Discerning Relevant Model Features in a Content-based Collaborative Recommender System

Alejandro Bellogín, Iván Cantador, Pablo Castells, and Álvaro Ortigosa

Escuela Politécnica Superior
Universidad Autónoma de Madrid
28049 Madrid, Spain
{alejandro.bellogin, ivan.cantador, pablo.castells,
alvaro.ortigosa}@uam.es

Abstract. Recommender systems suggest users information items they may be interested in. User profiles or usage data are compared with some reference characteristics, which may belong to the items (content-based approach), or to other users in the same context (collaborative filtering approach). These items are usually presented as a ranking, where the more relevant an item is predicted to be for a user, the higher it appears in the ranking. In this scenario, a preferential order has to be inferred, and therefore, preference learning methods can be naturally helpful. The relevant recommendation model features for the learning-based enhancements explored in this work comprise parameters of the recommendation algorithms, and user-related attributes. In the researched approach, machine learning techniques are used to discover which model features are relevant in providing accurate recommendations. The assessment of relevant model features, which is the focus of this paper, is envisioned as the first step in a learning cycle in which improved recommendation models are produced and executed after the discovery step, based on the findings that result from it.

Key words: Preference Learning, Recommender Systems, Machine Learning, Decision Trees, evaluation

1 Introduction

A recommender system suggests a user products or services he might be interested in. Tastes, interests and goals are explicitly declared by the user, or implicitly inferred by the system based on the user's behavior. User profiles and usage data are then compared to some reference characteristics, which might belong to the recommended items (in content-based approaches) [22], to the user's social environment (in collaborative filtering approaches, CF) [18,19], or to both information sources (in hybrid approaches) [6]. These comparisons usually result in numeric preference values that are used to rank (order) the suggested

items for the user. The recommendation process can thus be considered as an information-ranking problem where a suitable preference model, consisting of user interests, item content features, and system settings, has to be built.

In this context, research efforts to date can be said to have mainly focused on the study of the improvement of the recommendation algorithms by using all the available knowledge and profiling information. However, few studies have addressed the issue of finding out which of the preference model characteristics are actually most significant when accurate and non-accurate recommendations are generated. If these characteristics were identified, recommendation strategies could be enhanced by reinforcing or turning down their dependencies with specific stereotypes of users and items.

We thus envision the construction of a recommender system as a virtuous cycle with three main steps. First, an initial recommendation model is created with all the available information. This model is used to compute suggestions for the given user and item repositories. Next, the obtained outputs are analyzed in order to evince links between specific (input) model characteristics and the quality of recommendations. Finally, considering and adapting the identified characteristics, a new recommendation model, which is expected to generate more accurate results, is produced. The main challenge in this cycle, which is the focus of the research presented here, is in the second step, namely, how to discern (learn) those relevant preference model characteristics based on sets of system inputs, outputs, and user feedback.

In our proposed approach, Machine Learning (ML) techniques are used as a tool to determine which user and system characteristics are shared by most of the top items in a recommendation ranking. Specifically, for each recommendation evaluated (rated) by the user, a training sample is created. The attributes of the sample are the characteristics we aim to analyze, and their values are obtained from log information databases. The class of the training example can be assigned two possible values, correct and incorrect, depending on whether the user evaluated the corresponding recommendation as relevant or irrelevant. By classifying these examples, a ML algorithm facilitates the analysis of the above preference characteristics.

We have tested this proposal with News@hand [8], a news recommender system that suggests news articles according to several recommendation models, namely: 1) a personalized content-based model, the item suggestions from which are based on long-term user profiles [9], 2) a context-aware model that exploits user preferences which are not expressed in the user profile, but can be implicitly detected in the current user recommendation context [23], and 3) a collaborative model that finds and exploits implicit interest relations among users to provide enriched recommendations [7].

As described in the following, the identification of the user profile features and system settings from which each recommendation model should be executed is achieved by means of decision trees. The easy interpretability, the possibility of adding prior knowledge, and the selection of most informative attributes are

the main advantages brought by of the ML techniques to our recommendation mechanisms.

The rest of the chapter is organized as follows. Section 2 gives an overview of related works in which ML techniques have been applied to automatically learn preferences in personalized content retrieval, recommender and adaptive systems. Section 3 introduces News@hand, the news recommender system in which our preference analysis proposal is evaluated. Along with this system, the base recommendation algorithms, and the attributes that have been chosen for the analyzed samples are also described. Section 4 briefly explains decision trees, the ML techniques used in our proposal. Section 5 reports on the conducted experiments to evaluate the proposed approach. Finally, Section 6 concludes with some discussion and future research lines.

2 Related Work

ML techniques are useful when huge amounts of data have to be classified and analyzed, which nowadays is a very common situation in many scenarios, such as web information exploitation [20]. They have also proved to be of use in adaptive e-learning environments, where student data is used to adapt a system to user preferences and capabilities in order to facilitate the learning process. Hence, for example, in Becker and Marquardt's work [3], students' logs are analyzed with the goal of finding patterns that reveal the system browsing paths followed by students. Talavera and Gaudioso [21] use classification techniques to analyze student behavior in a cooperative learning environment. Their main goal is to discover patterns that reflect the students' behavior, supporting tutoring activities on virtual learning communities.

Other authors have also investigated the application of these techniques to Recommender Systems [28], evaluating the performance of personalization mechanisms, particularly Adaptive Hypermedia Systems (AHS) and Adaptive Educational Systems (AES) [5]. For example, Zaïane proposed using association rules in AEH domains [27]. His work focuses on two basic points: the first point is to give automated support to students who take an online course proposing the use of advising systems; the second is to support the instructor in identifying student behavior patterns, based on the information that students provide when taking online courses. In the same context, Vialardi et al. [25] use data mining techniques to discover and present relevant pedagogic knowledge to the teachers. They propose to use classification trees and association rules to detect opportunities for improvement on the adaptation decisions of an AES. In [2], several examples where ML techniques are used to learn a user model (based on previous ratings) and classify unseen items are explained. A review of these techniques is also given by Adomavicius and Tuzhilin in [2], where Decision Trees, Clustering, Artificial Neural Networks and Bayesian classifiers are mentioned. Our system also takes into consideration the current user's interest context [23], which is similar to the idea of using short and long term profiles explained in [17].

Despite the above works, to our knowledge, there have been few attempts to use ML techniques as we propose here. In our approach, ML techniques are used to evaluate the system to make explicit improvement on its performance. In this way, we are more interested in the model generated (which variables are more informative, which can be discarded by the model, etc.) by the ML techniques than in the classification itself. This is different from the above approaches, where ML techniques are used as an integrated part of the (recommender, learning) system. Nevertheless, a similar idea can be seen in [24], where ML techniques find patterns for assisting adaptive hypermedia authors during design and evaluation phases. The authors build a model representing the student behavior on a particular course, and use it to obtain and exploit a vision of the behavior and performance of student groups.

3 News@hand: a news recommender system

News@hand is a news recommender system that combines textual features and collaborative information to make news suggestions, and uses a controlled and structured vocabulary to describe user preferences and news contents. For this purpose, it makes use of Semantic Web technologies. News items and user profiles are represented in terms of concepts appearing in domain ontologies. For example, a news item about a particular football match could be annotated with general concepts as "football" and "match", or specific instances of football teams and players (e.g., *Real Madrid F.C.*, *Zinedine Zidane*).

More specifically, user preferences are described as vectors $\mathbf{u}_m = (u_{m,1}, u_{m,2}, \ldots, u_{m,K})$ where $u_{m,k} \in [-1, 1]$ measures the intensity of the interest of user $u_m \in \mathcal{U}$ for concept $c_k \in \mathcal{O}$ (a class or an instance) in a domain ontology $\mathcal{O}$, K being the total number of concepts in the ontology. Similarly, items $d_n \in \mathcal{D}$ are assumed to be annotated by vectors $\mathbf{d}_n = (d_{n,1}, d_{n,2}, \ldots, d_{n,K})$ of concept weights, in the same vector-space as user preferences.

Ontology concept-based preferences are more precise, and reduce the effect of the ambiguity caused by simple keyword terms. For instance, if a user states an interest for the keyword "java", a system might not have information to distinguish *Java, the programming language*, from *Java, the Pacific island*. However, a preference stated as "ProgrammingLanguage:Java" (this is read as the instance Java from the Programming Language class) lets the system understand unambiguously the preference of the user, and also allows the exploitation of more appropriate related semantics (e.g., synonym, hypernym, subsumption, etc.). This, together with disambiguation techniques, might lead to the effective recommendation of text-annotated items.

In News@hand (Figure 1), news items are classified in 8 different sections: headlines, world, business, technology, science, health, sports and entertainment. When the user is not logged in the system, he can browse any of the previous sections, but the items are listed without any personalization criterion. He can only sort them by their publication date, source, or level of popularity (i.e., according to a classic rating-based CF mechanism). On the other hand, when the user

is logged in the system, recommendation and profile edition functionalities are enabled, and the user can browse the news according to his and others' semantic preferences in different ways. Short and long term preferences are considered. Click history is used to define the short-term user preferences, and the resultant rankings can be adapted to the current context of interest.

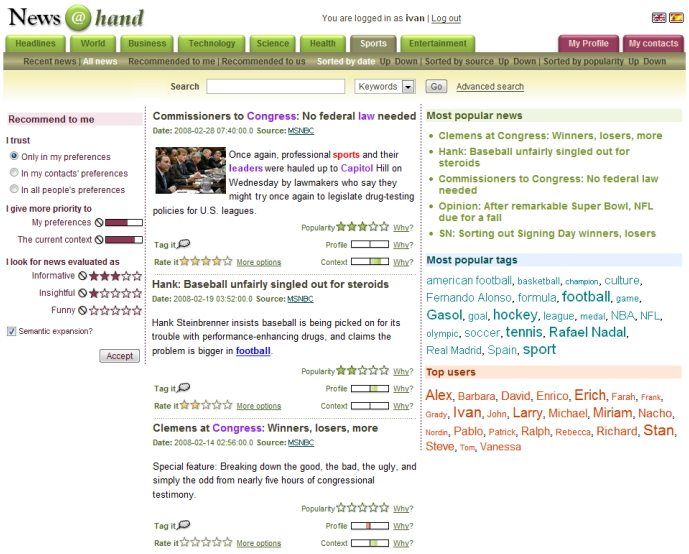

Fig. 1. A typical news recommendation page in News@hand system

Characteristics such as the topic section, the type of recommendation (personalized, context-aware, collaborative), and the number of the page in which accurate recommendations appear are analyzed by our preference learning proposal.

3.1 Semantic expansion of preference

Semantic relations among concepts are exploited to enrich the proposed ontology-based knowledge representations, and are incorporated within the recommendation processes. For instance, a user interested in animals (superclass of dog) is also recommended items about dogs. Inversely, a user interested in skiing, snowboarding and ice hockey can be inferred with a certain confidence to be globally interested in winter sports. Also, a user keen on Spain can be assumed to like Madrid, through locatedIn transitive relation, assuming that this relation had been seen as relevant for inferring previous underlying user's interests.

We have developed [23] a semantic preference spreading mechanism that expands the initial set of preferences stored in user profiles through explicit

semantic relations with other concepts in the ontology (Figure 2). The approach is based on the so-called Constrained Spreading Activation (CSA) strategy [13,14,15]. The expansion is self-controlled by applying a decay factor to the intensity of preference each time a relation is traversed, and taking into account constraints (threshold weights) during the spreading process.

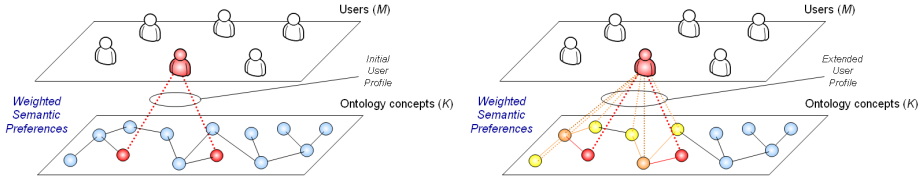

Fig. 2. Semantic preference extension

News@hand recommendation models output ranked lists of content items taking into account not only the initial user profiles, but also the semantic extension of user preferences and item annotations. The question of whether our semantic expansion technique really benefits the obtaining of more accurate item suggestions is in fact one preference characteristic we analyze in this work.

3.2 Architecture

Figure 3 depicts how ontology-based item descriptions and user profiles are created in News@hand. News items are automatic and periodically retrieved from several on-line news services via RSS feeds. The title, summary and category of the retrieved news are then annotated with concepts of the system domain ontologies. Thus, for example, all the news about actors, actresses, and similar terms might be annotated with the concept "actor". A TF-IDF technique is applied to assign weights to the annotated concepts.

With a client/server architecture, users utilize a web interface to receive on-line news recommendations, and update their profiles. A dynamic graphical interface allows the system to automatically store all the users' inputs, analyze their behavior, and adjust the news recommendations in real time. Explicit and implicit user interests are taking into account, via manual preferences, tags and ratings, and via automatic learning from the users' actions.

Deriving benefit from the semantically annotated news items, the defined ontology-based user profiles, and the knowledge represented by the domain ontologies, a set of recommendation algorithms is executed. Among other approaches, News@hand offers personalized [23], context-aware [9] and collaborative multi-facet recommendations [7]. Configurations and combinations of the above recommendation models are model feature characteristics included in the study presented herein.

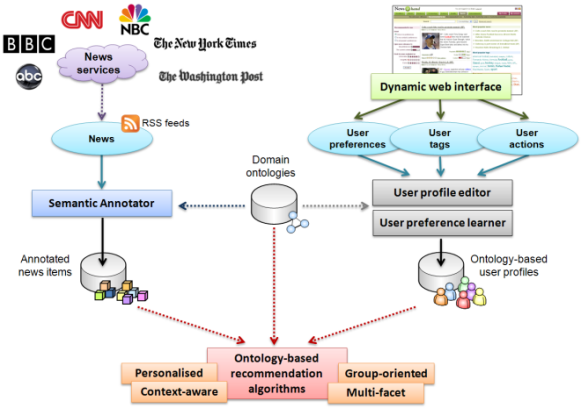

Fig. 3. Architecture of News@hand system

3.3 Content-based recommendations

Our notion of personalized content retrieval is based on a matching algorithm that provides a relevance measure $\text{pref}(u_m, d_n)$ of an item d_n. for a user u_m. This measure is set according to the semantic preferences of the user and the semantic annotations of the item and based on cosine-based vector similarities

$$\text{pref}(d_n, u_m) = \cos(\mathbf{d}_n, \mathbf{u}_m) = \mathbf{d}_n \cdot \mathbf{u}_m / \|\mathbf{d}_n\| \times \|\mathbf{u}_m\|$$

The formula matches two weighted-concept vectors, and produces a value in $[-1, +1]$. Values close to -1 are obtained when the vectors are dissimilar, and indicate that user preferences negatively match the content metadata. On the other hand, values close to $+1$ indicate that user preferences significantly match the content metadata, which means a potential interest of the user for the item.

The content-based recommendation results can be combined with query-based scores without personalization [12], and semantic context information, to produce combined rankings. This last approach is described in the next section.

In this model, the size of the user profile will be a reference characteristic to be studied when accurate recommendations are obtained.

3.4 Context-aware recommendations

We propose a particular notion of context, useful in semantic content retrieval: that of semantic runtime context, which we define as the background topics under which user activities occur within a given unit of time. A runtime context is represented in our approach [9,23] as a set of weighted concepts from the domain ontologies. This set is obtained by collecting the concepts that have been involved in the interaction of the user (e.g., accessed items) during a session.

The context is built in such a way that the importance (weight) of concepts fades away with time (number of accesses back when the concept was referenced)

by a decay factor ξ in $[0,1]$:

$$C_m^t[c_k] = \xi \cdot C_m^{t-1}[c_k] + (1-\xi) \cdot \text{Req}^t[c_k]$$

where $\text{Req}^t[c_k]$in $[0,1]^K$ is a vector whose components measure the degree in which the concepts c_k are involved in the user's request at time t. This vector can be defined in multiple ways, depending on the application: a query concept-vector (if a request is expressed in term of a concept-based search query), a concept vector containing the most relevant concepts in a document (if a request is a "view document" request), the average concept-vector corresponding to a set of items marked as relevant by the user (if a request is a *relevance feedback* step), etc.

Once the context is built, a contextual activation of preferences is achieved by finding semantic paths linking preferences to context, as follows:

$$\begin{aligned} \text{pref}_{C^t}(d_n, u_m) &= \lambda \cdot \text{pref}(d_n, u_m) + (1-\lambda) \cdot \text{sim}(d_n, C_t) \\ &= \lambda \cdot \cos(d_n, EU_m) + (1-\lambda) \cdot \cos(d_n, EC_t) \end{aligned}$$

where λ in $[0,1]$ measures the strength of the personalization component with respect to the current context. This parameter could be manually established by the user, or dynamically adapted by the system according to multiple factors, such as the current size of the context, the automatic detection of a change in the user's search focus, etc.

The perceived effect of contextualization is that user interests that are out of focus, under a given context, are disregarded, reinforcing those that are in the semantic scope of the ongoing user activity are considered for recommendation (see Figure 4).

Analogously to the personalization model, where the size of the user profile is a critical aspect, the context-aware recommendation approach will be affected by the size and precision of the current semantic context. These characteristics will be also included in the analytical experiments.

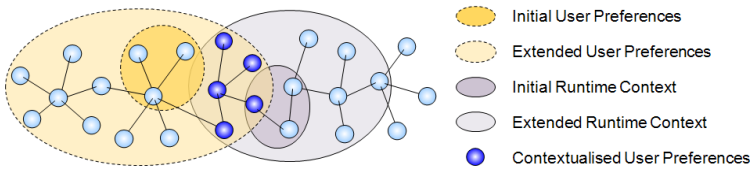

Fig. 4. Contextualization of user preferences

3.5 Collaborative recommendations

Collaborative filtering techniques match people with similar preferences in order to make recommendations. Unlike content-based methods, collaborative recom-

mender systems aim to predict the utility of items for a particular user according to the items previously evaluated by others [18,19]. One of the main benefits of these approaches is the possibility to recommend items that do not share features with respect to the items rated in the past by the user. However, these approaches introduce certain problems [1]; for example, a new item cannot be recommended to a user until other users rate it.

The utility gain function $g(u_m, i_n)$ of item $i_n \in \mathcal{I}$ for user $u_m \in \mathcal{U}$ is estimated based on the utilities $g(u_j, i_n)$ assigned to item i_n by those users u_j that are "similar" to user u_m. In this work, we use two different well-known collaborative filtering approaches: user-based and item-based [18,19]. In the first situation, the following approach has been taken:

$$g(u_m, i_n) = \frac{\sum_{u_j \in \hat{\mathcal{U}}_m} \text{sim}(u_m, u_j) \times r_{j,n}}{\sum_{u_j \in \hat{\mathcal{U}}_m} |\text{sim}(u_m, u_j)|},$$

$$\text{sim}(u_m, u_j) = \frac{\sum_{i_n \in \mathcal{I}_{m,j}} (r_{m,n} - \bar{r}_m) \cdot (r_{j,n} - \bar{r}_j)}{\sqrt{\sum_{i_n \in \mathcal{I}_{m,j}} (r_{m,n} - \bar{r}_m)^2} \sqrt{\sum_{i_n \in \mathcal{I}_{m,j}} (r_{j,n} - \bar{r}_j)^2}}$$

where the similarity function is called Pearson correlation.

In the item-based situation, we use a similar formulation:

$$g(u_m, i_n) = \frac{\sum_{i_j \in \hat{\mathcal{I}}_n} \text{sim}(i_n, i_j) \times r_{m,j}}{\sum_{i_j \in \hat{\mathcal{I}}_n} |\text{sim}(i_n, i_j)|},$$

$$\text{sim}(i_n, i_j) = \frac{\sum_{u_m \in \mathcal{U}_{n,j}} (r_{m,n} - \bar{r}_n) \cdot (r_{m,j} - \bar{r}_j)}{\sqrt{\sum_{u_m \in \mathcal{U}_{n,j}} (r_{m,n} - \bar{r}_n)^2} \sqrt{\sum_{u_m \in \mathcal{U}_{n,j}} (r_{m,j} - \bar{r}_j)^2}}$$

The predicted value $g(u_m, i_n)$ is a very solid information source in order to know if the above algorithms would work in a real scenario, so we will study it in our experiments, along with the type of collaborative filtering technique used.

3.6 Log database

The system monitors all the actions the user performs, and gathers them in a log database. Table 1 shows the attributes of the database tables.

Table 1. Summary of the log database tables and attributes. Session id is an inter-table identifier, whilst action id is an intra-table attribute. Action type is a string distinguishing between different actions a table can contain (for instance, LOGIN and LOGOUT are stored in user accesses table).

Table	Attributes
Browsing	actionID, actionType, timestamp, sessionID, itemID, itemRankingPosition, itemRankingProfile, itemRankingContext, itemRankingCollaborative, itemRankingHybridUP, itemRankingHybridNUP, itemRankingHybridUPq, itemRankingHybridNUPq, topicSection, interestSituation, userProfileWeight, contextWeight, collaborative, scoreSearch
Context updates	actionID, actionType, timestamp, sessionID, context, origin, changeOfFocus
Queries	actionID, actionType, timestamp, sessionID, keywords, topicSection, interestSituation
Recommendations	actionID, actionType, timestamp, sessionID, recommendationType, userProfileWeight, contextWeight, collaborative, topicSection, interestSituation
User accesses	actionID, actionType, timestamp, sessionID
User evaluations	actionID, actionType, timestamp, sessionID, itemID, rating, userFeedback, tags, comments, topicSection, interestSituation, duration
User preferences	actionID, actionType, timestamp, sessionID, concept, weight, interestSituation
User profiles	actionID, actionType, timestamp, sessionID, userProfile
User sessions	sessionID, userID, timestamp

In this work, we focus on the user evaluation and browsing tables, which respectively store information about ratings and rated items, and system configurations for specific actions. The database tables share a session identifier that allows us to recognize relationships among actions. More specifically, given a row from the user evaluation table, we extract the session identifier, the rated item, and the action timestamp in order to infer which system configuration was at that moment, as follows:

1. Get all the browsing actions matching a given session identifier.
2. Select the actions with the same item identifier, previously extracted from the browsing table.
3. Use the timestamp to obtain the system configuration, such as user profile weight (0 if personalization is off), and context weight.

4 Decision Trees for Model Feature Learning

The main goal of our research is twofold: the creation of training samples that correspond to positive (relevant) and negative (non-relevant) recommendation cases, and the analysis of these samples with ML techniques in order to determine which model features seem to be most significant to provide either positive and negative recommendations.

In this section, we describe the ML algorithms applied for our model feature learning purposes. We focus on one of these techniques: Decision Trees. However, a previous work also explored Attribute Selection technique [4]. Information for creating the samples is obtained from the log database introduced in section 3.6.

Decision Trees apply a divide-and-conquer strategy for producing classifiers with the following benefits [16]:

- They are interpretable.
- They enable an easy attachment of prior knowledge from human expert.
- They tend to select the most informative attributes measuring their entropy, boosting them to their top levels.
- They are useful for non-metric data (the represented queries do not require any notion of metric, as they can be asked in a "yes/no", "true/false" or other discrete value set representations).

However, despite these advantages, Decision Trees are usually over-fitted and might not generalize well to independent test sets. Two possible solutions are applicable: stopped splitting and pruning. C4.5 is one of the most common algorithms to build Decision Trees, and uses heuristics for pruning based on statistical significance of splits. In the experiments, we make use of its well-known revision J4.8.

It is worth noting that in this paper we are interested in the model generated by this classifier, instead of its predictive power. Proceeding in this way, Decision Trees will show which attributes are more informative (those appearing at the top of the tree), and which of their values tend to classify an instance as positive or negative.

5 Experiments

The experiments have been conducted using News@hand system, presented in Section 3. In the following, a description of the system item database and knowledge repository is provided. We also explain the two different experiments performed (*stages* from now on), including the tasks and phases fulfilled by users during the evaluation, and conclude with the obtained results.

5.1 News item database and Knowledge repository

For two months, RSS feeds were collected on a daily basis. A total of 9,698 news items were stored. With this dataset, we run our semantic annotation mechanism

mentioned in section 3.2, and a total of 66,378 annotations were obtained. For more details, see [11].

A set of 17 ontologies is used by the current version of the system. They are adaptations of the IPTC ontology[1], which contains concepts of multiple domains such as education, culture, politics, religion, science, technology, business, health, entertainment, sports, weather, etc. They have been populated with concepts appearing in the gathered news items using semantic information from Wikipedia, and applying a population mechanism explained in [11]. A total of 137,254 Wikipedia entries were used to populate 744 ontology classes with 121,135 instances.

5.2 Experimental setup

Two different stages have been designed in order to discover which model features are relevant in providing accurate recommendations. The first one is focused on personalization functionalities, in particular: ontology-based content retrieval, and semantic context-aware personalization. Ontology-based content retrieval is tested against a keyword-based approach, whilst context-aware personalization is turned on and off in order to investigate its contribution to the user's experience. Another important part of these methods has also been evaluated: semantic expansion of preferences.

In the second stage, we analyze which features of our model are more influential when using a collaborative filtering algorithm. With this objective in mind, we have integrated two well-known, state-of-the-art collaborative filtering algorithms into the system, and studied their discriminative power for classifying a news item as relevant or irrelevant.

First stage: Evaluation of content-based and context-aware recommendation

In this section, we present a first experiment conducted to evaluate the precision of the personalization and the context-aware recommendation functionalities available in News@hand (sections 3.3 and 3.4). We also aimed to investigate the influence of each mechanism in the integrated system, measuring the precision of the recommendations when a combination of both models is used. 16 members of our department were requested to participate. They were 12 undergraduate/graduate students, and 4 lecturers.

The experiment comprised two phases, each composed of two different tasks. In the first phase, only the personalization module was active, and the tasks were different in having the semantic expansion (see section 3.1) enabled or disabled. In the second phase, the contextualization and semantic expansion functionalities were active. In its second task, the personalized recommendations were also enabled. More details are given in the next subsection.

[1] IPTC ontology, `http://nets.ii.uam.es/mesh/news-at-hand/news-at-hand_iptc-kb_v01.zip`

Table 2. Summary of the search tasks performed in the experiment.

Profile	Section	Query		Task goal
1	*World*	$Q_{1,1}$	pakistan	News about media: TV, radio, Internet
Telecom	*Entertainment*	$Q_{1,2}$	music	News about software piracy, illegal downloads, file sharing
2	*Business*	$Q_{2,1}$	dollar	News about oil prices
Banking	*Headlines*	$Q_{2,2}$	fraud	News about money losses
3	*Science*	$Q_{3,1}$	food	News about cloning
Social care	*Headlines*	$Q_{3,2}$	internet	News about children, young people, child safety, child abuse

A task was defined as finding and evaluating those news items that were relevant to a given goal. Each goal was framed in a specific domain, and we considered three domains: telecommunications, banking and social care issues. For each domain, a user profile and two search goals were set as explained below. Table 2 shows a summary of the involved tasks.

To simplify the searching tasks, they were defined for a pre-established section and query. Hence, for example, the task goal of finding news items about software piracy, illegal downloads and file sharing, Q1,2, was reduced to evaluate those articles existing in *Entertainment* section that were retrieved with the query "music".

In order to cover as many system configurations as possible with the available users, the assignment of the tasks was set according to the following principles:

- A user should not repeat a query during the experiment.
- The domains should be equally covered by each experiment phase.
- A user has to manually define a user profile once in the experiment.

For each phase, the combination of personalized and context-aware recommendations was established as a linear combination of their results using two weights $w_p, w_c \in [0, 1]$:

$$\text{score}(d_n, u_m) = w_p \cdot \text{pref}(d_n, u_m) + w_c \cdot \text{pref}(d_n, u_m, \text{context})$$

In the personalization phase, the contextualization was disabled (i.e., $w_c = 0$). Its first tasks were performed without semantic expansion, and its second tasks had the semantic expansion activated. In the contextualization phase, wc was set to 1, and the expansion was enabled. Its first tasks were done without personalization (wp=0), and its second tasks were influenced by the corresponding profiles (wp=0.5).

As mentioned before, a fixed user profile was used for each domain. Some of them were predefined profiles, and others were created by the users during the experiment, using the profile editor of News@hand. In addition, some tasks were done with user profiles containing concepts belonging to all the three domains.

There is also an important issue about how the users rated. Every time the user read an item, he had to assess whether the item was relevant to the profile, to the current goal, or to both/neither of them. In each situation, a different rating criterion was defined:

- Rate with 1 star if the item was not relevant.
- Rate with 2 stars if the item was relevant to the current goal.
- Rate with 3 stars if the item was relevant to the profile.
- Rate with 4 stars if the item was relevant to the current goal and the profile.

These rating constraints gave us a bounded frame for evaluation. In the next subsections, it will be shown that they also allowed us to have different criteria to set the class values of the training samples.

Content-based phase The objective of the two tasks performed in the first experiment phase was to evaluate the importance of activating the semantic expansion of our recommendation models. The following are the steps the users had to do in these tasks:

- Launch the query with the personalization module deactivated.
- Rate the top 15 news items.
- Launch the query with the personalization module activated (and the semantic expansion enabled/disabled depending on the case).
- Rate again the top 15 news items.

At the end of this phase, each user had rated 30 items with expansion enabled and 30 with expansion disabled.

Contextualization phase The objective of the two tasks performed for the second experiment phase was to evaluate the quality of the results when the contextualization functionality is activated and combined with personalization. The steps done in this case are the following:

- Launch the query with the contextualization activated (semantic expansion enabled, and personalization enabled/disabled depending on the case).
- Rate the top 15 news items, and evaluate as relevant (clicking the title) the first item related to the task goal. Doing this the current semantic context is updated.
- Repeat the last two steps twice (the last time it is not necessary to update the context, since the evaluation will not continue).

At the end of this phase, each user had rated 45 items with personalization on and 45 items with personalization off. He had also evaluated as relevant 4 news items that were incorporated into the context.

Selection of sample attributes and classes based on evaluation parameters Each user had to assign a rating depending on the four existing possibilities for each news item: relevant to the goal (2), the profile (3), both (4), and neither of them (1). Considering these four options, we defined three different criteria to classify an item (sample) as relevant:

- The item is relevant in general, if the user has rated it with 2, 3 or 4.
- The item is relevant to the current goal, if the user has rated it with 2 or 4.
- The item is relevant to the profile, if the user has rated it with 3 or 4.

In this work, we focus on the second criterion, although a preliminary analysis with the first one is also tested because of its generality.

In addition to the sample classes, according to the evaluation made, we selected those attributes whose impact on the recommendations we wanted to analyze. For each item rating log entry, we chose several attributes that can be categorized as follows:

User-based features

- *Profile type*: a string attribute with two possible values: fixed or used-defined preferences (manual preferences).
- *Profile size*: an integer attribute indicating the total number of concepts included in the profile (number of non-zero components in the vector representation).
- *Context size*: an integer attribute indicating the number of concepts included in the current context.

Model-based features

- *Topic section*: name of the news section in which the rated item appeared.
- *Ranking result page*: number of the page in which the rated item appeared. Each page shows five news items.
- *Personalized recommendations*: a Boolean value indicating whether the personalized recommender was activated or deactivated.
- *Context-aware recommendations*: a Boolean value indicating whether the context-aware recommender was activated or deactivated.
- *Semantic preference expansion*: a Boolean value indicating whether the expansion of user preferences and item annotations was activated or not.
- *Context-aware phase*: a number indicating how many times the user has clicked as relevant an item when the context is activated. A value of -1 is given if the context-aware recommendations are off.

Second stage: evaluation of collaborative recommendation

A second experiment was conducted with News@hand to evaluate the collaborative recommendation models included in the system. One of the objectives of this experiment was to compare the relevance judgments given by the users with

Table 3. Topics and concepts allowed for the user profiles in the evaluation of the hybrid recommenders

Domain	Concepts	#prefs	Avg. #pref./user
Computers Technology Telecommunications	computer, digital, ebay, google, ibm, internet, mass, media, microsoft, networking, online, satellite, software, technology, video, website	135	8.4
Wars Armed conflicts	al-qaeda, army, battle, combat, crime, kidnapping, kill, memorial, military, murder, peace, prison, strike, terrorism, war, weapons	104	6.5
Social issues	aids, assassination, babies, children, death sentence, divorce, drugs, family, health, hospital, immigration, love, obesity, smoking, suburb, suicide	115	7.2
Television Cinema Music	actor, bbc, cinema, cnn, film, grammy, hollywood, movie, music, musician, nbc, radio, rock, oscar, singer, television	129	8.1
Sports	baseball, cricket, football, lakers, nascar, nba, new england patriots, new york giants, nfl, olympics, premier league, running, sports, soccer, super bowl, tennis	168	10.5
Politics	george bush, condolezza rice, congress, democracy, elections, government, hillary clinton, john maccain, barack obama, parliament, politics, president, senate, senator, voting, white house	104	6.5
Banking Economy Finance	banking, business, cash, companies, earnings, economy, employment, finance, fraud, gas price, industry, marketing, markets, money, oil price, wall street	120	7.5
Climate Weather Natural disasters	air, climate, earth, earthquake, electricity, energy, fire, flood, forecast, fuel, gas, pollution, sea, storm, weather, woods	128	8.0

the recommendations obtained using the CF approach explained in section 3.5. The comparison will be given by the model built applying the ML techniques explained in section 4, in such a way that CF values (potential recommended items) should be correlated with the relevance judgments, at least, when certain model conditions are fulfilled.

The 16 members of our department who participated in the previous experiment were again requested to take part of the evaluation presented herein. Each user performed three different tasks, assessing news recommendations for three news sections: *Business*, *Sports* and *World* (see below why we selected these sections). For each task, two subtasks were defined:

- In the first subtask, the users had to rate a number of news items from a random list.
- In the second subtask, the users had to rate several news items from a list generated with the personalization functionality activated.

Each subtask was defined as finding out and rating those news items that were "related to" a personal user profile. By "related to" we mean that a news item contains semantic annotations whose concepts appear in the user's profile.

Similarly to the experiment described in previous section, the evaluators were asked to define their preferences. However, in this case, they could only select preferences from a given list of semantic concepts. They were provided a form with a list of 128 semantic concepts, classified in 8 different domains. From this list the users had to select a subset of concepts, and assign them negative/positive weights according to personal interests. Table 3 shows the concepts available for each domain, and the average number of preferences per user. On average, each profile was created with 7.8 preferences per domain, duplicating the preferences introduced by the users when they had to manually search the concepts in the ontology browser (first stage).

In the next subsections, we explain in detail the different tasks performed by the users, and the data extracted from their interaction with the system in order to draw appropriate conclusions.

Interaction with the system The users had to perform three tasks, each of them in one of the following news sections: *Business*, *Sports* and *World*. Successively, for each section, a user had to:

- Deactivate the personalization functionality, and display the news items of the section. The goal is to present to all the users the same set of news items, in order to obtain a "shared" group of rated items (this is very important for the collaborative filtering model, since this step reduces the sparsity).
- Rate 20 news items that are related (with negative or positive weights) to the user profile. Taking into account the similarities between item annotations with user preferences, assign a 1-5 start rating to the selected news items. No restriction is placed on which items have to be rated.
- Activate the personalization functionality, and display again the news items of the section. This time the order (ranking) of the news items is different to the one shown previously.
- Rate (as explained before) 50 news items not evaluated previously.

With this strategy, the 16 users provided a total of 3,360 ratings for 859 different news items.

Selection of training sample attributes The training sample creation was performed similarly as in the previous experiment, but since the experiment setup was different, the attributes to consider also changed. For example, this time, context was not evaluated, all the profiles were manually defined, and

semantic preference expansion was always activated. After this simplification, the relevant sample attributes for this experiment are the following:

User-based features

- *Profile size*: an integer attribute indicating the total number of concepts included in the profile (the same as before).

Model-based features

- *Topic section*: name of the news section in which the rated item appeared.
- *Ranking result page*: number of the page in which the rated item appeared. Each page shows five news items.
- *Personalized recommendations*: a Boolean value indicating whether the personalized recommender was activated or deactivated.
- *Collaborative filtering algorithm*: since we consider two different collaborative filtering algorithms, this is a nominal attribute: user-based and item-based. However, we have preferred to combine samples with values obtained from the same algorithm to facilitate the ML algorithm classification task. This means that we have two sets of samples: one for each collaborative filtering algorithm.
- *Collaborative filtering value*: a number indicating the value given by the recommender system for a particular pair of user and item, given the rest of user ratings (predicted value, see section 3.5).

5.3 Results

This section presents the results obtained using ML techniques to analyze the evaluations previously described. These results are classified into three different categories, according to the consequences that can be drawn from them. Firstly, we present the results related to the personalization phase (first stage), where the impact of the semantic expansion is considered. Secondly, the contextualization phase (first stage) results are presented, where the importance of context combined with personalization is studied. Finally, results related to the collaborative filtering phase are shown (second stage), where correlation between predicted and real ratings is analyzed. Furthermore, some conclusions about the evaluation itself are shown in the discussion section.

For developing these results, Weka ML toolkit [26] and Taste library[2] were used. The Decision Trees presented in the figures of this section were generated using different parameters according to the stage they refer. Specifically, in the first stage we generated the trees using the following parameters[3]: "*-C 0.3 -M 5*", whereas in the second stage we used "*-R -N 3 -M 25*". Although we tried with

[2] `http://taste.sourceforge.net/`

[3] The meaning of these parameters is the following: '-C' sets confidence threshold for pruning, '-R' creates a decision tree using reduced error pruning, when this option is available, '-N' sets the number of folds used for reduced error pruning.

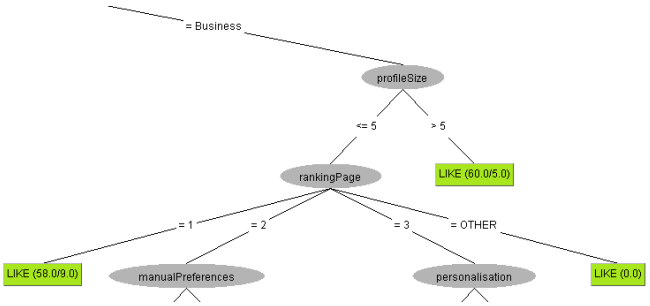

Fig. 5. Branch when profile size is less than 12, using all available logs (general evaluation)

different configurations, we chose the ones presented here because they generate comprehensible but detailed trees.

Furthermore, we have to note that every decision tree presented in this work report more than a 69% of predictive accuracy in a 10-fold cross-validation experiment (the maximum accuracy obtained is 80%). Based on these results, we can assume the induced models by the decision trees are trustworthy enough to obtain reliable results.

Learning model features from personalized recommendations

In the personalization phase, we wanted to investigate whether the semantic expansion helps the user to find relevant news. After using ML techniques we found more useful user and system features:

- **Profile size.** In Figures 5 and 6, it can be seen that this user feature is useful when retrieving relevant items, and is connected with expansion and activation of personalization. In Figure 5 we can see that, in the *Business* section, if the profile size is between 5 and 12 concepts, it is very likely that user will find relevant news. On the other hand, in Figure 6, it can be seen that a small profile produces more irrelevant news to be retrieved.
- **Ranking page.** Fortunately, the system retrieves relevant news in the first page (top 5 news items), as shown in Figures 5 and 7. Because of that, our analysis focused on sub-trees where the ranking page has a value of 1.
- **Expansion.** The importance of this model feature is shown in Figure 7, where users find relevant news only when personalization and expansion are activated.

In general, we have found that using personalization in combination with semantic expansion improves the performance in the first page. Although not all the news sections behave equally, this seems to be true in general sections such as Headlines, despite the fact that in the second and third pages, personalization improves little and needs the help of other strategies, such as contextualization (see next subsection).

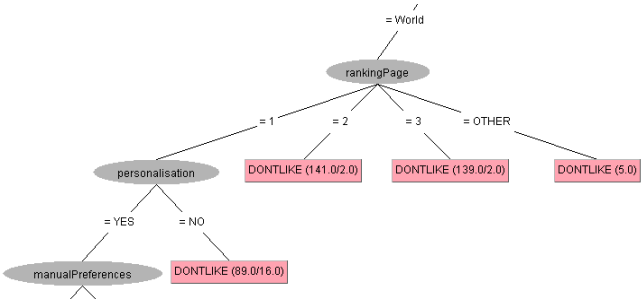

Fig. 6. Branch when profile size is less than 12, using all available logs (general evaluation)

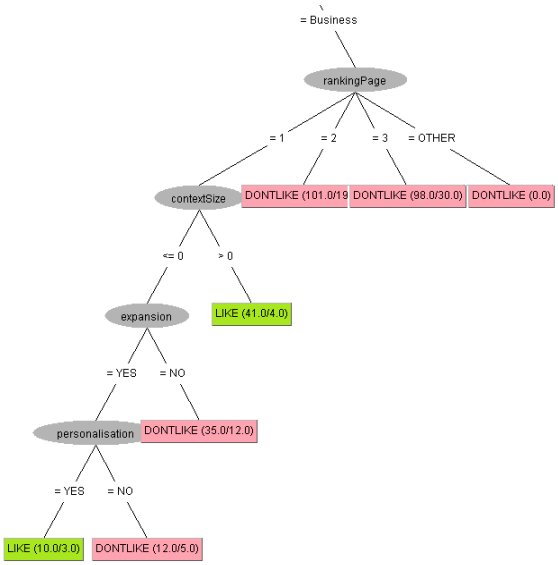

Fig. 7. Business branch and using all available logs (goal evaluation)

Learning model features from context-aware recommendations

In the experiments, we found some model features are more likely to help in context-aware recommendations. For instance, personalization was a well-performing system setting when it is combined with context. Although sometimes context alone performs well (Figure 7), in Figure 8 we show an example where context needs personalization to obtain good results.

Another relevant indicator is the context size (Figure 8). In previous experiments [4], it showed better discrimination power, but in the current ones, its main function is to distinguish between when the context was on or off. A model feature that does not have influence in context is the fact of having manual pref-

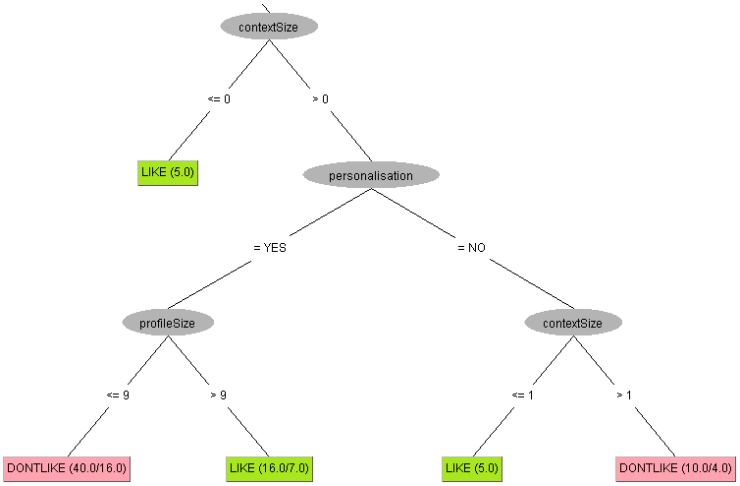

Fig. 8. Science branch in the third ranking page, using context-related logs (goal evaluation)

erences or not, since the context has more to do with the short term preferences, rather than long term ones.

Learning model features from collaborative recommendations

The goal of the second stage was to investigate whether collaborative filtering predicted values correlates with human relevance judgments or not. A first conclusion can be drawn after analyzing Figures 9, 10 and 11: in most cases, collaborative filtering algorithms predict successfully the relevant class of a news item. However, a different behavior can be found between the two algorithms used here (item-based and user-based). If we focus on Figures 10 and 11, we can see differences in the *Topic section* classification. *Business* is a very specific section, and most of its items are very similar. Item-based algorithm lacks of information, and needs the assistance of other methods to be able to classify items as relevant or irrelevant. At the same time, *World* section is a very general section, containing objects of different types, which gives a lot of information in order to fulfill its goal. User-based algorithm behaves quite the opposite, which gives us the results shown in Figures 9 and 10. *Sports* section has to be left aside, since most of the users choose no concepts related with this section, resulting in irrelevant news retrieved by the system very often.

The predictive power of these algorithms can be seen in Figure 11, where, in this case, two threshold values can be inferred, in order to guess if the news item will be relevant or not. In this figure, these two values are 2.752 (above this value a news item can be considered relevant) and 1.967 (below this value news are irrelevant with a great confidence). A similar situation happens in Figure 10, where only the threshold to set irrelevant news is shown. This situation allows

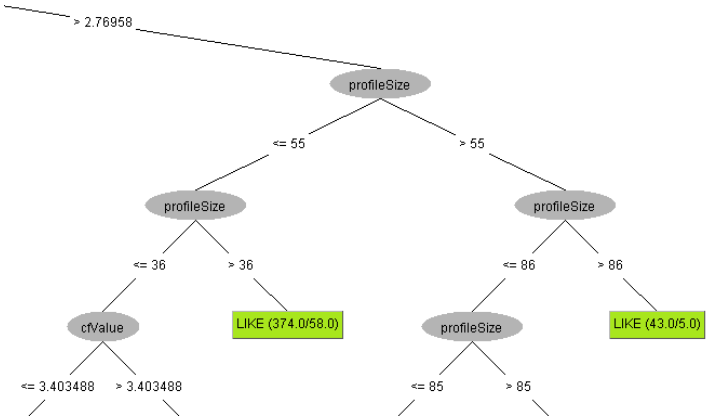

Fig. 9. Branch with CF value greater than 2.76, using a user-based algorithm

us to focus on a more limited set of news items: the ones located between the two thresholds. This set is not large, but ambiguous, since a value of 2.5 can be positive for one person but negative for another one. It is worthwhile noting personalization methods are indeed helpful in order to discriminate relevant news items pertaining to this small set.

Profile size has been found to be a very informative feature (see Figure 9). Another discriminative feature is the ranking page (Figure 10), although in this experiment it has less classification power than in the previous ones already explained (compare with Figure 7, for example).

Finally, personalization confirms its utility once again. For example, in Figure 11, activated personalization helps the collaborative filtering algorithm to classify an item as relevant when the predicted value is ambiguous (and, in this case, along with a profile not too big).

In general, we have found that the values predicted by collaborative filtering algorithms are very close to real ones. Indeed, if a predicted value is extreme (above 3.5 or below 1.5), in most of the cases, we can be confident of that, and classify the item accordingly. This situation is improved when it is combined with our personalization algorithms. In a future analysis, we have to verify if such a combination with our context and expansion models also leads to similar improvements.

6 Discussion

We have presented a method for the automatic, iterative refinement of a recommender system by a virtuous cycle with three main steps. First, an initial recommendation model is run on a set of available input data, to compute suggestions for a given user. Next, the obtained outputs are analyzed in order to identify latent dependencies between model characteristics recommendation quality, in

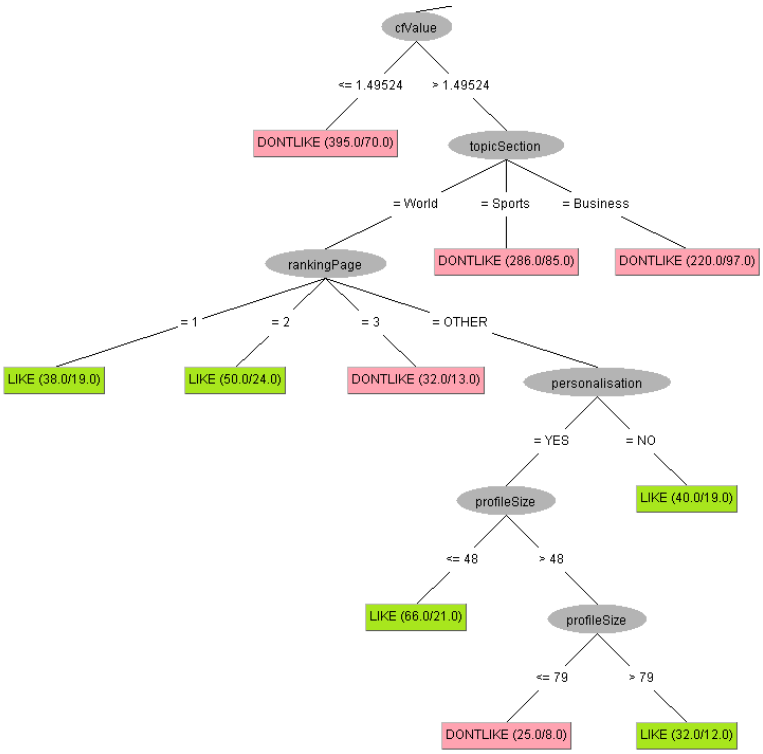

Fig. 10. Branch with CF value less than 2.76, using a user-based algorithm

order to single out the most relevant ones. Finally, adjusting the identified characteristics, a new recommendation model is produced, aiming to generate more accurate results. The work herein presented focuses on the identification of such relevant model characteristics.

The proposed approach applies ML techniques to learn the user and system features that favor correct recommendations by the system. Specifically, for every recommendation evaluated (rated) by the user a training sample is created. The attributes of the sample are the target characteristics for analysis, and their values are taken from log information databases. The training example is assigned one of two possible classes, correct or incorrect, depending on whether the user evaluated the corresponding recommendation as relevant or irrelevant. The ML strategy consists of a classification algorithm on these examples.

The presented approach has been tested in the News@hand recommender system [8]. Further work is needed to measure the performance improvements obtained in that system after applying the proposed strategy, as well as to investigate other machine learning techniques, apart from decision trees, in order to select the most relevant model features.

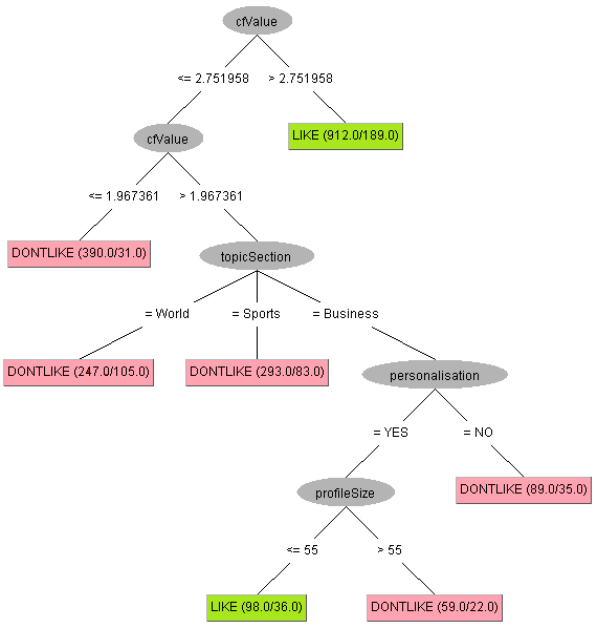

Fig. 11. Whole tree generated by using an item-based algorithm

Besides assessing the potential direct benefits on recommendation performance, further findings were drawn from the empiric experience with regards to the experimental methodology itself, identifying shortcomings and weaknesses. We found out that the first stage of our evaluation was unbalanced in terms of the difficulty to obtain news items relevant for each task. The decision tree in Figure 12 is such an example. The classifier infers that most of the users liked (almost) every news item in a particular section, while in other sections this is conditional on other parameters such as the ranking page or other model characteristics. The task related to the Science section was identified as 'very easy' by the users, probably because the query used in this task biased the results to be relevant to the goal. We also observed that some tasks performed better when contextualization was activated. This could be caused by the fact that a particular goal was very specific, and there was no profile focused on that domain (in our case, *Business* section). A similar situation was the one in which the profiles had to be very specific to get some results. Since the users were not finding relevant news items, the context was useless (this happened in *Entertainment* section). Finally, another important conclusion concerns the manual profiles. When users create theirs profiles, they do not know anything about which will be their goals or queries, which makes very difficult for personalization algorithms to rank relevant news in the first pages.

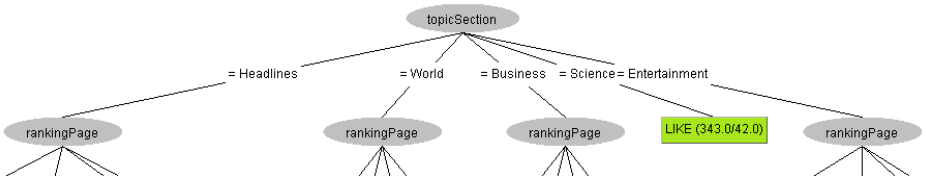

Fig. 12. Fragment of the decision tree with unbalance node load.

Acknowledgments This research has been supported by the Spanish Ministry of Science and Education (TIN2007-64718 and TIN2008-06566-C04-02).

References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A. (2005). Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Transactions on Information Systems*, 23(1), pp. 103-145.
2. Adomavicius, G, Tuzhilin, A. (2005). Toward 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), pp. 734-749.
3. Becker, K., Marquardt, C. G., Ruiz, D. D. (2004). A Pre-Processing Tool for Web Usage Mining in the Distance Education Domain. In: *Proceedings of the 8th International Database Engineering and Applications Symposium (IDEAS 2004)*, pp. 78-87.
4. Bellogín, A., Cantador, I., Castells, P., Ortigosa, A. Discovering Relevant Preferences in a Personalised Recommender System using Machine Learning Techniques. In: *Proceedings of the ECML/PKDD-08 Workshop on Preference Learning (PL 2008)*, pp. 82-96.
5. Brusilovsky, P. (2003). Developing adaptive educational hypermedia systems: From design models to authoring tools. In: *Authoring Tools for Advanced Technology Learning Environment*, pp. 377-409.
6. Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4), pp. 331-370.
7. Cantador, I., Bellogín, A., Castells, P. (2008). A Multilayer Ontology-based Hybrid Recommendation Model. *AI Communications*, 21(2-3), pp. 203-210.
8. Cantador, I., Bellogín, A., Castells, P. (2008). News@hand: A Semantic Web Approach to Recommending News. In: *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2008)*, pp. 279-283.
9. Cantador, I., Bellogín, A., Castells, P. (2008). Ontology-based Personalised and Context-aware Recommendations of News Items. In: *Proceedings of the 2008 Web Intelligence Conference*, pp. 562-565.
10. Cantador, I., Castells, P. (2008). Extracting Multilayered Semantic Communities of Interest from Ontology-based User Profiles: Application to Group Modelling and Hybrid Recommendations. *Computers in Human Behavior*. Elsevier.
11. Cantador, I., Szomszor, M., Alani, H., Fernández, M., Castells, P. (2008). Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations. In: *Proceedings of the 1st Intl. Workshop on Collective Intelligence and the Semantic Web (CISWeb 2008)*, pp. 5-19.

12. Castells, P., Fernández, M., and Vallet, D. (2007). An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. *IEEE Transactions on Knowledge and Data Engineering* 19(2), pp. 261-272.
13. Cohen, P. R., & Kjeldsen, R. (1987). Information Retrieval by Constrained Spreading Activation in Semantic Networks. *Information Processing and Management*, 23(4), pp. 255-268.
14. Crestani, F. (1997). Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review*, 11(6), pp. 453-482.
15. Crestani, F., & Lee, P. L. (2000). Searching the Web by Constrained Spreading Activation. *Information Processing and Management*, 36 (4), pp. 585-605.
16. Duda, R. O., Hart, P. E., Stork, D. G. (2000). *Pattern Classification*. Wiley-InterScience.
17. Rafter, R. and Smyth, B. (2005). Conversational Collaborative Recommendation: An Experimental Analysis. *Artificial Intelligence Review*, 24(3-4), pp. 301-318
18. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering on netnews. In: *Proceedings of the 1994 Conference on. Computer Supported Collaborative Work*, pp. 175–186.
19. Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001) Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 2001 WWW Conference*, pp. 285–295.
20. Srivastava J., Cooley R., Deshpande M., Tan P. (2000). Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, 1(2), pp.12-23.
21. Talavera, L., Gaudioso, E. (2004). Mining Student Data to Characterize Similar Behavior Groups in Unstructured Collaboration Spaces. In: *Proceedings Workshop on AI in CSCL*, pp. 17-23.
22. Terveen, L., & Hill, W. (2001). Beyond Recommender Systems: Helping People Help Each Other. In: *Human-Computer Interaction in the New Millennium*, pp. 487-509.
23. Vallet, D., Castells, P., Fernández, M., Mylonas, P., Avrithis, Y. (2007). Personalised Content Retrieval in Context Using Ontological Knowledge. *IEEE TCSVT* 17(3), pp. 336-346.
24. Vialardi, C., Bravo, J., Ortigosa, A. (2007). Empowering AEH Authors Using Data Mining Techniques. In: *Proceedings of the 5th Int. Workshop on Authoring of Adaptive and Adaptable Hypermedia*.
25. Vialardi, C., Bravo J., Ortigosa, A. (2008) Improving AEH Courses through Log Analysis. *Journal of Universal Computer Science* 14(17), pp. 2777-2798.
26. Witten, I. H., Frank, E. (2005).*Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems.
27. Zaïane, O. R. (2006). Recommender System for E-learning: Towards Non-Instructive Web Mining. In C. Romero and S. Ventura (eds.) *Data Mining in E-Learning*, pp.79-96.
28. Zhang, T., Iyengar, V. S. (2002). Recommender Systems Using Linear Classifiers. *Journal of Machine Learning Research*, 2, pp. 313-334.