see Eq. (1)). As shown in Eq. (3), the weight of a link is computed starting from either the *DOI* indexes of the two joined nodes (if they are defined in the user's profile) or from their semantic intensity values (otherwise). Consequently, it holds that:
  - The sigmoid function measures the highest activation values for nodes which are connected both to class instances very appealing to the user (whose *DOI* indexes are very significant), and to nodes greatly related to his/her positive preferences (whose semantic intensity is very high).
  - Analogously, according to the internals of our content-based strategy, the sigmoid function quantifies low activation levels for class instances which are related to the users' negative preferences, thus preventing from suggesting these items.

Finally, our strategy selects the items to be suggested to the user. Specifically, the strategy recommends only the items of the user's SA network whose activation level is greater than a configurable threshold δ. This parameter is clearly dependent on the application domain and the recommender system that adopts our strategy. Anyway, the values must be always very high (close to 1) to guarantee that the items suggested are closely related to the user's preferences.[3]

5. A sample scenario

The research work of our content-based recommendation strategy has been tested in the scope of a TV recommender system (named **R-AVATAR**), which is being deployed over the cable networks of a Spanish operator with about 80,000 subscribers that broadcasts daily 43 TV channels. The goal of this system is to identify potentially appealing programs to each subscriber among the contents available in the digital stream. In this section, we illustrate how to select the TV programs that are most appealing to a user by considering the knowledge formalized in the excerpt from the TV ontology depicted in Fig. 2. Even though this ontology contains a reduced number of classes, properties and instances, it serves to highlight the differences between our reasoning-based recommendations and those offered by traditional (syntactic) content-based approaches. Assume a TV viewer *U* whose positive and negative preferences are shown in Table 1, including the TV programs *U* liked and disliked in the past, his/her *DOI* indexes (in brackets), and the classes under which these programs are categorized in the hierarchy of genres defined in the TV ontology.

[3] As a guidance, note that we have used the parameters $\theta_1 = 10$, $\theta_2 = 0.8$ and $\xi = 0.08$ (for the Hopfield Net algorithm), and recommendation thresholds δ in the range [0.78, 0.9] in the tests carried out in the Digital TV field.

Table 1
Positive and negative preferences of the viewer *U*.

Classes in genre hierarchy	Subclasses in genre hierarchy	TV programs (and DOI indexes)
Leisure	Tourism	*Inside Sydney* (1) *New York in a nutshell* (0.9)
	Cookery	*On the stove* (−0.9)
Non fiction	Cultural Art Reality Shows	*Ganges: River to heaven* (1) *Renaissance sculpture* (1) *Hell's kitchen* (−1)
Fiction	Drama	*Hamlet* (0.9)

According to the *DOI* indexes, *U* has enjoyed two *Leisure* contents about major tourist attractions in Sydney and New York, respectively. User *U* also liked a documentary about the Indian region of *Varanasi* (through which river Ganges flows), the *drama* movie *Hamlet*, and an art documentary about the *sculpture* in the *Renaissance*. Regarding *U*'s negative preferences, note that two TV programs about *cookery* were unappealing to this viewer.

Considering the TV programs available in the ontology depicted in Fig. 2, existing content-based strategies would only be able to suggest to *U* contents sharing the same attributes defined in his/her profile, like (i) the movie *The merchant of Venice*, set in the *Renaissance* period just like *Renaissance sculpture*; (ii) the movie *Braveheart*, which involves the actor *Mel Gibson* like *Hamlet*; and (iii) the documentary entitled *Michelangelo's David* because it shares topics with *Renaissance sculpture*. In contrast, we shall see how our two-phase strategy exploits semantic associations and improved SA techniques to select diverse suggestions based on reasoning.

5.1. Pre-filtering phase (I): selection of instances relevant for U

First, our strategy selects *U*'s Ontology of Interest by using the pre-filtering criteria described in Section 4.1. The result is depicted in Fig. 4.

- The instances *Varanasi*, *Sculpture*, *Renaissance*, *Mel Gibson*, *New York*, *Sydney* and *Cooking* are selected by the two first criteria, because all of them are directly related to *U*'s preferences by means of properties in the excerpt from the ontology shown at the bottom of Fig. 2. This fact increases the semantic intensity values computed by our pre-filtering mechanism.
- As shown in Table 1, *U* has enjoyed a cultural program about a plastic art closely related to ceramics. For that reason, the instance *Ceramics* is selected, because it shares the common ancestor *Plastic Arts* with the instance *Sculpture* stored in *U*'s preferences, as shown at the top of Fig. 2.
- The hierarchical relationships between *U*'s preferences and other class instances in the ontology permit to select the nodes *Delhi* and *Bombay*. Specifically, these instances belong to the same class as *Varanasi* (i.e. *India cities* in Fig. 2), a city linked to the documentary *Ganges: River to heaven* that *U* has enjoyed.
- Finally, the programs *Taj Mahal travelers tour*, *The merchant of Venice*, *Braveheart*, *Indian culinary specialties*, *On the stove* and *Michelangelo's David* are included in *U*'s Ontology of Interest because they share common ancestors with his/her preferences. These are *Tourism*, *Drama*, *Cookery* and *Arts*, as shown at the top of Fig. 2.

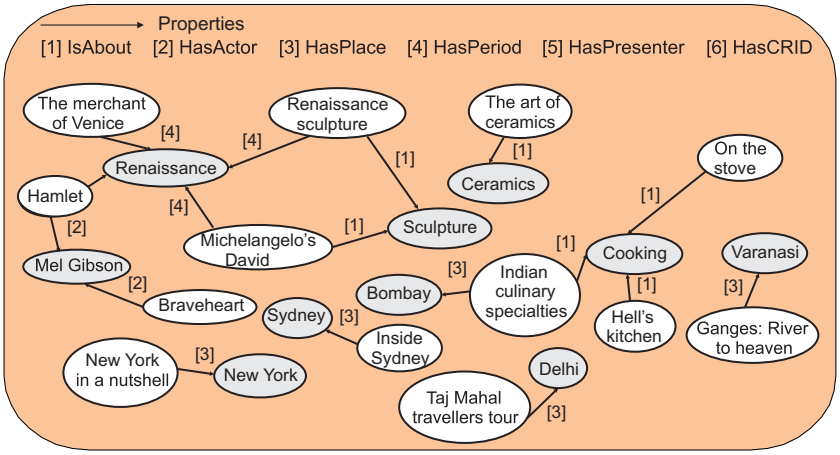

Fig. 4. Ontology of Interest selected for *U* during the pre-filtering phase.

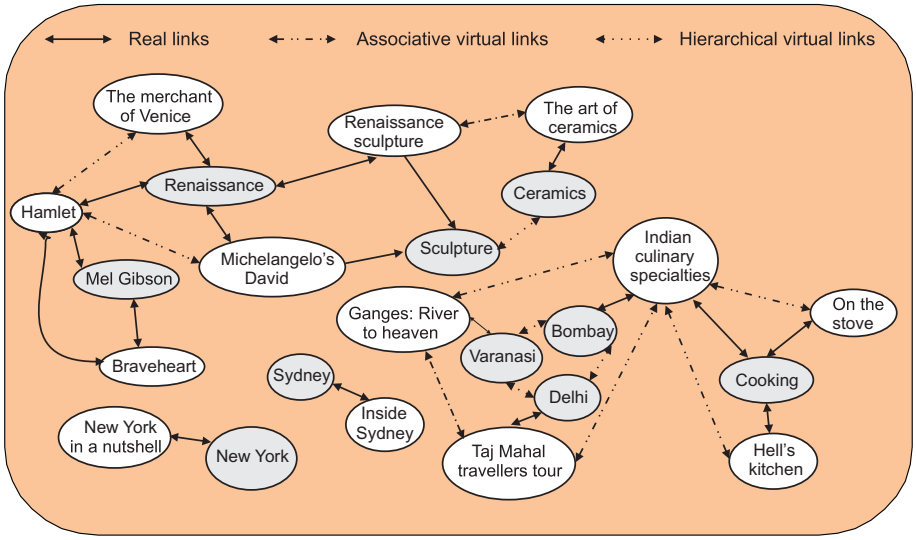

Fig. 5. Nodes and links in the user *U*'s SA network.

5.2. Pre-filtering phase (II): inference of semantic associations

After delimiting the user's Ontology of Interest, our strategy infers semantic associations between the (positive and negative) preferences of *U* and the TV programs represented in his/her Ontology of Interest, as explained in Section 4:

- Firstly, the documentary *The art of ceramics* is related to the program *Renaissance sculpture*, that was appealing to *U*. Specifically, we infer an association between these contents due to the fact that both are about closely-related plastic arts (ceramics and sculpture, respectively). This leads to a ρ-*join* association by the union class *Plastic arts*.
- Secondly, the documentary *Ganges: River to heaven* defined in *U*'s profile is associated with the tourism program *Taj Mahal travelers tour*, because both contents are linked to different cities in India (*Varanasi* and *Delhi*, respectively), leading to a ρ-*join* association through the union class *India cities*.
- Finally, our strategy also discovers semantic associations between the programs included in the user's Ontology of Interest and *U*'s negative preferences. As shown at the bottom of Fig. 2, *Indian culinary specialties* is related to the programs *On the stove* and *Hell's kitchen*, because they are devoted to cooking, which seems to be an unappealing topic to *U* (see Table 1). The property sequence established between *Indian culinary specialties* and *On the stove* by the instance *Cooking* leads to a ρ-*path* association between both programs. Analogously, our approach also infers a ρ-*path* association between the program about Indian cookery and *Hell's kitchen*. However, as *Bombay* is located in the country of interest for *U*, it is also possible to establish a ρ-*join* association between *Indian culinary specialties* and his/her positive preferences (by the union class *India cities*). In this case, SA techniques must explore the two kinds of relationships and decide whether the program about Indian cookery should be suggested to *U* or not. For that purpose, our strategy firstly builds the user network by including both the instances selected by the pre-filtering phase and the discovered semantic associations, as shown in Fig. 5.

5.3. Recommendation phase: reasoning via SA techniques

U's SA network is weighed by assigning the highest values to the links established between nodes that are very relevant for the user (i.e. strongly related to his/her preferences).[4] After activating initially *U*'s preferences and spreading their activation levels though his/her network, our strategy ends up recommending the following programs:

- ***The art of ceramics***. Firstly, we discover the interest of the user in the program *The art of ceramics* due to a high activation level obtained after the spreading process. According to the user's SA network depicted in Fig. 5, this level is computed by the Hopfield Net algorithm by combining two contributions in the sigmoid function: (i) the activation levels of the two nodes reaching *The art of ceramics* (i.e. *Renaissance sculpture* and *Ceramics*) and (ii) the weights of the two links joining both nodes to the program about ceramics.

[4] In order to compute these weights, it would be necessary to consider the full domain ontology, instead of just a brief excerpt from it. For that reason, we do not consider numerical values in the example; instead, we value qualitatively the weights assigned to the links in *U*'s SA network.

– Node *Renaissance sculpture*. In accordance with the weighting process described in Section 4.3, the weight of a link between two nodes depends either on their *DOI* indexes (if these nodes are defined in the user's profile) or on their semantic intensity values (otherwise). Since the node referred to the documentary *Renaissance sculpture* is defined in *U*'s profile, Eqs. (3) and (4) lead to the following weight for the link between the program about sculpture and the content about ceramics (which are related by a ρ-*join* association)

$$w_{ij} = 0.5 \cdot \mu_{i,j} \cdot (DOI_U(\textit{Renaissance sculpture}) + \lambda_{Sem}(\textit{The art of ceramics}))$$

According to *U*'s SA network in Fig. 5, *Renaissance sculpture* and *The art of ceramics* are joined by a virtual path of length 2, so that Eq. (5) leads to $\mu_{ij} = 0.5$, and besides $DOI_U(\textit{Renaissance sculpture}) = 1$. Regarding the semantic intensity of *The art of ceramics*, it also gets a high value because this documentary is closely related to *U*'s preferences, as we commented in previous sections. Gathering these contributions, we compute a high weight for the link coming from the node *Renaissance sculpture*. In order to obtain the activation level for *The art of ceramics*, the Hopfield Net algorithm combines the weight of this link with the level propagated from the node *Renaissance sculpture*. As the program about sculpture is defined in *U*'s profile, its activation level equals its *DOI* index in the preferences depicted in Table 1. These contributions help to increase the activation level of the program about ceramics.

– Node *Ceramics*. The weight of the real link between this node and the program *The art of ceramics* is computed analogously to what we explained before. In this case, the class instance *Ceramics* is not defined among the *U*'s preferences, hence Eqs. (3) and (4) lead to the following weight for this link:

$$w_{ij} = 0.5 \cdot (\lambda_{Sem}(\textit{Ceramics}) + \lambda_{Sem}(\textit{Theartofceramics}))$$

The semantic intensity of the instance *Ceramics* is high in this example because this node is strongly related to the positive preferences of the user *U* (who is greatly interested in the plastic art of sculpture). Regarding the activation level that *Ceramics* propagates to the program about ceramics, note that it gets a high value because the node is joined by a hierarchical virtual link to other node very relevant for *U* (*Sculpture* in Table 1). This hierarchical link tends to increase the level of *Ceramics*, thus helping to make the activation level propagated to the program about ceramics higher. Thanks to this level and the weight of the real link mentioned before, the Hopfield Net algorithm measures a high activation level for the program about ceramics, which is finally recommended to the user.

- ***Taj Mahal Travelers Tour***. Our strategy suggests to *U* the program *Taj Mahal travelers tour* by considering: (i) the highly weighed links coming from the nodes *Ganges: River to heaven* and *Delhi*, and (ii) the high activation levels of both nodes after the spreading process carried out by the Hopfield Net algorithm.
  – Node *Ganges: River to heaven*. On the one hand, this program is defined in *U*'s profile with a maximum *DOI* index (whose value is 1); for that reason, it propagates a high activation level to the content about Taj Mahal. On the other hand, the pre-filtering phase measures a high value of semantic intensity for the program *Taj Mahal travelers tour* due to its relationship with the positive preferences of the user *U* (who is interested in programs about tourist attractions and related to India). The *DOI* index of *Ganges: River to heaven* and the semantic intensity value of the program about Taj Mahal are mixed by Eqs. (4) and (5), leading to a high weight for the associative virtual link established between the two programs. This fact increases greatly the activation level of *Taj Mahal travelers tour*.
  – Node *Delhi*. According to Table 1, *U* enjoyed the region of *Varanasi* along which river Ganges flows. The relevance of this region is propagated to the node *Delhi* thanks to the hierarchical virtual link established between the two class instances in *U*'s SA network. Analogously, thanks to a real link the relevance of *Delhi* for *U* is spread to the content *Taj Mahal travelers tour*, thus increasing its activation level and causing this program to be finally suggested to the user.

At the core of our proposal, our reasoning mechanism based on SA techniques allow to identify items that must not be recommended to the user. The semantic associations establish only bonds among the available items, whereas SA techniques are in charge of looking into those relationships in order to decide whether an item must be suggested to the user when it is related to both his/her positive and negative preferences. Let us clarify this point through an example.

As shown in *U*'s network, *Indian culinary specialties* receives links from the programs *Ganges: River to heaven*, *Hell's kitchen* and *On the stove*. The first program has a very significant *DOI* index which helps to increase the activation level of the program about Indian cookery. On the contrary, the negative *DOI* indexes of the two other programs tend to reduce this level. These values are finally combined with the negative value injected from the node *Cooking*. As a result, a low activation level is obtained for the program *Indian culinary specialties*. In other words, even though this program is bound to a country of interest for *U* (India), our reasoning processes discover that its topic is unappealing to him/her, so *Indian culinary specialties* is not finally suggested.

As a conclusion, note that the diverse nature of our reasoning-based recommendations is due to the fact that the suggested programs do not have the same attributes defined in the user's profile (e.g. *Sculpture* and *Varanasi* bound to the programs *Renaissance sculpture* and *Ganges: River to heaven*, respectively, in Table 1). Rather, they are related to his/her preferences from a semantic point of view. For instance, recall that a user who had liked a program about sculpture has

received recommendations about ceramics (branch of plastic arts related to his/her interests) and about *India*, a country of deep-rooted art tradition.

6. Preliminary evaluation

We have set our first testing experiences in the context of Digital TV, because this is the scope where the R-AVATAR system will be deployed, implementing our recommendation strategy. Our long-term goal is to carry out a rigorous quantitative evaluation – driven by accuracy metrics typically adopted in personalization works – after the deployment of the system, when we will handle a real scenario with about 1000 TV contents every day and 80,000 potential users accessing our recommendations over a long period of time. Unfortunately, the end of the deployment is expected for the second half of 2011, which would cause a significant delay in our research works to continue improving the approach. For that reason, this section focuses only on the tests that we have carried out in a pre-deployment stage. Our evaluation was organized as follows:

- First, the goal was to qualitatively measure the opinions of a set of 150 users about our reasoning-driven content-based recommendations, which is a must before deploying the system toward ensuring successful recommendations. The users involved in the tests were recruited from among our (under) graduate students, their relatives and friends. As a common method of evaluation in the personalization area (see [31,20,5]), we have resorted to questionnaires which lead to a quick feedback from the users about the personalization quality achieved.
- Second, the purpose was to analyze that relevance feedback by statistical techniques that will help us identify which parameters influence on the ratings given by the users to our reasoning-driven content-based recommendations.

We did not consider recommendations offered by existing content-based approaches [41,30] in our tests because they do not furnish any alternative solution for *overspecialized* recommendations that could be compared against our semantic reasoning mechanisms (remember the sample recommendations described in Section 5). Regarding existing semantics-based collaborative (and hybrid) systems [33,13,29,40], their recommendations were not compared against ours because their philosophies are essentially different. Specifically, some recommendations selected by collaborative approaches would go unnoticed to our strategy (because we do not consider the profiles of other users), and vice versa (because existing approaches disregard the relationships discovered by our semantic reasoning techniques). An evaluation to assess quantitatively these approaches against our proposal in terms of scalability and performance will be postponed until the end of R-AVATAR deployment.

6.1. Experimental setup

For our pre-deployment experiments, we have built an ontology about the TV domain containing about 50,000 nodes referred to specific TV programs and their semantic attributes. Besides, we have developed a validation tool whose main functionalities are: (i) initialization and modeling of user profiles, (ii) updating of preferences from the user-provided relevance feedback, and (iii) delivery of personalized recommendations by executing our reasoning-driven content-based strategy.

Specifically, the structure of classes and properties in the TV ontology has been automatically extracted from the TV-Anytime metadata specification [38], which standardizes XML files describing multiple attributes of audiovisual contents (e.g. genres, credits involved in the programs, and target audience), as shown in Fig. 2. To tackle this automatic process, we have used an XSL sheet with several transformation rules to convert XML elements into OWL components. Once the classes and properties had been defined, we populated the OWL knowledge base by including specific TV programs and their semantic attributes. Specifically, the programs were extracted from the databases AMG (*All Movie Guide*) and IMDB (*Internet Movie DataBase*), the BBC website[5] and even RSS websites like *The History Channel*.[6] The population process was semi-automatic: firstly, we automatically retrieved most of the semantic descriptions of specific TV programs from the mentioned databases; next, those annotations were refined manually by adding values for TV-Anytime attributes defined in our ontology.

The knowledge formalized in our ontology was queried by an OWL-specific API (*Application Programming Guide*) provided by ProtTgT,[7] a free open-source tool that includes mechanisms to create, view and query the classes, properties and specific instances formalized in OWL ontologies. Thanks to these mechanisms, we also obtained graphical interfaces to (i) initialize and update the users' profiles, (ii) display the list of TV programs suggested for each viewer, and (iii) develop auxiliary tools to explore the semantic reasoning processes. Specifically, we have developed a tool (named *Reasoning Inspector*) that permits to understand the kind of semantic associations that lead to our diverse content-based recommendations. In order to implement this tool, we have used an ontology-viewing plugin provided by ProtTgT, named TGVizTab,[8] that allows to create and browse generic graphs in a dynamic and interactive way. The nodes in the graph are the classes (and their instances) in the domain ontology, whereas the links identify the properties and relationships existing among the nodes. For clear and intuitive exploration, the plugin includes configuration options in order to: (i) control the knowledge represented in the graph,

[5] http://backstage.bbc.co.uk/data/7DayListingData.
[6] http://www.history.com/.
[7] http://protege.stanford.edu/
[8] See http://users.ecs.soton.ac.uk/ha/TGVizTab/ for details.

(ii) show only some properties of a class or specific instance, or (iii) depict subgraphs where it is possible to show or hide the links that relate some nodes to others.

Thanks to our Reasoning Inspector, we can display a graph that represents only the user's Ontology of Interest, including his/her preferences and the semantic associations inferred among them and the TV programs to be suggested. We can even interact with the depicted graph by selecting some nodes and showing both the properties and hierarchical relationships formalized in the graph, and the semantic associations inferred from them. Since this graph identifies the user's SA network, our Reasoning Inspector allows also to control the spreading process carried out by the Hopfield Net algorithm, thus enabling the exploration of the nodes in the user's SA network, the real and virtual links established among the nodes, and the activation levels resulting from the propagation process, as shown in the snapshot of Fig. 6.

After implementing our validation tools, the experimental design was organized as follows:

- Initially, the 150 users involved in our tests logged in a web page by their e-mails and filled in a form to initialize their personal profiles. This form included a list of 200 TV programs to be rated by the users on a scale from 0 to 9 (with 0 representing the greatest disliking and 9 the greatest liking). The TV programs were classified into a hierarchy of genres (extracted from the TV ontology) including CATEGORIES/subcategories (e.g. NEWS: national, economy; SPORTS: athletics, cycling; MUSIC: jazz, rock, etc.). Besides, each TV program was shown with a brief textual description, in such a way that the users could even rate contents they did not know.
- Then, the information about the users' preferences was processed by our validation tool, that was in charge of (i) modeling the users' profiles as described in Section 3.2, and (ii) executing our reasoning-driven recommendation strategy.
- Next, we processed the output of our strategy in order to promote our reasoning-based recommendations to the detriment of traditional content-based ones driven by syntactic metrics. This is because our strategy is able to select both contents associated to the user's preferences and programs that share the same attributes defined in his/her profile.

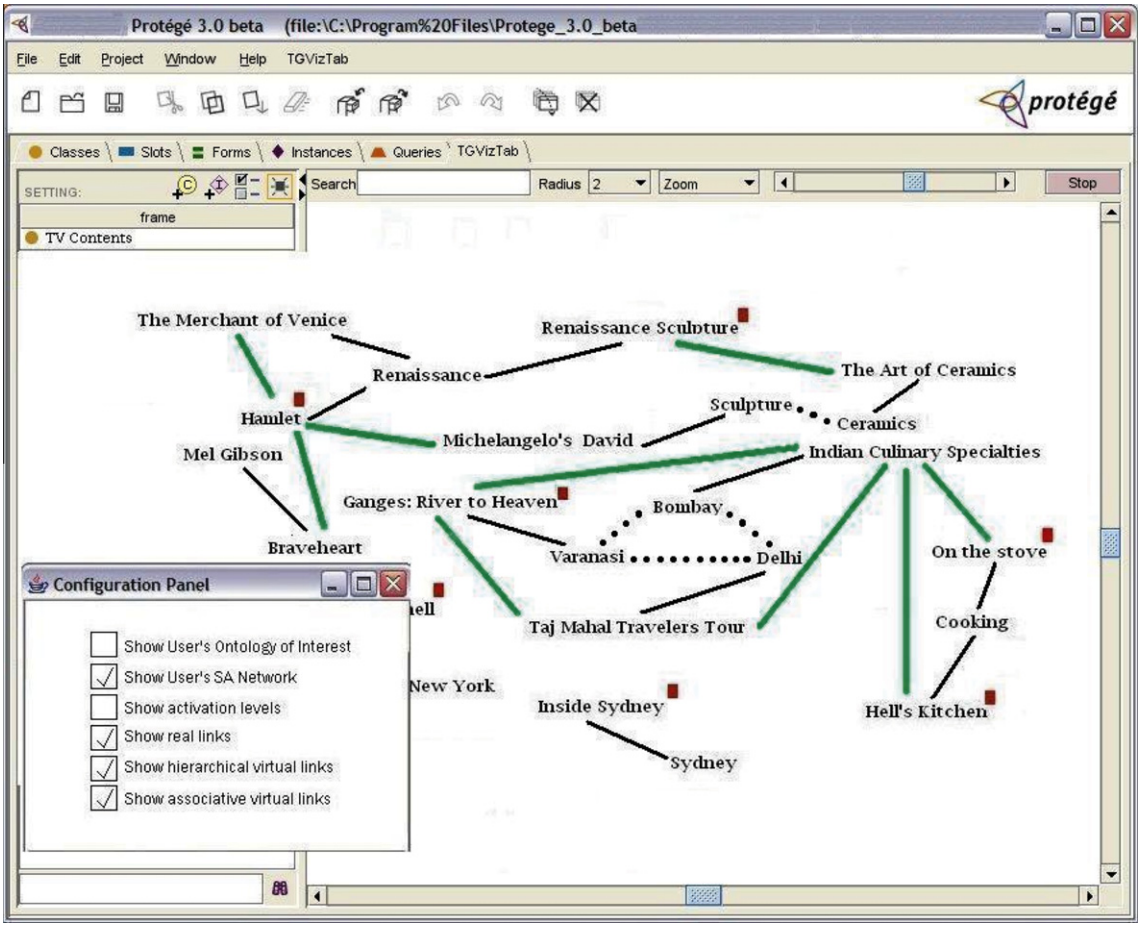

Fig. 6. Snapshot of our testing tool based on TGVizTab plugin: the user's preferences are marked with red squares in his/her SA network; the real links are represented by black thick lines, associative virtual links are identified by green lines appearing thicker, and hierarchical virtual links are marked with dotted lines. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- In order to help the users understand our recommendations, we created a brief synopsis for each suggested TV program with the aid of the Reasoning Inspector. Specifically, the synopsis of a TV program explained the existing associations between that content and the programs the users had rated highly when initialized his/her profile.
- Finally, we daily e-mailed to each user a recommendation with 8 TV programs (and their synopses) and requested them to provide us with relevance feedback by rating each suggested content. These ratings were employed to update the user profiles and select new recommendations day after day over the 7-day testing period.

6.2. Preliminary experimental results

In order to assess the quality of our recommendations, we measured daily their *precision* for each user, defined as the percentage of suggested TV contents that were interesting for him/her (i.e. the ones that got a rating greater than 6 in the relevance feedback). Finally, we averaged the daily precision values over the 7-day testing period, and sent the users an e-mail with a questionnaire (where we reminded the list of 56 TV programs recommended to the user during the experiments) including questions such as: "*do you think that your recommendations were diverse or very repetitive, and why?*", "*did you know the TV programs we suggested?*", "*would you be willing to pay for receiving our recommendations?*", "*how do you assess globally our personalization capabilities?*", "*how many hours do you spend each week watching TV?*", just to name a few. From the users' answers, we could evaluate the global utility of our recommendation strategy and draw interesting observations:

- The precision values ranged between 79% and 51% approximately: most of the users (82%) obtained more than 70% of precision, while just 8% of them achieved less than 60%. These values did increase over time thanks to the feedback provided by the users after the recommendations, which helped our reasoning-driven strategy to know their preferences better.
- Most of the users involved in the tests (78%) evaluated our personalization capabilities *positively* or *very positively*, while the remaining ones either remained indifferent (18%) or did not find the reasoning-driven recommendations appealing (4%).
- Nearly all the users noticed the diverse nature of the recommendations received during the last days of testing period. In fact, most of these users (about 76%) told us that they did not know some of the suggested TV programs; however, they admitted that the way to relate the programs to their personal preferences was really "*ingenious*", "*peculiar but appropriate*" and even "*intelligent*".
- From the questionnaires, we also discovered that most of the users (84%) would be willing to pay a (small) added fee for receiving our recommendations, which undoubtedly evidences the interest of our content-based approach.

In spite of the preliminary nature of our experimental tests, this evaluation has permitted us to check good results of recommendation precision, as well as the users' acceptance about our reasoning-driven recommendations. The next step was to use statistical techniques in order to analyze up to which extent the users' ratings in the recommended programs were influenced by parameters that we knew from the initial questionnaires and from the relevance feedback. In order to consider these dependences, we relied on: (i) a multiple linear regression model, which allows to predict a user's score on one variable (named the *criterion variable*) on the basis on his/her score on several other variables (named the *predictor variables*) and (ii) the package SPSS,[9] a commonly adopted software for statistics-driven analysis.

6.3. Statistical analysis

The statistical analysis carried out in our evaluation considers variables such as diversity of the recommendations, ratings in genres related to the suggested programs, viewing habits and so on, which might all contribute towards the user satisfaction with our suggestions. If we handle data on all of these variables, we can see how many and which of them gave rise to the most accurate prediction of user satisfaction. In our statistics-driven evaluation, we used as criterion variable the user ratings in the recommended programs (hereafter rating_program), and as *possible* significant predictor variables the following ones:

- Users' ratings in the genres of the recommended programs (hereafter ratings_genres). This is a continuous variable with values between 0 and 9. These values were inferred from the ratings the users assigned to 200 TV programs when initializing their profiles.[10]
- Positive perception about the recommendations (perception_recom). This variable takes as values 1 (when users rated as *positive* or *very positive* the recommendations including the suggested program) or 0 (otherwise).
- Diversity of the recommendations (diversity_recom). The values are 1 (if the user has rated as *diverse* the recommendation in which the suggested program was included) or 0 (in case of *specialized* recommendation).

[9] Statistical Package for the Social Sciences (http://www.spss.com).

[10] We assume that the rating of a program is inherited by the genre(s) which it belongs to. If the program is categorized under several genres, we average all their ratings to set the value of the predictor variable rating_genres.

- Information about recommended programs (info_program). This variable takes as values 1 (if the user said in the questionnaire that he/she knew the program prior to our recommendation) or 0 (otherwise).
- Number of hours per week that the user spends watching TV (TV_hours). It is a continuous variable whose values fall in the range $[0, 24 * 7]$.

The interpretation of the regression model driven by the predictor variables above is as follows: a user's rating in a recommended program depends on: (i) his/her level of interest in the genres under which the program is classified in the TV ontology, (ii) his/her opinion about the recommendation including this program, (iii) the diverse or specialized nature of such recommendation, (iv) the fact that the user might know the suggested program before our suggestion, and (v) the amount of time the user watches TV per week.

According to the guidelines included in Appendix A, using linear regression requires a large number of observations and non-collinear predictor variables, to draw valid statistical inferences from the resulting model. Regarding the first requirement, we handled 150 observations (justs as many as users took part in our experiments). In order to study the linear dependencies between the predictor variables and the users' ratings in our recommendations, we randomly chose one of the programs our strategy suggested over 7-day testing period to each one of the 150 involved users. Finally, to corroborate non-multicollinearity among the 5 predictor variables, we examined their correlation matrix with the aid of SPSS. We did find a multicollinearity problem in the series: specifically, the high correlation value between perception_recom and diversity_recom (0.918) revealed a narrow relationship between the user's global perception about our recommendations and their diversity. Analogously, the correlation value 0.930 between info_program and TV_hours means that the fact that the user knows beforehand a recommended program is strongly related to the time he/she spends watching TV.

In order to remove the multicollinearity problem, we resorted to factor analysis (see Appendix C) to find the latent factors that account for the relationships existing among multiple metric variables. In Section 6.3.1, we describe the factors extracted from the variables that correlate highly with each other. Only the most significant variable of each factor (whose selection is explained in Section 6.3.2) was used as a predictor in the multiple regression model presented in Section 6.3.3.

6.3.1. Factor analysis

As per the guidelines explained in Appendix C, our factor analysis was organized as follows:

- First, we determined the factorability of the *correlation matrix* through Kaiser–Meyer–Olkin measure of sampling adequacy. The resulting value (KMO = 0.86) confirms that the data are suitable for factor analysis because the factors extracted will account for a "meritorious" amount of the variance of the predictor variables (see Table C.1).
- Second, we extracted factors from the correlation matrix by exploiting the *principal components* method provided by SPSS. According to Table 2 and Kaiser's criterion, three components or factors were extracted for having eigenvalues greater than 1.0.
- In order to interpret/understand what the extracted factors measure, we need to identify which variables load (correlate) highest on each factor. To this aim, it is necessary to analyze the factor loadings included in the component matrix computed by SPSS. With the goal of clarifying the factor pattern, we have resorted to the *varimax rotation* method (see Appendix C), obtaining the *rotated components matrix* shown in Table 3.
- As per the rotated component matrix, perception_recom and diversity_recom load on the first component or factor (hereafter F1); info_program and TV_hours are highly correlated on the second component (F2), and finally rating_genres load on the third component (F3). According to the meaning of our predictor variables, we named the factors as follows: F1 was named *nature of the recommendations* because it brings together the users' global perception and the type of recommendation. Factor F2 has been named *experience of the viewer* because it summarizes the programs known by the user and the time he/she spends with TV. Regarding F3, it is difficult to understand this factor since only one variable loaded high on it. In absence of more variables, we named it *interest in TV content genres*.

This three-factor solution seems reliable due to two main reasons. On the one hand, the extracted factors account for a high percentage of the variance in the criterion variable (88.2% from Table 2). On the other, even though F3 is ambiguous (because it only involves one predictor variable), the factor pattern is very clear for the first two components.

Table 2
Eigenvalues and total variance explained by the components extracted by principal components method.

Component	Initial eigenvalues			Extraction sum of squared loadings		
	Total	% of variance	Cumulative %	Total	% of variance	Cumulative %
1	2.02	40.4	40.4	2.02	40.4	40.4
2	1.28	25.6	66	1.28	25.6	66
3	1.11	22.2	**88.2**	1.11	22.2	**88.2**
4	0.56	11.2	99.4			
5	0.03	0.6	100			

Table 3
Rotated component matrix.

Predictor variables	Component		
	1	2	3
Rating_genres	0.134	0.1722	0.852
Perception_recom	0.954	0.131	0.268
Diversity_recom	0.938	0.081	0.174
Info_program	0.2272	0.751	0.362
TV_hours	0.380	0.814	0.374

Table 4
Model summary: model 1 (Predictors: diversity_recom, info_program, rating_genres); model 2 (Predictors: diversity_recom, TV_hours, rating_genres); model 3 (Predictors: perception_recom, info_program, rating_genres); model 4 (Predictors: perception_recom, TV_hours, rating_genres).

Model	R	R^2	Adjusted R^2
1	0.9349	**0.874**	0.8714
2	0.9123	0.832	0.829
3	0.889	0.7903	0.786
4	0.862	0.743	0.738

After identifying the factors hidden behind the five initial variables, we selected the most significant one of each component and included it in our multicollinearity-free multiple regression model.

6.3.2. Selecting the most significant variable for each factor

As the most significant variable of each factor, we have chosen the one with the highest correlation to the criterion variable of our regression model. To make this selection, we computed via SPSS the correlation matrix including the five initial predictor variables along with rating_program. As per the resulting correlation matrix, the most significant variables for F1 and F2 and diversity_recom and info_program, respectively.[11]

This selection has been validated by computing the linear regression models for the possible combinations of the predictor variables that load on F1 and F2, and focusing on their respective squared multiple correlation coefficients (R^2). As per the results depicted in Table 4, the best regression model corresponds with the series including as predictor variables diversity_recom, info_program and rating_genres. This model accounts for 87.4% of the variance in the criterion variable. This means that if we select randomly a user about whom we know nothing, there is uncertainty (variance) about which will be his/her value for rating_program; however, if we know additional information about the predictor variables, our regression model allows us to predict the user's rating with 87.4% less uncertainty compared to the previous scenario.

6.3.3. A multiple linear regression model

Having identified the predictor variables of our regression model, we computed their coefficients to measure how much they influence the criterion variable. With the aid of SPSS, we obtained the values of Table 5 and assessed the overall significance of the resulting regression model by analyzing its variance by ANOVA tests (Table 6).

- In order to analyze the impact of each predictor variable on the rating_program, we focused on the standardized β coefficients in Table 5. The large value of the β coefficient of diversity_recom suggests that this predictor variable is having a large impact on the user's ratings in our recommendations. In other words, the level of diversity of the recommendations has a huge effect on the user's ratings in the programs included in them. According to the remaining β coefficients, the level of interest of the users in the genres of the recommended programs (rating_genres) and the fact of knowing information beforehand about these contents (info_program) affect their ratings (rating_program) to a much lesser extent.
- As explained in Appendix B, the null hypothesis in ANOVA for multiple linear regression states that all of the coefficients weighing the predictor variables are 0. According to Table 6, the value for the F test statistic is less than 0.001, providing strong evidence against the null hypothesis and confirming that the predictor variables are linearly related to the criterion variable.

After fitting the regression line, we harnessed the graphical features provided by SPSS to corroborate the hypotheses of normality, homocedasticity and independence by analyzing the residuals of the model (these are necessary conditions as explained in Section A.1). First, we used a Q–Q plot [21] to compare the standardized residuals with a standard normal population, where we noticed the linearity of the points round the principal diagonal. This suggests that the residuals do not

[11] The correlation values of diversity_recom, perception_recom, info_program and TV_hours with rating_program were 0.821, 0.692, 0.448 and 0.339, respectively.

Table 5
Coefficients computed for our multiple linear regression model.

Model	Unstandardized coefficients		Standardized coefficients
	B	Std. error	β
Constant	1.936	0.249	
Diversity_recom	5.882	1.157	0.81
Info_program	0.21	0.146	0.14
Rating_genres	0.108	0.151	0.05

Table 6
ANOVA table over regression: criterion variable is rating_program and predictor variables are diversity_recom, info_program and rating_genres.

Model	Degrees of freedom	Sum of squares	Mean squares	*F*	Sig.
Regression	3	2325.117	775.04	337.58	<0.001
Error	146	335.2	2.296		
Total	149	2660.317			

seem to deviate from a random sample from a normal distribution in any systematic manner. Second, we corroborated the independence hypothesis via a graph including observations *versus* residuals, where we found that the cloud of points was focused on a strip parallel to *x*-axis and 0-centered.

Lastly, to corroborate the relevance of diversity_recom in the users' ratings in our recommendations, we computed a new multiple linear regression model by omitting the diversity of recommendations as predictor variable, as shown in Tables 7 and 8. According to both tables, the mean square error term is smaller with diversity_recom included, indicating less deviation between the observed and fitted values. In ANOVA table, the value for the *F*-test is less than 0.001, providing strong evidence against the null hypothesis. Regarding the squared multiple correlation coefficient R^2, it has decreased up to 0.354, which means that the model without diversity_recom is just able to account for the 35.4% of the variance in rating_program. This is a significant worsening over the multiple regression model computed with diversity_recom as a predictor variable, which accounted for 87.4% of the variance of the criterion variable.

6.3.4. Discussion from statistical results

To sum up, we present some thoughts drawn from the statistical results:

- First, from β coefficients in Table 5, it follows that when the user rates a recommended program, his/her interest in the genres of this program has much lower influence than the diversity of the recommendations. In other words, to suggest programs belonging to the genres defined in the user's profile does not guarantee a successful recommendation (maybe because the user can get bored of programs too similar to his/her preferences). This confirms the need to endow content-based recommendations with diversity.
- Second, the low value of the β coefficient for info_program reveals that the fact that the user knows information about the recommended program does not assure that it will be appealing to him/her. In our tests, users relied on the synopses of the recommended programs where we explained the semantic associations existing between their interests and our reasoning-driven recommendations, so that they could even rate unknown contents.
- In conclusion, the diversity of the recommendations is a predominant factor in the high ratings given by the users to the suggested programs, as inferred from our observations, where the users' lowest ratings were associated with non-diverse recommendations. The lack of diversity noticed by the users in their relevance feedback was referred to

Table 7
Model summary with info_program and rating_genres as predictor variables (diversity_recom is omitted).

Model	*R*	R^2	Adjusted R^2
1	0.595	**0.354**	0.3452

Table 8
ANOVA table over regression: criterion variable is rating_program and predictor variables are info_program and rating_genres.

Model	Degrees of freedom	Sum of squares	Mean squares	*F*	Sig.
Regression	2	452.02	226.01	40.27	<0.001
Error	147	825	5.612		
Total	149	1277.02			

recommendations made during the first days of the testing period, when our strategy knew little information about their personal preferences. In our approach, the diversity is narrowly bound to the inference of semantic associations and their processing by SA techniques, which justifies our reasoning-driven content-based strategy.

6.4. Some thoughts about scalability and computational feasibility

To finish the description of our experimental evaluation, here we describe some features aimed at ensuring scalability and computational viability, two critical parameters for the deployment of a recommender system implementing our recommendation strategy.

- As explained in Section 4.4, the Hopfield Net algorithm predicts the level of interest of the user in each TV content included in his/her SA network, in such a way that if this level is significant enough the program is finally recommended to the user. Due to the iterative nature of the Hopfield Net algorithm, our system can return *suboptimal solutions* to ensure fast responses to the users practically in real-time.
- In running the Hopfield Net algorithm, we do not compute the values of the sigmoid function (see Eq. (1)), but rather look them up in a pre-computed table with a precision of 10 decimal numbers for the argument. In the meantime before we can quantify the computational cost with the users of R-AVATAR, we have carried out some in-lab tests considering the number of users and TV contents this system will handle. As per our results, the final values of Hopfield Net (i.e. optimal solutions) for 80,000 users and 1000 TV contents per day could be computed in about 22 h, using a dual-core server with 3 GHz processors and 8 GB RAM memory.
- Our implementation works with a *master server* that shares out the computational burden among several *slaves personalization servers*. Specifically, the Digital TV receiver of each viewer requests the master server to assign a computationally available slave server, which returns recommendations by running our content-based strategy and accessing the database that lodges the TV ontology and the user profiles.
- The third feature consists of identifying tasks involved in our content-based strategy that can be carried out simultaneously, to distribute them among several servers. On the one hand, the *ontology server* updates the contents formalized in the ontology (starting from the TV schedule of the cable operator), and computes off-line many parameters that can be reused as new users log into the recommender system (e.g. distances between nodes in the property sequences, and common ancestors between each pair of nodes in the domain ontology). In the meanwhile, the *profiles server* takes charge of updating the users' profiles by adding new preferences and ratings. Lastly, the *slave personalization server* (assigned to each viewer by the master server) executes the pre-filtering and recommendation phases of our strategy by considering the user's preferences and information required for reasoning purposes (sent by profiles server and ontology server, respectively).
- Finally, we can maintain multiple instances of the profiles server and the ontology server. In order to avoid bottlenecks when accessing the TV ontology and users' profiles, each instance of these servers works with a replica of the system database. This is possible in our content-based approach because the users are independent, that is, the preferences of one user do not influence the recommendations made to others. Obviously, a collaborative approach would not be able to harness this optimization.

7. Conclusions and further work

In this paper, we have fought the *overspecialized* nature of traditional content-based recommender systems, which only suggest items very similar to those the user already knows (mainly due to the adoption of syntactic matching techniques). The novelty is that our content-based approach overcomes this limitation without considering the preferences of other individuals, which was the only solution proposed so far in literature at expenses of introducing other severe drawbacks.

Instead of resorting to a collaborative approach, our strategy diversifies the recommendations by exploiting semantic reasoning techniques about the user's preferences, which brings important closely-related benefits. On the one hand, our content-based approach is less demanding in computational terms than collaborative solutions, which require matching techniques to compare the profiles of many users before offering recommendations to an individual. Besides, the non-reliance of other users' profiles frees our strategy from privacy concerns related to the confidentiality of their personal preferences, and it permit also to offer recommendations to a user at any time. In contrast to this, collaborative recommenders are strongly limited for the matching techniques adopted to form each user's neighborhood: if like-minded users cannot be found, then collaborative recommendations cannot be offered to a given user. In this regard, note also that our approach does not imply latencies in the presentation of the recommendations. This is not true for a collaborative system where the preferences of many users must be known before elaborating recommendations for a given user.

Our recommendation strategy harnesses the benefits of semantic reasoning over an underlying ontology as a means to discover additional knowledge about the user preferences, enabling to compare them to the available items in a more effective way. This way, instead of suggesting items very similar to those the user liked in the past, our strategy recommends items *semantically related* to his/her preferences. For that purpose, we have extended existing semantic reasoning mechanisms, so that they can be adopted in a personalization scenario where the focus is put on the user's preferences. Specifically, we have described how semantic associations and SA techniques fit together in our content-based recommendation

strategy: the associations help to diversify the recommendations because they discover hidden (semantic) relationships between the user's preferences and the available items, while our improved SA techniques enable to (i) process efficiently the knowledge inferred by those associations, and (ii) evolve the recommendations as the user' preferences change over time.

Our contribution is flexible enough to be reused in multiple contexts, becoming an easy-to-adopt starting point to implement diverse personalization services. Specifically, the strategy has been incorporated into a TV recommender system named R-AVATAR that will be deployed over the cable networks of a Spanish operator. The goal of this system is to identify potentially appealing programs to each subscriber among the contents available in the digital stream. Since the deployment is expected for the second half of 2011, we have carried out some in-lab tests aimed at assessing the personalization quality achieved and the perception of 150 users about our reasoning-driven recommendations. Even though these results are preliminary, we think that they are a good indication of (i) the capability of our approach to adapt the recommendations as the reasoning techniques learn new knowledge about the users' preferences, and (ii) the users' satisfaction regarding the accuracy and diversity of our content-based suggestions. In a second phase of our evaluation, we conducted statistical tests (based on multiple linear regression, factor analysis and ANOVA tests), which revealed that the diversity of our reasoning-driven content-based recommendations very positively affects on the users' ratings.

As future work, we plan to carry out a quantitative evaluation involving the 80,000 potential users of R-AVATAR. Besides assessing our reasoning-driven personalization capabilities, we will exploit the data gathered in order to compare our approach against existing collaborative and hybrid works in terms of performance and personalization quality.

Appendix A. Multiple Linear Regression

Multiple regression is a statistical technique that allows to identify a set of predictor variables which together provide a useful estimation of a user's likely score on a criterion variable. This technique attempts to model the relationship between two or more predictor variables and the criterion variable by fitting a linear equation to observed data.

Every value of the predictor variable x is associated with a value of the criterion variable y. The population regression line for p predictor variables $x_1,\ldots,x_p$ is defined to be $\mu_y = \beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_p \cdot x_p$. This line describes how the mean response μ_y changes with the predictor variables. The observed values for y vary about their means μ_y and are assumed to have the same standard deviation σ. The fitted values $b_0,\ldots,b_p$ estimate the parameters $\beta_0,\ldots,\beta_p$ of the population regression line.

Since the observed values for y vary about their means μ_y, the multiple regression model includes a term for this variation. This way, the model can be expressed as *DATA* = *FIT* + *RESIDUAL*, where:

- *FIT* refers to the expression $\beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_p \cdot x_p$, and
- *RESIDUAL* represents the deviations of the observed values y from their means μ_y, which is typically denoted as ϵ.

Formally, given n observations, the model for multiple linear regression is expressed as follows:

$$Y_i = \beta_0 + \beta_1 \cdot x_{i1} + \cdots + \beta_p \cdot x_{ip} + \epsilon_i \quad \text{for } i = 1, 2, \ldots, n$$

The values fit by the equation $b_0 + b_1 \cdot x_1 + \cdots + b_p \cdot x_p$ are denoted as $\hat{y}_i$, while the residuals e_i are equal to $y_i - \hat{y}_i$ and represent the difference between the observed and fitted values.

In the following sections, we describe the requirements to use multiple linear regression and the terminology adopted in this statistical technique.

A.1. Requirements

(1) *Linearity*. Multiple regression can be used when exploring linear relationships between the predictor and criterion variables.
(2) *Independence*. The residuals must not follow a systematic pattern with regard to the sequence of observed data.
(3) *Homocedasticity*. The errors ϵ of the regression model must have constant variance.
(4) *Normality*. The errors of the regression model must be normally distributed.
(5) The criterion variable should be measured on a *continuous scale*. A nominal predictor variable is valid but only if it is *dichotomous*, that is, if there are no more than two possible values. Dummy variables can be used to describe variables that take more than two values.
(6) Multiple regression requires a *large number of observations*, which must substantially exceed the number of predictor variables used in the multiple regression model (40:1 is a typically accepted ratio [2]).
(7) When choosing a predictor variable, it is necessary to select one that might be correlated with the criterion variable, but that is not strongly correlated with the other predictor variables. The term *multicollinearity* (or simply collinearity) is used to describe the situation when a high correlation is detected between two or more predictor variables. Such high correlations cause problems when trying to draw inferences about the relative contribution of each predictor variable to the success of the model.

A.2. Terminology

Here we present certain terms that must be explained to understand the results of a linear regression model.

- *Multiple correlation coefficient* (R). This value measures the correlation between the observed value and the predicted value of the criterion variable.
- *Squared multiple correlation coefficient* (R^2). It indicates the proportion of the variance in the criterion variable which is accounted for by the regression model. In essence, it is a measure of how good a prediction of the criterion variable can be made by knowing the values of the predictor variables. However, R^2 tends to over-estimate the success of the model when applied to the real world, so an adjusted R^2 value is calculated.
- *Adjusted squared multiple correlation coefficient* (adjusted R^2). This value takes into account the number of variables in the model and the number of observations considered in it. For that reason, adjusted R^2 is the most useful indicator of the success of the resulting regression model.
- *Standardized regression coefficients* (β coefficients). The β values are measures of how strongly each predictor variable influences the criterion variable. Thus, the greater the β coefficient the greater the impact of the predictor variable on the criterion variable. These coefficients are computed by statistical packages, which typically provides tests to assess the significance of the model. A basis for these tests of significance is analysis of variance (ANOVA).

Appendix B. ANOVA over regression

ANOVA provides information about levels of variability within a regression model. Since the regression line is expressed as *DATA* = *FIT* + *RESIDUAL*, it can be rewritten as follows:

$$(y_i - \bar{y}) = (\hat{y}_i - \bar{y}) + (y_i - \hat{y}_i)$$

The first term is the total variation in the criterion variable y, the second term is the variation in mean criterion variable, and the third term is the residual value. Squaring each of these terms and adding over all of the n observations gives the equation:

$$\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 + \sum (y_i - \hat{y}_i)^2$$

This equation can be rewritten as *SST* = *SSM* + *SSE*, where *SS* denotes Sum of Squares and *T*, *M* and *E* are notation for Total, Model and Error, respectively. The square of the sample correlation is equal to the ratio of the model sum of squares (*SSM*) to the total sum of squares (*SST*). This formalizes the interpretation of R^2 as explaining the fraction of variability in the data accounted for by the regression model.

All the computations involved in an analysis of variance model are shown in Table B.1, where p is the number of predictor variables; n is the number of observations; *DFM*, *DFE* and *DFT* are the degrees of freedom for model, error and total, respectively; *MSM* and *MSE* are the sum of squares for model and error, respectively; and the F column provides a statistic for testing the null hypothesis $\beta_1 = \beta_2 = \cdots = \beta_p = 0$. The alternative hypothesis simply states that at least one of the coefficients $\beta_j \neq 0$ with $j = \{1, \ldots, p\}$. Large values of the test statistic provide evidence against the null hypothesis, confirming that criterion and predictor variables are linearly related.

Appendix C. Factor analysis

Factor analysis is a very popular statistical technique whose purpose is to reduce multiple variables to a lesser number of underlying factors that are being measured by those variables. Mathematically, a factor is a linear combination of variables and factor analysis groups variables according to their correlation.

The steps to carry out factor analysis are the following ones:

(1) *Computation of the correlation matrix and evaluation of its factorability*. The goal is to determine how much common variance exists among the variables and to decide whether the correlation matrix is appropriate for factor analysis. One of the most commonly adopted methods is Kaiser–Meyer–Olkin measure of sampling adequacy (KMO). As detailed in [28], this method computes a value that indicates the degree of common variance among the variables that

Table B.1
ANOVA table for multiple linear regression.

Source	Degrees of freedom	Sum of squares	Mean square	F
Model	$DFM = p$	$SSM = \sum (\hat{y}_i - \bar{y})^2$	$MSM = \frac{SSM}{DFM}$	$\frac{MSM}{MSE}$
Error	$DFE = n - p - 1$	$SSE = \sum (y_i - \hat{y}_i)^2$	$MSE = \frac{SSE}{DFE}$	
Total	$DFT = n - 1$	$SST = \sum (y_i - \bar{y})^2$		

Table C.1
Kaiser–Meyer–Olkin measure of sampling adequacy (KMO).

KMO value	Degree of common variance
0.9 to 1	Marvelous
0.8 to 0.89	Meritorious
0.7 to 0.79	Middling
0.6 to 0.69	Mediocre
0.5 to 0.59	Miserable
0 to 0.49	Do not factor

can be explained by the factors extracted by factor analysis. As shown in Table C.1, the greater the KMO value, the more suitable the correlation matrix for factor analysis.

(2) *Extraction of an initial set of factors*. One of the most used techniques to extract factors from the correlation matrix is the principal components method. Other techniques are *maximum likelihood* method, *principal axis* method, *unweighted least-squares* method, *generalized least squares* method, *alpha* method and *image factoring*, whose internals are described in [28]. These methods allow to compute the so-called *factor loading matrix*, whose components (or factor loadings) measure the correlations between the factors (columns) and their underlying variables (rows). The square of the factor loadings represents the variation in the variable explained by the factor. The sum of the squares of the factor loadings in each column is an *eigenvalue*, which represents the amount of variance in the original variables that is associated with that factor. Analogously, the *communality* is the proportion of variability in each variable accounted for by the extracted factors.

(3) *Selection of the appropriate number of factors to be extracted in the final solution*. A rule for deciding on the number of factors is that each included factor must explain at least as much variance as does an average variable. In other words, only factors for which the eigenvalue is greater than 1 are used, which is known as Kaiser's criterion. Other criteria for determining the number of factors are the Scree plot criteria and the percentage of variance criteria (see [34] for details).

(4) *Rotation of the factors* to clarify the factor patterns in order to better interpret the nature of the factors, if necessary. Sometimes one or more variables may load high on more than one factor, making the interpretation of the factors ambiguous. To facilitate interpretation, it is possible to rotate the axis, which is equivalent to forming linear combinations of the factors. A commonly used rotation strategy is the varimax rotation, which attempts to force the column entries to be either close to 0 or 1 (see [34]).

(5) *Computation of the scores of each observation on each factor*, once they have been identified and named. This step is useful to carry out multiple regression models where the factors are used as predictor variables. Recall that the factors are non-collinear, so that the resulting regression model does not suffer from multicollinearity problems (recall the requirement (7) in Section A.1).

References

[1] G. Adomavicius, A. Tuzhilin, Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 739–749.
[2] P.D. Allison, Multiple Regression: Undergraduate Research Methods & Statistics in the Social Sciences, Pine Forge Press, 1998.
[3] G. Antoniou, E. Franconi, F. van Harmelen, Introduction to Semantic Web ontology languages, Reasoning Web, LNCS 3564 (1) (2005) 1–21.
[4] K. Anyanwu, A. Sheth, ρ-Queries: enabling querying for semantic associations on the Semantic Web, in: 12th International World Wide Web Conference (WWW-03), 2003, pp. 115–125.
[5] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Difino, B. Negro, User modeling and recommendation techniques for personalized electronic program guides. in: L. Ardissono, A. Kobsa, M. Maybury (Eds.), Personalized Digital TV, 2004, pp. 3–26.
[6] M. Balabanovic, Y. Shoham, Combining content-based and collaborative recommendation, Communications of the ACM 40 (3) (1997) 1–9.
[7] D. Beckett, RDF Syntax Specification, 2004. <http://www.w3.org/TR/rdf-syntax-grammar>.
[8] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American, 2001.
[9] Y. Blanco-Fernández, J.J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, M. López-Nores, A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems, Knowledge-Based Systems 21 (4) (2008) 305–320.
[10] D. Brickley, R. Guha, RDF vocabulary description language 1.0: RDF Schema, 2004. <http://www.w3.org/TR/rdf-schema>.
[11] F. Carmagnola, F. Cena, User identification for cross-system personalisation, Information Sciences 179 (1–2) (2009) 16–32.
[12] H. Chen, K.J. Lynch, K. Basu, T. Ng, Generating, integrating, and activating thesauri for concept-based document retrieval, IEEE Experts (special series on Artificial Intelligence in Text-based Information Systems) 8 (1) (1993) 25–34.
[13] C. Cornelis, J. Lu, X. Guo, G. Zhang, One-and-only item recommendation with fuzzy logic techniques, Information Sciences 177 (1) (2007) 4906–4921.
[14] F. Crestani, Application of spreading activation techniques in information retrieval, Artificial Intelligence Review 11 (6) (1997) 453–482.
[15] DAML: The DARPA Agent Markup Language, 2000. <http://www.daml.org>.
[16] R. Diestel, Graph Theory, Springer-Verlag, 2000.
[17] D. Elenius, D. Martin, R. Ford, G. Denker, Reasoning about resources and hierarchical tasks using OWL and SWRL, in: Proceedings of the 8th International Semantic Web Conference, 2009, pp. 795–810.
[18] D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, P. Patel-Schneider, SOIL: An ontology infrastructure for the Semantic Web, IEEE Intelligent Systems 16 (2) (2001) 38–45.
[19] J. Ge, Y. Qui, Z. Chen, Cooperative recommendation based on ontology construction, in: Proceedings of the 2008 International Conference on Computer Science and Software Engineering, 2008, pp. 301–314.
[20] C. Gena, Designing TV viewer stereotypes for an electronic program guide, in: Proceedings of the 8th International Conference on User Modeling, 2001, pp. 274–286.

[21] R. Gnanadesikan, M.B. Wilk, Probability plotting methods for the analysis of data, Biometrika 55 (1) (1968) 1–17.
[22] N. Good, J.B. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, J. Riedl, Combining collaborative filtering with personal agents for better recommendations, in: Proceedings of 16th International Conference on Artificial Intelligence, 1999, pp. 439–446.
[23] Z. Huang, H. Chen, D. Zeng, Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, ACM Transactions on Information Systems 22 (1) (2004) 116–142.
[24] O. Kwon, J. Kim, Concept lattices for visualizing and generating user profiles for context-aware service recommendations, Expert Systems with Applications 36 (2) (2009) 1893–1902.
[25] D. Liu, C. Lai, W. Lee, A hybrid of sequential rules and collaborative filtering for product recommendation, Information Sciences 179 (20) (2009) 3505–3519.
[26] D. McGuinness, F. Harmelen, OWL: Web Ontology Language Overview, W3C Recommendation, 2004.
[27] S. Middleton, Capturing Knowledge of User Preferences with Recommender Systems, Ph.D. Thesis, Department of Electronic and Computer Science, University of Southampton, 2003.
[28] S.A. Mulaik, Foundation of Factor Analysis, Chapman & Hall, 2009.
[29] H. Oufaida, O. Nouali, Exploiting Semantic Web technologies for recommender systems: a multi view recommendation engine, in: Proceedings of 7th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, 2009, pp. 26–32.
[30] M. Pazzani, D. Billsus, Content-based recommendation systems, Lecture Notes in Computer Science 4321 (1) (2007) 325–341.
[31] F. Pittarello, The time-pillar worlds. A 3D paradigm for the new enlarged TV domain, in: L. Ardissono, A. Kobsa, M. Maybury (Eds.), Personalized Digital TV, Kluwer Academic Publisher, Dordrecht, 2004, pp. 287–320.
[32] C. Rocha, D. Schawabe, M. Poggi, A hybrid approach for searching in the Semantic Web, in: Proceedings of the 13th International World Wide Web Conference (WWW-04), 2004, pp. 74–84.
[33] J. Salter, N. Antonopoulos, CinemaScreen recommender agent: combining collaborative and content-based filtering, IEEE Intelligent Systems 21 (1) (2006) 35–41.
[34] P. Shaw, Multivariate Statistics for the Environmental Sciences, Wiley, 2003.
[35] N. Stojanovic, R. Struder, L. Stojanovic, An approach for the ranking of query results in the Semantic Web, in: Proceedings of the 2nd International Semantic Web Conference, 2003, pp. 500–516.
[36] J. Su, B. Wang, C. Hsiao, V. Tseng, Personalized rough-set-based recommendation by integrating multiple contents and collaborative information, Information Sciences 180 (1) (2010) 113–131.
[37] A. Troussov, M. Sogrin, J. Judge, D. Botvich, Mining socio-semantic networks using spreading activation techniques, in: Proceedings of the 8th International Conference on Knowledge Management and Knowledge Technologies, 2008, pp. 8–16.
[38] TV-Anytime Forum, TV-Anytime Specification Series: S-3 on Metadata, 2003. <http://www.tv-anytime.org>.
[39] J. Vertommen, F. Janssens, B. De Moor, J. Duflou, Multiple-vector user profiles in support of knowledge sharing, Information Sciences 178 (17) (2008) 3333–3346.
[40] R. Wand, F. Kong, Semantic-enhanced personalized recommender system, in: Proceedings of the International Conference on Machine Learning and Cybernetics, 2007, pp. 4069–4074.
[41] A. Zenebe, A. Norcio, Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems, Fuzzy Sets and Systems 160 (1) (2009) 76–94.
[42] C. Ziegler, G. Lausen, J. Konstan, On exploiting classification taxonomies in recommender systems, AI Communications 21 (2–3) (2008) 97–125.