Computers in Human Behavior 30 (2014) 663–673

Contents lists available at ScienceDirect

Computers in Human Behavior

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

Elicitation of latent learning needs through learning goals recommendation

Nicola Capuano [a], Matteo Gaeta [b], Pierluigi Ritrovato [b,*], Saverio Salerno [b]

[a] Centro di Ricerca in Matematica Pura ed Applicata, Fisciano SA, Italy
[b] University of Salerno, Fisciano SA, Italy

ARTICLE INFO

Article history:
Available online 1 October 2013

Keywords:
Adaptive learning
Recommender systems
Intelligent tutoring systems

ABSTRACT

The aim of a recommender system is to estimate the relevance of a set of objects belonging to a given domain, starting from the information available about users and objects. Adaptive e-learning systems are able to automatically generate personalized learning experiences starting from a learner profile and a set of target learning goals. Starting form research results of these fields we defined a methodology and developed a software prototype able to recommend learning goals and to generate learning experiences for learners using an adaptive e-learning system. The prototype has been integrated within IWT: an existing commercial solution for personalized e-learning and experimented in a graduate computer science course.

1. Introduction

A significant educational action able to guide the learner in a comprehensive learning process is not only focused on learning (cognition level) but also on fostering a correct learning behavior that empowers learners to achieve their learning goals in a controlled and directed way (metacognition level) (Mangione, Gaeta, Orciuoli, & Salerno, 2010).

Starting from this principle we defined and developed an e-learning system able to build personalized learning experiences starting from a set of target concepts selected on an ontology-based domain model (Capuano, Gaeta, Miranda, Orciuoli, & Ritrovato, 2008). We then extended such system in order to allow course generation from an explicit request in terms of needs to be satisfied and expressed by the learner in natural language (CapuaCapuano, Gaeta, Orciuoli, & Ritrovato, 2009).

The work presented in this paper deals with the definition of a further process of course building starting from an implicit request rather than from an explicit one. In other words, a methodology to recommend learning goals based on the analysis of a learner' profile (including known topics) and on the comparison of this profile with profiles of similar learners is defined.

The proposed methodology upholds the social presence (Acampora, Gaeta, Orciuoli, & Ritrovato, 2010; Capuano, Gaeta, Orciuoli, & Ritrovato, 2010), while supporting the development of self-regulated learning. Educational recommendations serves as a pedagogical advance organizer for the learners, as it anticipates and spreads needs, knowledge and learning paths. Furthermore the proposed solution also supports help seeking processes improving the students' control over learning. This makes the solution adequate nononly for educational settings but also for enterprise training (Capuano, Gaeta, Ritrovato, & Salerno, 2008).

The paper is organized in this way: Section 2 introduces some background about recommender systems; Section 3 briefly introduces the starting point of our research and then describes the proposed methodology; Section 4 introduces the developed prototype and presents some example of use; Section 5 compares our approach with some existing recommender systems for e-learning; eventually Section 6 presents conclusions and planned future work.

2. Background on recommender systems

Recommender Systems (RS) are aimed at providing personalized recommendations on the utility of a set of objects belonging to a given domain, starting from the information available about users and objects.

A formal definition of the recommendation problem can be expressed in these terms (Adomavicius & Tuzhilin, 2005): C is the set of users of the system, I the set of objects that can be recommended, R a totally ordered set whose values represent the utility of an object for a user (e.g. integers between 1 and 5 or real numbers between 0 and 1) and u: $C \times I \rightarrow R$ a utility function that measures how a given object $i \in I$ is useful for a particular user $c \in C$. The purpose of the system is to recommend to each user c the object i'_c that maximizes the utility function so that:

* Correspoding author.
E-mail addresses: capuano@crmpa.unisa.it (N. Capuano), mgaeta@unisa.it (M. Gaeta), pritrovato@unisa.it (P. Ritrovato), salerno@unisa.it (S. Salerno).

0747-5632/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.chb.2013.07.036

$$i'_c = \arg\max_{i \in I} u(c, i) \quad (1)$$

The central problem of the recommendations is that the function u is not completely defined on the space $C \times I$. In fact, in typical applications of such systems, a user never expresses p on each object of the available catalogue. A RS shall then be able to estimate the values of the utility function also in the space of data where it is not defined, extrapolating from the points of $C \times I$ where it is known.

In other words, the goal is to make a prediction about the vote that a particular user would give to an object that has not been rated yet. Several techniques for RS exist in literature, they are usually divided in three broad categories:

- *cognitive (or content-based) approaches:* specific objects are recommended to the user, similar to those that have been positively rated in the past (they are therefore based on the calculation of similarity between objects);
- *collaborative approaches:* specific objects are recommended to the user, in particular those objects that are liked by other people with similar tastes (they are therefore based on the calculation of similarity between users);
- *hybrid systems*: they combine the two previous approaches.

In following sub-sections, we present the three approaches by considering the advantages and disadvantages of each of them. As described in Section 3, our methodology applies a hybrid approach, combining cognitive and collaborative elements.

2.1. Cognitive approaches

In cognitive approaches (Balabanovic & Shoham, 1997), the value of the utility function $u(c, i)$ of the user c for the object i is predicted by considering the values $u(c, i_k)$ to be assigned to items found similar to c. In general, each object $i \in I$ is associated with a profile, i.e. a set of attributes able to characterize the content, that is represented by a vector $content(i) = (w_{i,1}, \ldots w_{i,k})$ where $w_{i,j}$ is the weight of the j-th attribute or an indication of how the j-th attribute is able to characterize the object i. The attributes' weight can be created automatically by the system or manually by a user.

As for the objects, users are also associated with a profile based on the attributes of the objects preferred in the past. The profile is defined as $profile(c) = (w_{c,1}, \ldots w_{c,k})$, where each weight $w_{c,j}$ denotes the importance of the j-th attribute for the user c. The profile of user c can be obtained, in the simplest formulation, averaging all profiles of the objects for which c has expressed a rating and weighting them on the basis of the rating itself. Obviously, the profile varies over the time depending on the assessments that the user gradually provides.

Once the profiles that characterize objects and users have been defined, the utility of an object i for user c is calculated basing on the similarity between the two profiles. In other words $u(c, i) = sim(profile(c), content(i))$. Several similarity measures can be used for this purpose: one of the most common is the so-called cosine similarity based on the calculation of the cosine between two vectors using the following formula:

$$Sim(profile(c), content(i)) = \frac{\sum_{j=1}^{k} W_{c,j} W_{i,j}}{\sqrt{\sum_{j=1}^{k} W_{c,j}^2}\sqrt{\sum_{j=1}^{k} W_{i,j}^2}} \quad (2)$$

The main advantage of cognitive approaches is that the recommendations are only based on data related to the domain objects: first useful recommendations are then made immediately, with only one assessment made by the user. This feature is important in environments where it is necessary to produce immediate results or in which new users are added frequently. On the other hand this approach tends to over-specialize predictions, therefore making them uninteresting.

2.2. Collaborative approaches

In collaborative approaches, unknown values of the utility function $u(c, i)$ are estimated from those made available by people considered similar to c (Konstan et al., 1997). The basic idea is that users who evaluated in the same way the same objects are likely to have the same tastes (and are therefore similar).

Collaborative systems are very popular and are classified in categories depending on the algorithm used to explore the connections between users. Among the others, user-to-user memory-based algorithms (Perugini, Goncalves, & Fox, 2004) calculate the utility $u(c, i)$ as aggregation of the utility expressed for i by users similar to c; in other words:

$$u(c, i) = aggr_{c' \in C'} u(c', i) \quad (3)$$

where C' is the set of n users considered most similar to c (with n chosen between 1 and the total number of system users).

The simplest aggregation function is the average of ratings given to the users of C' or, as expressed below, the average of such ratings weighted on the degree of similarity between users who have expressed them:

$$u(c, i) = \frac{\sum_{c' \in C'} u(c', i) \cdot sim(c, c')}{\sum_{C' \in C'} |sim(c, c')|} \quad (4)$$

where $sim(c, c')$ indicates the degree of similarity between users c and c' calculated using similarity measures such as the cosine similarity (2) or the Pearson's correlation coefficient (Adomavicius & Tuzhilin, 2005). These measures are applied to vectors $(w_{c,1}, \ldots, w_{c,m})$ that characterize users, where $w_{c,i} = u(c,i)$, if defined.

By computing recommendations basing on the similarity between users, the advantage is to provide more accurate and less obvious advice. Conversely, the main problem occurs in domains with a large number of objects and/or users. Preferences in such environments are extremely sparse and the utility function is defined on a tiny part of the space $C \times I$. In these scenarios, it is difficult to calculate the correlation between users, so the recommendations are generated in an inaccurate way.

Directly linked to this limit, there is the commonly called *cold start problem*, that occurs in the early days of life of a system, when the available number of assessments is still lower than those of a fully operational system.

2.3. Hybrid approaches

Hybrid approaches try to overcome problems of both cognitive and collaborative approaches by using the two techniques simultaneously. There are several methods by which collaborative and cognitive approaches may be combined into a single system. Among them we quote the following (Burke, 2007):

- *weighted hybridization* (a cognitive and a collaborative algorithms are developed and, as final result, a combination of predictions from the two is used);
- *switching* (it is like the previous one but the system chooses, as appropriate, only one algorithm among those developed and it only returns results from it);
- *cascade hybridization* (available algorithms are ranked in order of priority and lower-level ones can only refine the results calculated from higher-level ones);
- *ad hoc algorithms* (they are specific implementations that combine cognitive and collaborative elements).