Discovering Relevant Preferences in a Personalised Recommender System using Machine Learning Techniques

Alejandro Bellogín, Iván Cantador, Pablo Castells, Álvaro Ortigosa

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

Abstract. Personalised recommender systems learn about a user's needs, and identify and suggest information items (news articles, images, videos, etc.) that meet those needs. User needs can be explicitly or implicitly defined either in the form of user tastes, interests and goals, or by system parameters and configurations. Most research efforts in the Recommender Systems field can be said to have been directed towards either defining and improving techniques that provide item recommendations from available preference data, or defining techniques for learning the latter. However, little research has focussed on learning which preferences are really relevant to provide accurate recommendations, and which ones imply anomalous behaviour of the recommendation mechanisms. We present a meta-evaluation methodology that applies Machine Learning techniques to analyse log information of a personalised news recommender system in order to discover (and rank) which user preferences and system settings are suitable for accurate recommendations. We also show how the proposed methodology can be used to ease the system evaluation itself.

Keywords: evaluation, preference learning, recommender systems, decision trees.

1 Introduction

Personalised recommender systems provide advice to users about products or services s/he might be interested in. The generated suggestions are obtained from the consideration of tastes, interests and goals of the user, and usually depend on multiple system parameters and configurations.

In general, a recommender system compares the user's profile or usage data to some reference characteristics. These characteristics might belong to the information items (in the content-based approach), or to the user's social environment (in the collaborative filtering approach). Combinations of both approaches have been also investigated in the so-called hybrid recommender systems [4].

In this scenario, contextual sources of information can also be exploited to improve recommendations [1]. For example, recent and current queries, ratings, evaluations, etc. from the user might be taken into account at the time when context-aware personalised recommendations are produced.

As more user and system characteristics are taken into account in the recommendation mechanisms, it usually becomes increasingly difficult to understand the reasons and circumstances under which the given recommendations match the user profile.

Research efforts to date can be said to have mainly focussed on the study of the improvement of the recommendation models by using all the available knowledge and user profiling information. However, few studies have addressed the issue of finding out which item features, user interests and system settings are the most significant when accurate and non-accurate recommendations are generated. If those characteristics were identified, recommendation strategies might be enhanced by strengthening or discarding their dependencies with specific stereotypes of user profiles and information items.

In this paper, we propose the use of Machine Learning (ML) techniques to analyse and learn which user and system features/settings in a personalised recommender system are more adequate for correct recommendations. In the proposed approach, for every recommendation evaluated (rated) by a user we create a pattern. The attributes of the pattern correspond to the characteristics we aim to analyse, and their values are obtained from log information databases. The class of the pattern can be assigned two possible values, correct or incorrect, depending on whether the user evaluated the recommendation as relevant or irrelevant. By classifying these patterns, ML algorithms facilitate the analysis of the above preferences.

We have tested this proposal with the personalised content-based and context-aware recommendation models of News@hand, a recommender system that makes use of semantic-based technologies to suggest news articles. As described herein, Decision Trees and Attribute Selection are used to identify the preferences for which the suggestions of each recommendation model should be exploited.

The rest of the paper is organised as follows. Section 2 gives an overview of related works in which ML techniques have been applied to automatically learn preferences in personalised content retrieval, recommender and adaptive systems. Section 3 introduces News@hand, the news recommender system that has been used to evaluate our preference analysis proposal. Section 4 briefly explains Decision Trees and Attribute Selection, the ML techniques used in our proposal, and describes which attributes have been chosen for the analysed patterns. Section 5 includes preliminary experiments conducted to evaluate the approach. Finally, section 6 concludes with some discussions and future research lines.

2 Related Work

ML techniques are useful when huge amounts of data have to be classified and analysed, which nowadays is a very common situation in many scenarios, such as web information exploitation [10]. They have also proved to be of use in adaptive e-learning environments, where student data is used to adapt the system to the user preferences and capabilities in order to facilitate her learning process. Hence, for example, in Becker and Marquardt's work [3], students' logs are analysed to find patterns that reveal the system browsing paths followed by students. They used this information to improve the student experience. Talavera and Gaudioso [11] use classification techniques to analyse student behaviour in a cooperative learning environment. Their main goal is to discover patterns that reflect the students' behaviour, supporting tutoring activities on virtual learning communities. In this area, Zaïane's proposal [15] is one of the first approaches that used association rules. This work consists of a recommender agent able to suggest online activities based on usage data from previous participants of a course. The objective is to enhance the browsing of the course material for individual users.

Many authors have also investigated the application of these techniques to Recommender Systems [16]. 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 [12], which is similar to the idea of using short and long term profiles in [9].

However, 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. 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 [13], where ML techniques find patterns for assisting adaptive hypermedia authors during the design and evaluation phases. The authors build a model representing the student behaviour on a particular course, and use it to obtain and exploit a vision of the behaviour and performance of student groups.

3 News@hand

News@hand (Figure 1) 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 the news contents and user preferences. 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 *soccer* and *match*, or specific instances of football teams and players (e.g., *Real Madrid*, *Zidane*). Semantic relations among those concepts are exploited to enrich the above representations, and are incorporated within the recommendation processes.

The 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, she can browse any of the previous sections, but the news items are listed without any personalisation criterion. She can only sort them by their publication date, source or level of popularity (i.e., according to a classic rating-based collaborative filtering 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 her and others' semantic preferences in different ways. Short and long term preferences are considered. Click history is used to define the short-term concepts, and the resultant ranks can be adapted to the current context of interest.

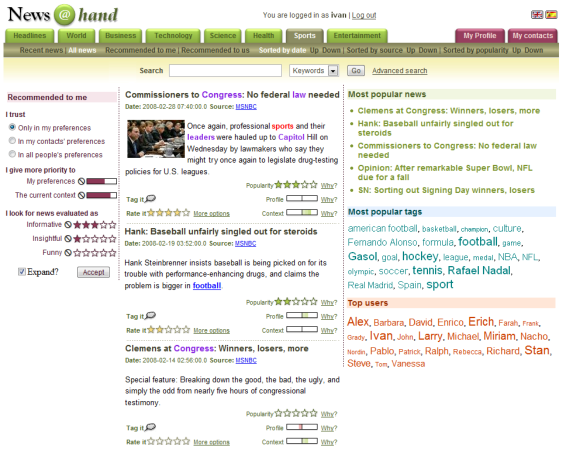

Fig. 1. An example of a news recommendation page in News@hand

3.1 Architecture

Figure 2 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.

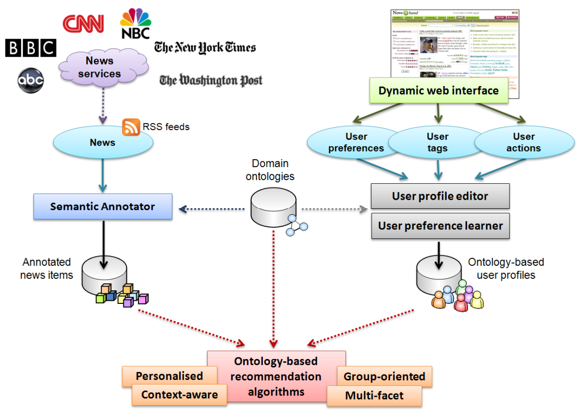

Fig. 2. Architecture of News@hand

News@hand follows a client/server architecture, where users utilise a web interface to receive on-line news recommendations, and update their preferences. A dynamic graphical interface allows the system to automatically store all the users' inputs, analyse their behaviour with the system, and adjust the news recommendations in real time. Explicit and implicit user preferences 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 are executed. As explained below, News@hand offers personalised, context-aware, group-oriented [6] and multi-facet recommendations [5].

3.2 Recommendation models

In the knowledge representation we propose, user preferences are described as vectors $\mathbf{u}_m = (u_{m,1}, u_{m,2}, ..., u_{m,K})$ 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 $\mathbf{d_n} = (d_{n,1}, d_{n,2}, ..., d_{n,K})$ of concept weights, in the same vector-space as user preferences.

With this knowledge representation, the existence of sparsity in user profiles is evident. Users are usually not willing to spend time describing their preferences to the system, even less to assign weights to them. On the other hand, automatic learning algorithm might tend to recognise the main characteristics of user preferences. To overcome this problem, we propose a preference expansion mechanism, which spreads the initial set of preferences stored in user profiles through explicit semantic relations with other concepts in the ontology. Our approach [5, 6] is based on Constrained Spreading Activation (CSA), and is self-controlled by applying a decay factor to the intensity of preference each time a relation is traversed.

3.2.1 Personalised recommendations

Our notion of personalised content retrieval is based on a matching algorithms that provide a relevance measure $pref\left(u_m, d_n\right)$ 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 $\cos(\mathbf{u}_m, \mathbf{d}_n)$, and can be combined with query-based scores without personalisation and semantic context information, to produce combined rankings.

3.2.2 Context-aware recommendations

We propose a particular notion 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 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. Once the context is built, a contextual activation of preferences is achieved by finding semantic paths linking preferences to context (see Figure 3). The perceived effect of contextualisation is that user interests that are out of focus, under a given context, are disregarded, and only those that are in the semantic scope of the ongoing user activity are considered for recommendation.

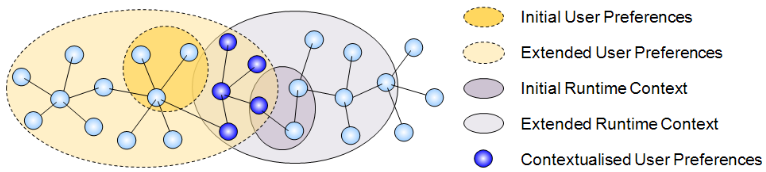

Fig. 3. Semantic contextualization of user preferences. Nodes represent ontology concepts and edges are associated to semantic relations between those concepts.

3.3 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. 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.

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

Database tables share a session identifier that allows us to recognise 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 personalisation is off), and context weight.

4 Decision Trees and Attribute Selection for Preference Learning

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

In this section, we describe the attributes selected for the patterns, and the ML algorithms applied for our preference learning purposes. The attributes are divided in two classes: the ones describing a user preference and the ones describing a system setting. Information for creating the patterns is obtained from the log database introduced in section 3.3.

4.1 Decision Trees

Decision Trees apply a divide-and-conquer strategy for producing classifiers with the following benefits [8]:

- 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 the 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 do not generalise 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 use its well-known revision J4.8.

4.2 Attribute Selection

Most ML algorithms are designed to learn the most appropriate attributes to exploit for making their decisions [14]. For instance, Decision Tree methods choose the most promising attribute to split on at each point and should never select irrelevant attributes.

Because of the negative effect of irrelevant attributes on most ML schemes, it is common to precede learning with an Attribute Selection stage that strives to eliminate all but the most relevant attributes. Reducing the dimensionality of the data by deleting unsuitable attributes usually improves the performance of learning algorithms.

Methods for attribute selection involve searching the space of attributes for the subset that is more likely to predict the class best. In our experiments, in order to obtain a ranking between attributes we use a single-attribute evaluator that measures their information gain with respect to the class. This strategy, together with the attributes and classes selected for the patterns to analyse, are explained in the next section.

5 Experiments

The experiments have been conducted using News@hand system described in section 3. In this section, a description of the news item database and the knowledge repository is provided. We also explain the tasks and phases fulfilled by users during the evaluation, and conclude with the experiments 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 3.1, and a total of 66,378 annotations were obtained. For more details, see [7].

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 [7]. A total of 137,254 Wikipedia entries were used to populate 744 ontology classes with 121,135 instances.

5.2 Experimental setup

We present an experiment conducted to evaluate the precision of the personalisation and the context-aware recommendation functionalities available in News@hand (sections 3.1 and 3.2). With this experiment 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 personalisation module was active, and the tasks were different in having the semantic expansion (see section 3.2) enabled or disabled. In the second phase, the contextualisation and semantic expansion functionalities were active. In the second task, the personalised recommendations were also enabled. More details are given in the next subsection.

In the experiment, 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. We considered three domains: telecommunications, banking and social care issues. For each domain a user profile and two search goals were defined as explained below. Table 2 shows a summary of the involved tasks.

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

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

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

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, $Q_{1,2}$, was reduced to evaluate those articles existing in Entertainment section that were retrieved with the query “music”.

Table 3 shows the tasks performed by the 16 users. 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 personalised and context-aware recommendations was established as a linear combination of their results using two weights $w_p, w_c \in [0,1]$.

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

In the personalisation phase, the contextualisation 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 contextualisation phase, w_c was set to 1 and the expansion was enabled. Its first tasks were done without personalisation (w_p=0), and its second tasks were a bit influenced by the corresponding profiles (w_p=0.5).

As mentioned before, a fixed user profile was used for each domain. Some of them were predefined profiles, and some of them where created by the users (those marked with ‘*’ in the table) 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. They are marked with an ‘A’ in Table 3.

Table 3. Experiment tasks configurations

	Personalised recommendations		**Context-aware recommendations**	
	Without expansion	With expansion	With expansion	
User	w_p=1 w_c=0	w_p=1 w_c=0	w_p=0 w_c=1	w_p=0.5 w_c=1
1	$*Q_{1,1}$	$Q_{2,1}$	$Q_{3,1}$	$^{A}Q_{1,2}$
2	$Q_{2,2}$	$*Q_{3,2}$	$^{A}Q_{2,1}$	$Q_{1,2}$
3	$Q_{3,1}$	$^{A}Q_{3,2}$	$*Q_{1,1}$	$Q_{2,1}$
4	$^{A}Q_{1,1}$	$Q_{1,2}$	$Q_{2,2}$	$*Q_{3,2}$
5	$Q_{1,2}$	$*Q_{2,2}$	$Q_{3,2}$	$^{A}Q_{2,1}$
6	$Q_{2,1}$	$Q_{3,1}$	$*^{A}Q_{3,2}$	$Q_{1,1}$
7	$Q_{3,2}$	$^{A}Q_{1,1}$	$Q_{1,2}$	$*Q_{2,2}$
8	$*^{A}Q_{2,2}$	$Q_{1,1}$	$Q_{2,1}$	$Q_{3,1}$
9	$Q_{1,1}$	$Q_{2,1}$	$*Q_{3,1}$	$^{A}Q_{3,2}$
10	$Q_{2,2}$	$Q_{3,2}$	$^{A}Q_{1,1}$	$*Q_{1,2}$
11	$*Q_{3,1}$	$^{A}Q_{2,2}$	$Q_{1,1}$	$Q_{2,1}$
12	$^{A}Q_{3,1}$	$*Q_{1,2}$	$Q_{2,2}$	$Q_{3,2}$
13	$Q_{1,2}$	$Q_{2,2}$	$Q_{3,2}$	$*^{A}Q_{1,1}$
14	$*Q_{2,1}$	$Q_{3,1}$	$^{A}Q_{2,2}$	$Q_{1,1}$
15	$Q_{3,2}$	$*^{A}Q_{3,1}$	$Q_{1,2}$	$Q_{2,2}$
16	$^{A}Q_{1,2}$	$Q_{1,1}$	$*Q_{2,1}$	$Q_{3,1}$

There is also an important issue about how the users rated. Every time the user read an item, she had to think 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 3 stars if the item was relevant to the current goal.
- Rate with 4 stars if the item was relevant to the profile.
- Rate with 5 stars if the item was relevant to the current goal and the profile.

This rating constraint gave us a bounded frame for evaluation. In section 5.2.3, it will be shown that it also allowed us to have different criteria over the ratings when the classes of the patterns had to be set.

5.2.1 Personalisation phase

The objective of the two tasks performed in the first experiment phase was to assess 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 personalisation module deactivated.
- Rate the top 15 news items.
- Launch the query with the personalisation 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.

5.2.2 Contextualisation phase

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

- Launch the query with the contextualisation activated (semantic expansion enabled, and personalisation 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 personalisation on and 45 items with personalisation off. She had also evaluated as relevant 4 news item (adding them to the context).

5.2.3 Selection of pattern attributes and classes based on evaluation parameters

Each user has to assign a rating depending on the four existing possibilities for each news item: relevant to the profile (3), the goal (2), both (4), and neither of them (1). Considering these four options, there are three different criteria to classify an item (pattern) 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 pattern classes, according to the evaluation made, we selected those variable attributes whose impact on the results we wanted to analyse. For each item rating log entry, we chose several attributes that can be categorised as follows:

- **User preferences**
  - *Profile type*: a string attribute with two possible values: *fixed* or *used-defined* preferences (manual preferences).
  - *Profile size*: a string attribute with three possible values corresponding to the relative size of the user profile (i.e., number of semantic preferences): *small*, *medium* or *large* profile. If the personalisation is off, the value is *none*.
  - *Context size*: a string attribute with three possible values corresponding to the size of the context-aware preferences: *small*, *medium* or *large* context. Obviously, if the context is off, this attribute takes a different value: *none*.
- **System settings**
  - *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.
  - *Personalised recommendations*: a Boolean value indicating whether the personalised 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.

5.3 Results

This section presents the results obtained using ML techniques to analyse the evaluation previously described. These results are classified into three different categories, according to the consequences that can be drawn from them. First, we present the results related to the personalisation phase, where the impact of the semantic expansion is considered. Secondly, the contextualisation phase results are presented, where the importance of context combined with personalisation is studied. Finally, some conclusions about the evaluation itself are shown. For developing these results, Weka ML toolkit [14] was used.

5.3.1 Learning preferences from personalised recommendations

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

- **Profile size**. In Figures 4 and 5a it can be seen that this user preference is useful when retrieving relevant items, and is connected with expansion and activation of personalisation (*profileSize* is the third node in both figures).

- **Ranking page**. Fortunately, the system retrieves relevant news in the first page (top 5 news items). Because of that, our analysis focused on sub-trees where the ranking page has a value of 1.
- **Expansion**. The importance of this preference is shown in Figure 5b, where users find relevant news only when personalisation and expansion are activated.

In general, we have found that using personalisation in combination with semantic expansion improves the performance in the first page. Although not all the sections behave equally (see section 5.3.4), this is true for general sections such as Headlines, despite the fact that in the second and third pages, personalisation improves little and needs the help of other techniques, such as context (see next subsection).

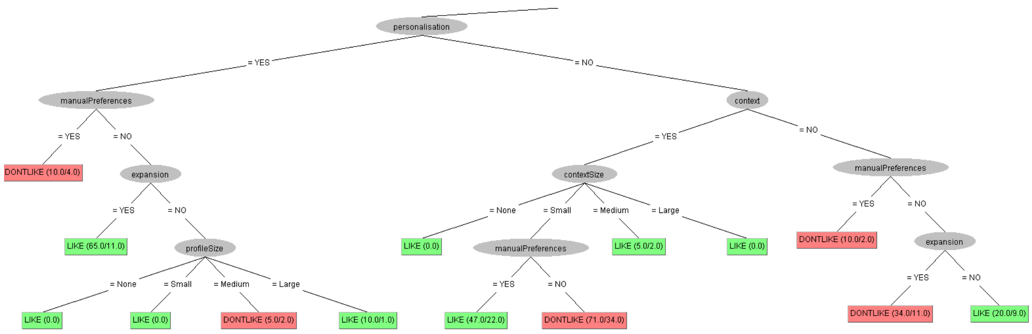

Fig. 4. Headlines branch in the first page and using all the available logs (general evaluation)

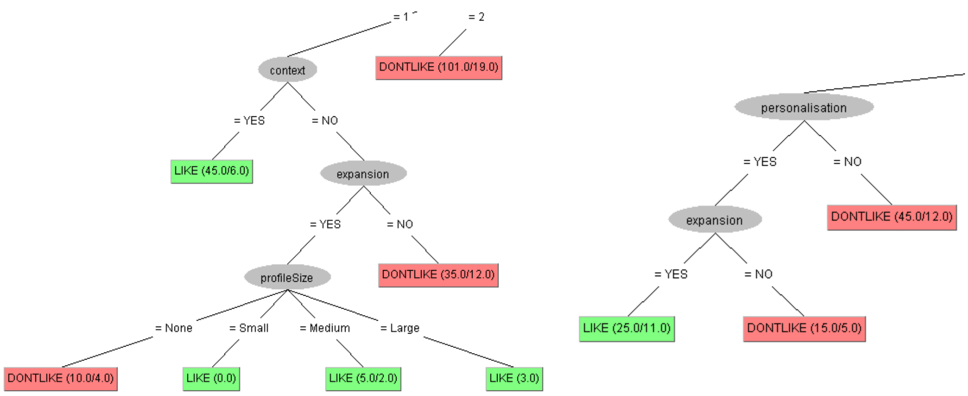

Fig. 5. (a) Business branch in the first page and using all the available logs (goal-based evaluation), (b) Headlines branch in the first page, no manual preferences, and using personalisation-related logs (goal-based evaluation)

5.3.2 Learning preferences from context-aware recommendations

In the experiments, we found some preferences more likely to help in context-aware recommendations. For instance, personalisation was a well-performing system setting when it is combined with context. Although sometimes context alone performs well (Figure 5a), in Figure 6 we show an example where context needs personalisation to obtain good results.

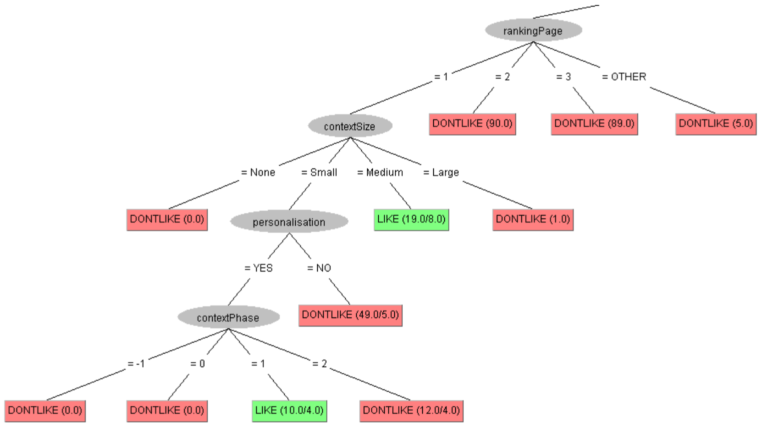

Fig. 6. World branch using context related logs (goal-based evaluation)

Another good indicator is the context size. Usually the best performance is achieved when context has a medium size (Figure 6). Something similar happens with the context phase, in which an intermediate value (phase 1) causes the best performing situation. A user preference not having influence in context is the fact of using manual preferences or not, since the context has more to do with the short term preferences, rather than long term preferences represented by the user profile.

5.3.3 Ranking preferences

Leaving apart our goal of obtaining good recommendations can lead to simplify the system in such a way that less parameters are involved and, because of this, a better performance could be achieved. In order to do this, we use the attribute selection technique explained in section 4.2. Table 4 shows attribute rankings according to the amount of information each attribute gains with respect to the class. They are shown in decreasing order for different subsets of log entries: logs associated to personalisation, to context-aware recommendations, and all of them.

Table 4. Attribute selection for the goal-based evaluation. The underlined attributes are the one characteristic to that column.

Using personalisation logs	Using context logs	Using all logs
Topic section	*Topic section*	*Topic section*
<u>Expansion</u>	*Ranking page*	*Ranking page*
Ranking page	*Profile size*	*Profile size*
<u>Manual preferences</u>	*Personalisation*	*Personalisation*
Profile size	*<u>Context size</u>*	*<u>Expansion</u>*
Personalisation	*<u>Context phase</u>*	*Context size*
Context	*Context*	*Context phase*
Context phase	*Manual preferences*	*Context*
Context size	*Expansion*	*Manual preferences*

It is important to notice that expansion is informative in the general case, and even more in the personalisation situation. Manual preferences are informative only when we are dealing with personalisation, on the contrary to what happens with context size and phase, which are more informative in the contextualisation phase.

5.3.4 Meta-evaluation results

Using ML techniques to learn user and system characteristics we have discovered several anomalies in the evaluation. Some of them are situations to avoid when designing an evaluation (e.g., easy tasks that provide evident results and conclusions, resources are limited, etc.), but others are inherent to any evaluation plan, and sometimes convenient.

In the first group, we found that our evaluation was unbalanced in terms of difficulty of obtaining relevant news items for each task. A clear sign of this is a decision tree such as the one presented in Figure 7, where there are sections being expanded, and others in which the classifier assigns a great percentage of the instances to a particular class with a high confidence. In this example, the task related to *Science* section was found very 'easy' for users, probably because the query used in this task biased the results to be relevant to the goal.

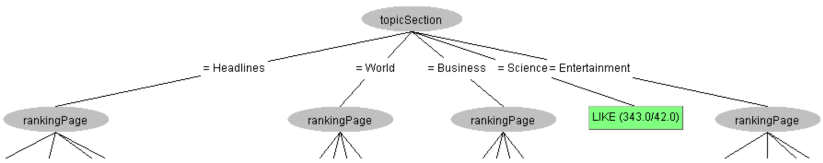

Fig. 7. Split showing unbalanced anomaly

In the second group, we found more interesting situations. The first one was that some tasks performed better when context was used. This might be caused to 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 profile should be very specific to get some result, and since the user never finds a relevant news item, the context is useless (this is what happens with *Entertainment* section). Another important conclusion concerns the *manual profiles*. When the user creates her profile, she knows nothing about which will be her goal or query, and this makes very difficult for personalisation algorithms to rank relevant news in the first pages.

6 Conclusions and Future Work

We have presented a general approach that applies ML techniques to learn those user and system features of a recommender system that are (un)favourable for correct recommendations. The approach has also seemed useful to learn deficiencies and weaknesses of the experiments conducted to measure the system performance.

For every recommendation evaluated (rated) by a user we create a pattern. The attributes of the pattern correspond to the characteristics we want to analyse, and their values are obtained from log information databases. The class of the pattern can be assigned two possible values, correct or incorrect, depending on whether the user evaluated the recommendation as relevant or irrelevant. Classifying the created patterns, Machine Learning strategies facilitate the analysis of the inner preferences.

Specifically, we have tested our log analysis proposal using the personalised content-based and context-aware recommendation models of News@hand, a recommender system that makes use of Semantic Web technologies to suggest news articles. Decision Trees and Attribute Selection have allowed us to infer those preferences for which the suggestions of each recommendation model should be exploited, and rank them according to their informative value.

The next step is to make the system adaptive to the current status of the analysed preferences, and evaluate whether the recommendations improve with the changes. We want to conduct similar evaluations with the collaborative group-oriented and multi-facet recommendation strategies of News@hand. We expect this process will be more difficult because profile similarities among several users will have to be taken into consideration.

Acknowledgments. This research has been supported by the Spanish Ministry of Science and Education (TIN2005–6885 and TIN2007-64718).

References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A. (2005). *Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach*. In: 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*. In: 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. Burke, R. (2002). *Hybrid Recommender Systems: Survey and Experiments*. In: User Modeling and User-Adapted Interaction, 12(4), pp. 331-370.
5. Cantador, I., Bellogín, A., Castells, P. (2008). *A Multilayer Ontology-based Hybrid Recommendation Model*. In: AI Communications, special issue on Rec. Systems. In press.
6. Cantador, I., Castells, P. (2008). *Extracting Multilayered Semantic Communities of Interest from Ontology-based User Profiles: Application to Group Modelling and Hybrid Recommendations*. In: Computers in Human Behavior, special issue on Advances of Knowledge Management and the Semantic Web for Social Networks. Elsevier. In press.
7. Cantador, I., Szomszor, M., Alani, H., Fernández, M., Castells, P. (2008). *Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations*. In: 1st Intl. Workshop on Collective Intelligence and the Semantic Web (CISWeb 2008), pp. 5-19.
8. Duda, R. O., Hart, P. E., Stork, D. G. (2000). *Pattern Classification*. Wiley-InterScience.
9. Rafter, R. and Smyth, B. (2005). *Conversational Collaborative Recommendation: An Experimental Analysis*. In: Artificial Intelligence Review, 24(3-4), pp. 301-318
10. Srivastava J., Cooley R., Deshpande M., Tan P. (2000). *Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data*. In: SIGKDD Explorations, 1(2), pp.12-23.
11. Talavera, L., Gaudioso, E. (2004). *Mining Student Data to Characterize Similar Behavior Groups in Unstructured Collaboration Spaces*. In: Workshop on AI in CSCL, pp. 17-23.
12. Vallet, D., Castells, P., Fernández, M., Mylonas, P., Avrithis, Y. (2007). *Personalized Content Retrieval in Context Using Ontological Knowledge*. In: IEEE TCSVT 17(3), pp. 336-346.
13. Vialardi, C., Bravo, J., Ortigosa, A. (2007). *Empowering AEH Authors Using Data Mining Techniques*. In: Proceedings of 5th International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H 2007).
14. Witten, I. H., Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems.
15. Zaïane, O. R. (2006). *Recommender System for E-learning: Towards Non-Instructive Web Mining*. In: Data Mining in E-Learning, pp.79-96.
16. Zhang, T., Iyengar, V. S. (2002). *Recommender Systems Using Linear Classifiers*. In: The Journal of Machine Learning Research, 2, pp. 313-334.