

Toward a Hybrid Recommender System for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval

Mohamed JEMNI, Mohamed Koutheaïr KHRIBI
Ecole Supérieure des Sciences et Techniques de Tunis
5, Av. Taha Hussein, B.P. 56, Bab Mnara 1080, Tunis, TUNISIE
mk.khribi@uvt.rnu.tn , mohamed.jemni@fst.rn.tn

Olfa NASRAOUI
Speed School of Engineering and Computer Science
University of Louisville KY 40292 USA
o0nasr01@louisville.edu

Abstract: The last decade has witnessed a great interest in e-learning and Web based education areas. Unfortunately, most of the e-learning environments used in the educational field today are still delivering the same educational resources and services in the same way to different learners. Hence, observing the increasing need for personalization in e-learning systems, we aim to make these systems deliver the most appropriate content to learners according to their interests and needs. This paper outlines the use of on-line automatic recommendations in e-learning systems based on learners' access history. First we start by mining learner profiles using usage Web mining techniques and content-based profiles using information retrieval techniques. Then, we use these profiles to compute relevant links to recommend for an active learner by applying a number of recommendation strategies.

Introduction

The use of e-learning platforms in education has been recently spreading out widely in schools and universities. The most known and used e-learning environments, such as, Moodle¹ and WebCT², are involving a large set of tools and functionalities to use in online courses, like synchronous and asynchronous communication tools, evaluation modules, course content delivery tools, collaborative activities, etc. However, the task of teaching online learners having different levels of knowledge, interest and need at a massive scale, seems to be very hard and time consuming for teachers. Moreover, most educational resources are conceived and designed while respecting similar well defined pedagogic models and navigation patterns. Therefore, similar resources are generally given to learners in the same way in these e-learning environments, however the resources are not followed in the same way by different learners who are usually asking for continuous online support and guidance throughout the e-learning process. To remedy such shortcomings, several works have dealt with adaptive e-learning systems and adaptive educational hypermedia. These systems are generally based on using one or more types of knowledge (learners' knowledge, learning material knowledge, learning process knowledge, etc) to perform personalization. Most of these systems have relied on explicit information given by a learner (demographic, questionnaire, etc) and have applied known methods and techniques of adapting the presentation and navigation (Chorfi et al., 2004). However, *automatic* personalisation and recommendation methods have not received sufficient attention in e-learning, even though they are well known and used in other fields. In fact, Web recommender systems have been used with success in e-commerce, information filtering, information retrieval, etc, but not used enough in e-learning.

In this paper, we present our design of a recommender system for e-learning environments by applying Web usage mining techniques and taking into account Web access history of learners and learning material. The proposed recommender system aims at automatically guiding the learner's activities and recommend relevant links and actions to him/her during the learning process. The following section presents an overview of Web personalization including personalization in e-learning. Section 3 discusses Web recommender systems and

[1] <http://www.moodle.org>

[2] <http://www.webct.com>

recommendation techniques. Section 4 describes the proposed approach and corresponding phases of profiling and recommendation. Section 5 presents some concluding remarks and future work.

Background on Web and e-learning personalization

During the last few years, Web personalization has become an important field of research in many application areas that are based on hypermedia and the Web, such as e-commerce, information retrieval, Web search system, e-learning, etc. The main goal of Web personalisation is to deliver to a given user information that is tailored to his/her preferences and interests. In the adaptive hypermedia area, adaptation concerned mainly the *content* of pages and *links* from pages. Therefore, as explained in (Brusilovsky, 1996), two different classes of adaptation can be considered: *adaptive presentation* and *adaptive navigation support*. Later, in (Brusilovsky, 2001), the taxonomy of adaptive hypermedia technologies was updated to add some extensions in relation with new technologies. Then, the distinction between two modes of adaptive navigation support became a necessity, especially with the growth of recommender systems. In fact, adapting links that were already prepared and presented on a certain page is quite different from generating *new* ones. Automatic recommendation implies that the user profiles are created and eventually maintained dynamically by the system without explicit user information (or at least a minimum). Examples include amazon.com's personalized recommendations, music recommenders like Mystrand.com in commercial systems (Mobasher, 2006) and smart recommenders in e-learning system (Zaiane, 2002), etc. In general, such systems differ in the input data, in user profiling strategies, and in prediction techniques. Several approaches for automatic personalization were reported in the literature, such as content based or item based filtering, collaborative filtering, rule based filtering, and techniques relying on Web usage mining, etc (Nasraoui, 2005).

In (Paramythis et al., 2004), a learning environment is considered *adaptive* if it performs the following tasks: monitoring user activities, interpreting them using specific models and using knowledge available on users to deliver content tailored to their needs. Thus, two main pillars of such systems should be distinguished : *user* (learner) and *content* (learning material). Hence, we should speak about *user modeling* and *content modeling*. On the other hand, it should be noted that personalization in e-learning systems generally concerns , *adaptive interaction*, *adaptive course delivery*, *content discovery and assembly*, and *adaptive collaboration support*. The category of *adaptive course delivery* represents the most common and widely used collection of adaptation techniques applied in e-learning systems today. Typical examples include dynamic course restructuring and adaptive selection of learning objects, as well as adaptive navigation support, which have all benefited from the rise of using recommendation strategies to generate new and relevant links and items. In fact, one of the new forms of personalization in e-learning environment is to give recommendations to learners in order to support and help them through the e-learning process.

Web Recommender systems

Web recommender systems are used to locate relevant items in which the user is interested. This can be done based on the user's data that is collected implicitly (Web access logs) or explicitly (ratings). Generally, it is more efficient and user-friendly to provide users with what they need *automatically* and without asking them explicitly for it (Mulvenna et al., 2000). Web recommender systems are used frequently in e-commerce and information access in order to assist the user in locating relevant products, items or services such as on Amazon³ and CDNow⁴. A number of knowledge discovery and statistical techniques are generally used in advanced recommendation systems. The latter can be divided, depending on the techniques used, into content-based filtering, collaborative filtering, and hybrids, which are summarized below. Other approaches such as demographic or knowledge based, also exist (but tend to require extensive private user information or manual construction of knowledge).

Content based filtering: Content-based filtering (or item-based filtering) systems recommend items to a given user based on the correlation between the content of these items and the preferences of the user (Meteren et al.,

[3] <http://www.amazon.com/>

[4] <http://www.cdnnow.com/>

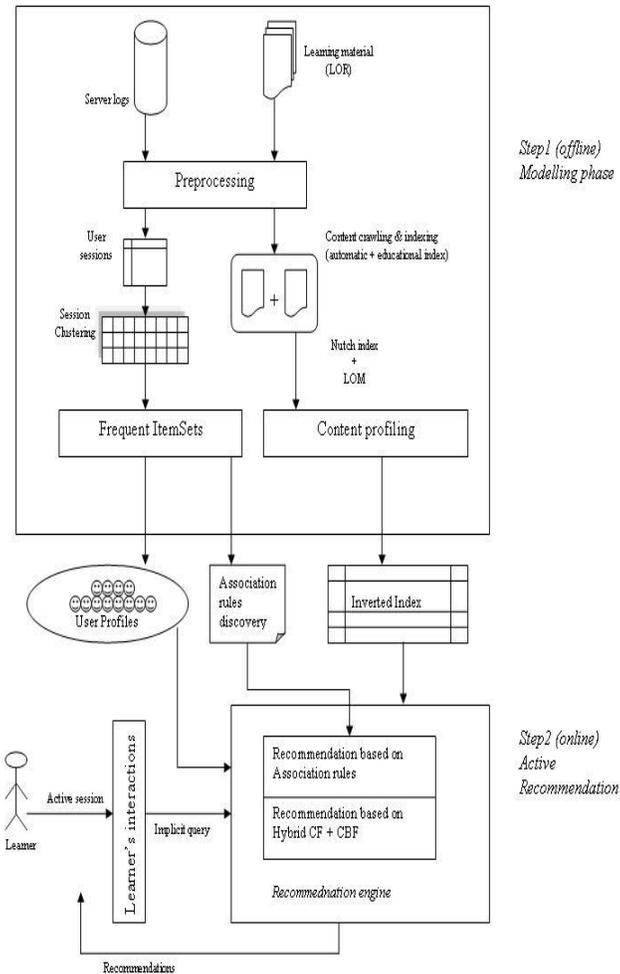


Figure 1: Proposed personalization approach

2000). This means that the recommended items are considered to be similar to those seen and liked by the same user in the past. Thus, there is no notion of a *community* of users, rather only *one* user profile is considered while making recommendations.

Collaborative based filtering: Collaborative filtering systems have tried to remedy for the shortcomings listed earlier. Collaborative filtering system recommends items that are liked by other users with similar interests. So, the exploration of new items, in this approach, is assured by the fact that *other similar* user profiles are also considered. Thus, the history of *community* of users is combined. Examples of such systems include GroupLens (Konstan et al., 1997) and (Sarwar et al., 1998). This approach relies on a historic record of all user preferences and interests which can be obtained *explicitly* by asking users to give a rating on each item (product, Web page, etc) or *implicitly* by observing the user's behaviour when browsing a Website, for instance via their clickstreams.

Hybrids: Hybrid recommender systems combine *several* recommendation strategies to provide better performance than either strategy alone. Most hybrids work by combining several input data sources or several recommendation strategies. There are many hybridization methods reported in the state of the art. Generally, content/collaborative hybrids are the most popular hybrid strategies.

Proposed framework for building automated recommender systems on e-learning platforms

Our aim is to build a recommender system which, when integrated within an e-learning system, will

collect and mine Web usage access logs and learning material for the user, in both usage and content profiling phases, and will later determine what to recommend to an active learner⁵. The latter task can be considered as the online task of e-learning personalization. This task will consist of computing a recommendation set of URLs that the active learner (current session) may visit, based on similar usage patterns and based on page content. Recommended URLs could be divided into two sets : the first set contains the most relevant URLs indicating the next pages to visit by the learner (next choice), while the second set contains a list of related links which are recommended to visit, based on similar content. To generate such recommendations, our recommender system must perform the following steps :

- Preliminary offline mining of usage profiles (done offline) based on Web mining techniques. First, we apply a clustering approach to directly cluster user sessions. Each cluster will contain similar sessions, thus showing , similar interests of different learners. Each cluster can also be viewed as one user *profile*;
- Preliminary offline mining of association rules (e.g. “Resource A → Resource B”) from clustered sessions;
- Preliminary crawling and indexing of learning resources (done offline): this step consists of crawling the entire learning resources available in a course repository and forming an *inverted* index mapping each keyword to a set of pages in which it is contained;
- Matching a new user session (*active session*) to related association rules that will be used in recommendation;
- Matching a new user session to previous cluster profiles to form a *collaborative session*;
- Query formation and scoring: mapping a new user session (or a collaborative session) into a set of terms to use as an *implicit query* in order to retrieve related URLs from the inverted index.

Like most Web recommender systems, our proposed system is composed of two modules: an off-line module which pre-processes data to build user and content profiles, and an on-line module which uses these models on-the-fly to recognize user goals and predict a recommendation list. The proposed approach, with main features shown in (Fig 1), is essentially based on two components: *the Modeling phase* and the *Recommendation phase*.

Modeling phase

User profiling: represents the key process of recommendation systems. In fact, (Koch, 2000) describes the necessity of user models as follows: “*Users are different: they have different background, different knowledge about a subject, different preferences, goals and interests. To individualise, personalise or customise actions, a user model is needed that allows selection of individualised responses to the user.*” In our approach, we’ll apply data mining techniques to build user profiles. In fact, the prediction of the user model is accomplished using, not explicit user interaction but implicit information collected from the past usage sessions. Thus, user behavior becomes, in this case, the source of implicit prediction for his/her model. The input data for this first step is composed mainly of Web server access log files and learning object files, necessary for data preparation and pattern discovery. When dealing with Web data, we speak about Web mining (as for the case of data mining) which is one of the most important stages in the “Knowledge Discovery in Data” or *KDD* process (Tanasa et al., 2004). Three axes of Web mining could be explored : *Web Content Mining*, *Web Structure Mining* and *Web Usage Mining*. *Web usage* mining can be defined as the process of analyzing the user’s browsing behaviour by relying on Web access logs. In fact, in Web based systems, it is possible to get continuous feedback detailing a learner’s access and usage information since every request received by the Web server is automatically saved in a specific file called the log file. In our future experiments, we will use as input data Web usage logs collected from access server log files of RPL platform hosted on the Virtual University (*UVT*) servers. This log data must first be pre-processed, using task that include : data cleaning, user identification, and session identification. Data cleaning involves tasks such as deleting uninteresting elements in log files (graphics, icons, requestss generated by crawlers/bots, etc). *User identification* aims at identifying unique user records, thus recognizing the sequence of logged activities belonging to the same user. *Sessionization* represents the process of segmenting the user activity log of each user into distinct *sessions*, each occurring within a single visit to the e-learning platform. It should be noted that the obtained sessions are considered as sets of URLs visited by the learners. Let U a set of n unique URLs appearing in preprocessed log files : $U = \{ u_1, u_2, u_3, \dots, u_n \}$ and S a set of m users’ sessions extracted from pre-processed log files : $S = \{ S_1, S_2, S_3, \dots, S_m \}$, where each $S_i \in S$ is a subset of U . Once sessions are delimited properly, we can apply data mining methods to build user profiles. First, we’ll cluster sessions based on their access similarity. Next, each obtained cluster will be split into a set of URLs based on

[5] A demo is available on RPL platform : <http://cours.uvt.rnu.tn/rpl/>

their co-occurrence frequency. A variety of clustering techniques can be used for clustering sessions. (Shahabi et al., 1997) describe a prototype system that uses viewing time as the primary feature to describe a user session and then clusters the sessions using K-Means clustering. (Banerjee et al., 2001) utilized the combination of time spent on a page and Longest Common Subsequences (LCS) to cluster the user sessions. (Mobasher et al., 2000) used a multi-variate K-Mmeans algorithm to obtain transaction clusters and the Association Rule Hypergraph Partitioning (ARHP) technique to obtain usage clusters. (Yao et al., 2002) applied the Leader algorithm for clustering. (Nasraoui et al., 2001) presented a new hierarchical clustering technique based on the concept of evolving genetic niches, called Hierarchical Unsupervised Niche Clustering (HUNC) which offers the advantage of multi-resolution clustering. The output of most of these methods is a set of page clusters, each of which indicate the Web pages frequently visited together by similar users. To apply clustering approaches on S , we have to view each $S_i \in S$ as a k -length sequence of ordered pairs $S_i = \langle (u_1^{S_i}, w(u_1^{S_i})), (u_2^{S_i}, w(u_2^{S_i})), (u_3^{S_i}, w(u_3^{S_i})), \dots, (u_k^{S_i}, w(u_k^{S_i})) \rangle$, where each $u_j^{S_i} = u_i$ for some $i \in \{1, \dots, n\}$, and $w(u_j^{S_i})$ is the weight associated with URL reference $u_j^{S_i}$ in the session S_i . The order of URL references within session is temporal. These weights can be binary, representing the existence or non-existence of a URL reference in the session; or they can be computed as a function of a number of features based essentially on the frequency of occurrence of URLs within a session and the time a learner spends on a particular page as a manner to determine indirectly the fact that a learner liked or disliked the URL (Shahabi et al., 1997) and (Yan et al., 1996). However, due to the fact that we don't use explicit ratings by learners of learning resources, we have chosen to compute each page weight based on occurrence frequency and time spent to express user preferences. Essentially, session clustering will result in a set $C = \{C_1, C_2, \dots, C_k\}$ of clusters, where each C_i is a subset of user sessions. Therefore, each cluster will represent a group of similar learners with similar access patterns. Such a cluster can contain hundred of user sessions, which in their turn can include hundred of URL references. But, to obtain accurate clusters giving an effective aggregated view of user profiles, we have to split each cluster C_i into URL clusters using a frequent itemset mining algorithm that extracts frequently co-occurring URL sets in the sessions of each cluster.

Content profiling: generally involves applying indexing and text mining (Web content mining) techniques. Of particular interest to the proposed approach is the use of the Nutch open source search engine in the content modeling phase, followed by content base filtering as a recommendation strategy. So, we will (1) automate the indexing of the learning material using crawling and indexing techniques as done in the Nutch search engine, and (2) automate the indexing of educational content based on norms and standards used in e-learning. Nutch⁶ is an open source *Web search software* built on Lucene Java (high-performance full-featured text search engine library written entirely in Java). Nutch's core is based on four major components: Searcher, Indexer, Database and Fetcher (Khare et al., 2004). The Nutch crawler uses a URL file to create (accomplishing fetching, parsing and indexing phases) the inverted index which will be used to represent the model of educational content. This inverted index maps each keyword to a set of pages in which it is contained. In order to adapt this content modeling to the pedagogical area, we have to enrich the inverted index that is automatically obtained via Nutch's crawling and indexing mechanisms, by adding an educational index used in LOM (Learning Objects Metadata Standard). Workgroups of IEEE, that included the participation of many consortiums such as ARIADNE⁷ and IMS⁸, have proposed using a set of descriptors known as LOM (*Learning Object Metadata*) and used to index learning content. This method of indexing has the advantage to give learning objects more sense, semantic and pedagogical value, so that they can be accurately referred and found. But, despite librarian and author efforts to index learning objects, many of these entities remain without any added indexing information because of the difficulty of this task. Thus, adding indexing used in LOM for learning content (if available) to the preliminary crawling and indexing phase done by Nutch should improve the accuracy of the final index, and likewise improve search and recommendations.

Recommendation phase

The Recommendation phase is done *online*. The active user session is transformed into an implicit

[6] <http://lucene.apache.org/nutch/>

[7] <http://www.ariadne-eu.org/>

[8] <http://www.imsproject.org/>

query extracted from logs and submitted to the system. The input to the recommendation system is deduced from the active user session by transforming a new user session (Fig. 2) into a *query* composed by the last access pages representing a short-term history for the current user. These last visited pages in the active user session are called a sliding window. For example, let W be a fixed size for a sliding window, then if the active user session with $W = 3$ is $\{A, B, C\}$, and the URL “ D ” was viewed last by a user, then the new sliding window becomes $\{B, C, D\}$. It should be noted that W can lead to lower or higher recommendation coverage. In our case, we consider a fixed window size $W = 3$, hence only the last three visited pages will affect the recommendation. The system will extract visited URLs from the Web log file from the time that the learner connects to the platform until he/she asks for recommendations.

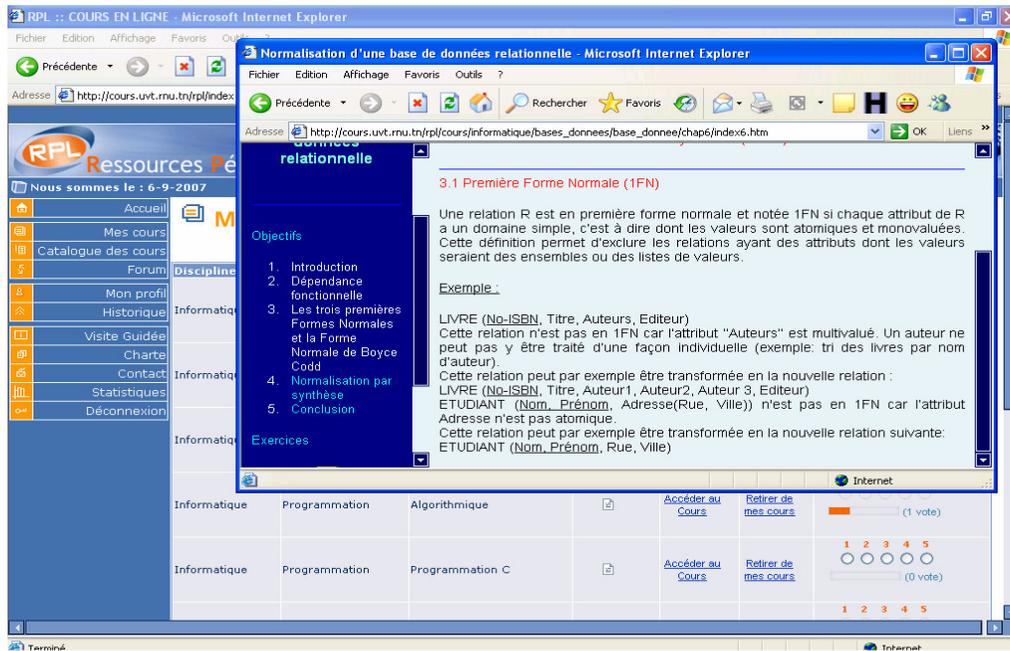


Figure 2: The active learner is browsing material (active user session)

Prediction of next pages to visit (Type 1): this task computes the first type of recommendations and consists of accurately computing all relevant possibilities of next pages that can be recommended to visit. We'll use the discovered cluster association rules to accomplish this task. These association rules (AR) will capture the relationships among URLs based on their co-occurrence across sessions. AR discovery methods such as the Apriori algorithm (Agrawal et al., 1994) can be used directly especially when frequent itemsets (pages occurring frequently together in many sessions within a given cluster) have been determined previously in the first phase of user profiling. An association rule r is an expression of the form: $A \Rightarrow B$, where A and B are itemsets, r must satisfy a minimum confidence threshold to be involved in the recommended page set. The generated association rules will take, for example, the form: 70% of learners who accessed the Web page A , also accessed B . All rules should first be sorted by their confidence (highest to lowest) when forming the recommendations. Also, the current session window is matched against the "condition" or left side of each rule. (Example : "75.43 % of learners who visited : http://cours.uvt.rnu.tn/rpl/cours/informatique/bases_donnees/base_donnee/chap6/menu.htm visited : http://cours.uvt.rnu.tn/rpl/cours/informatique/bases_donnees/base_donnee/chap6/obje.htm and http://cours.uvt.rnu.tn/rpl/cours/informatique/bases_donnees/base_donnee/chap6/index6.htm")⁹.

Generation of related recommended links (Type 2): this step yields the second type of recommendations, and is accomplished using hybrid recommendation strategies. Several methods of recommendation will be investigated in our work. First, we applied content based filtering approach using Nutch search engine (Fig. 3) and collaborative based filtering approach using association rules (Fig. 4) alone to recommend related links. But, basically, the idea is to apply *both* of them (content based filtering and collaborative based filtering

[9] Example of Association Rules extracted from RPL platform using Apriori algorithm

approches) simultaneously in order to improve the recommendation quality and generate the most relevant learning objects to learners. Two approaches can be considered : *Hybrid content via profile based collaborative filtering* with cascaded/feature augmentation combination and *Hybrid content and profile based collaborative filtering* with weighted combination (Nasraoui et al., 2006).

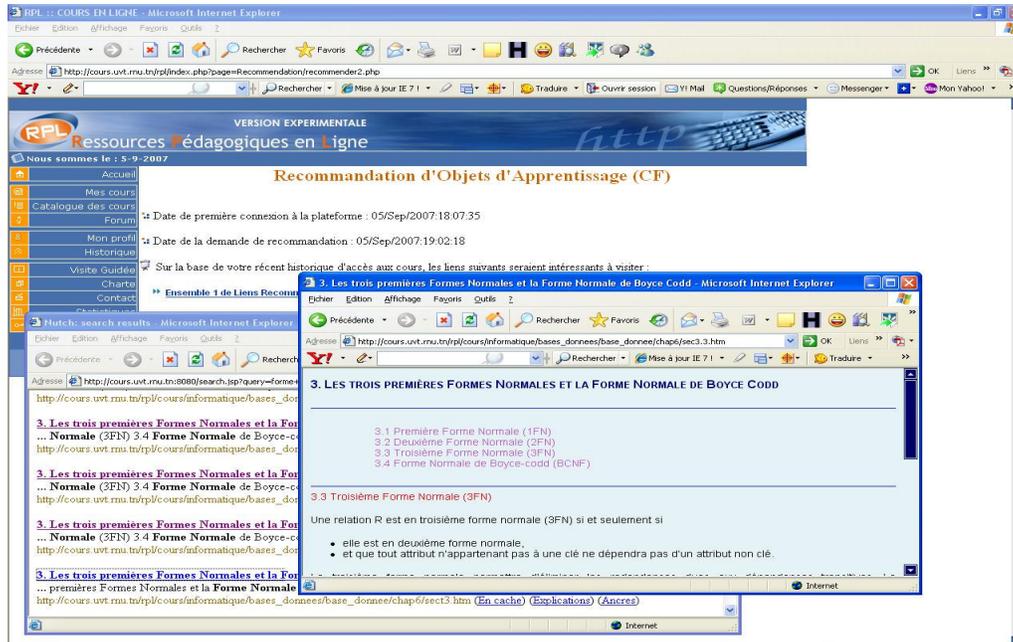


Figure 3: Recommendation using content based filtering approach

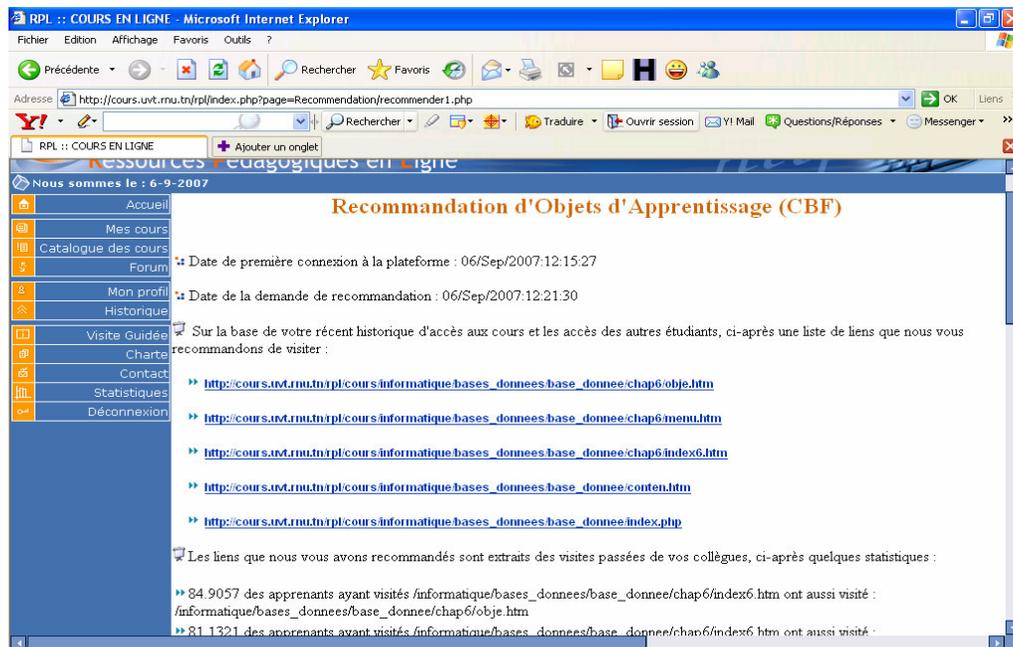


Figure 4: Recommendation using collaborative based filtering approach

In the Hybrid content via profile based collaborative filtering with cascaded/feature augmentation combination approach (Fig. 5), once we have user and content profiles resulting from the profiling phase and the recommendation input consisting of the last $W=3$ visited pages (sliding window), the following steps remain:

- i) Matching sliding window pages to previous profiles to form a *collaborative session*: having a set of pages representing the active session window of a given learner, this step consists of searching for these

- pages in existing profiles in order to determine the closest profiles. URLs belonging to these closest profiles will be combined based on their overlap ratio to form a collaborative session;
- ii) Each URL in the above collaborative session is then mapped to a set of content terms characterizing as well as possible this URL (top k frequent terms), a *parser* tool must be used for this task. Finally, these terms are submitted to the search engine Nutch which returns related URLs (Fig. 6).

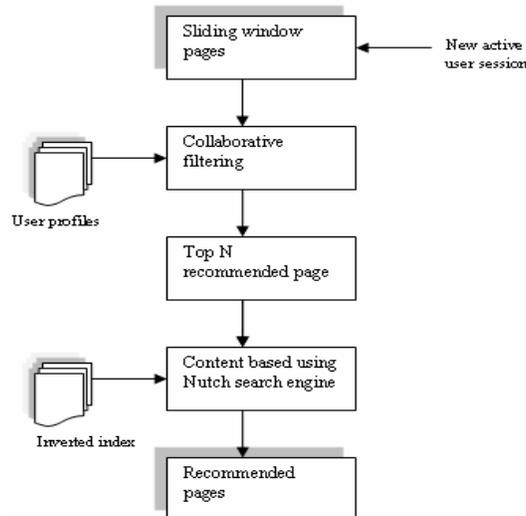


Figure 5: Cascaded content based via collaborative profile filtering

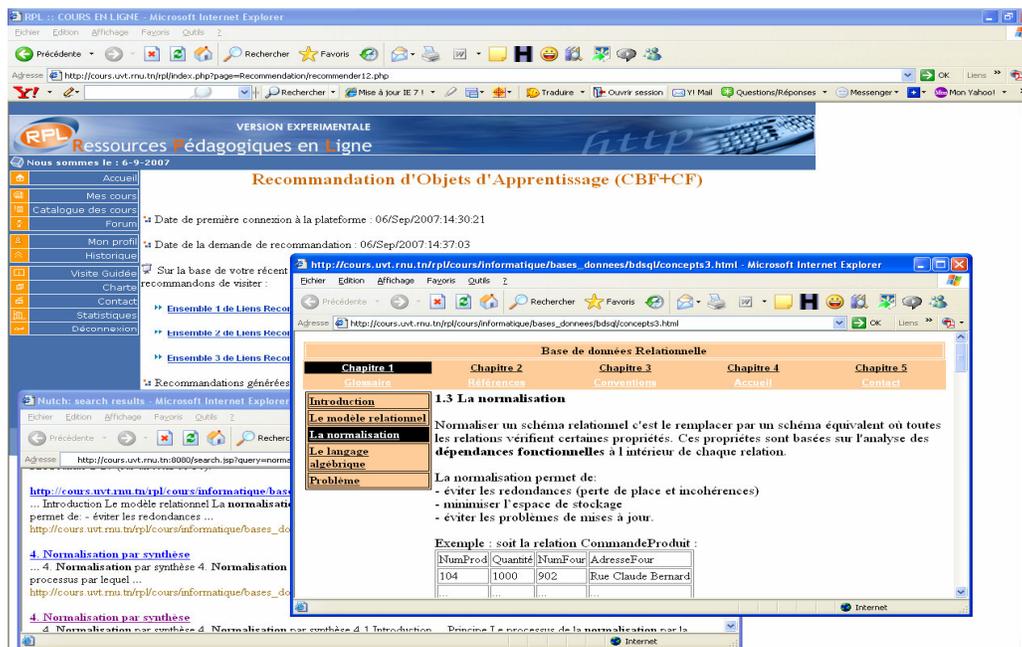


Figure 6: Recommendation using cascaded hybrid (collaborative filtering followed by content-based filtering)

In the Hybrid content and profile based collaborative filtering with weighted combination approach (Fig. 6), the collaborative filtering and content based filtering are performed *separately*, then the results of both techniques are combined together to produce a single recommendation set. This process uses the following steps:

- i) Step 1 is performed in the same way as the previous approach, the result is called *Recommended Set 1* ;

- ii) Step 2 maps each URL in the sliding window to a set of content terms characterizing as well as possible this URL (top k frequent terms). Then these terms are submitted to the search engine which returns related URLs. This result is called *Recommended Set 2*.
- iii) Final collaborative and content based filtering recommendation combination: both recommended sets obtained previously are combined together to form a coherent list of related recommendation links, which are ranked based on their overlap ratio (Fig. 8).

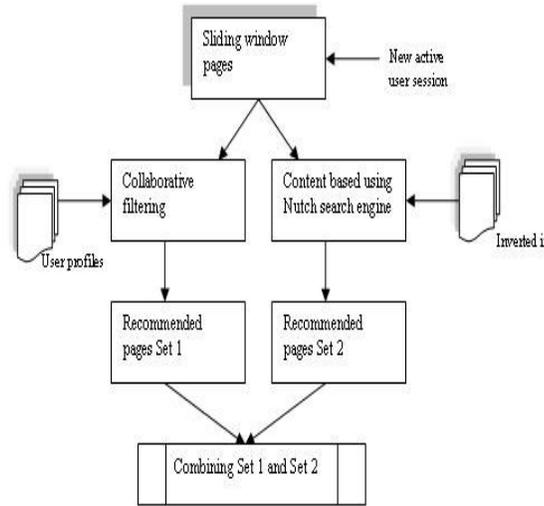


Figure 6 : Weighted content based and collaborative profile filtering

Conclusions and future work

In this paper, we are interested in automatic personalisation based on recommendation in e-learning. We have outlined the general principles of the proposed approach using recommendation strategies to perform personalization in e-learning platforms. In the modeling phase, we used Nutch's automated crawling and indexing techniques as well as standardized educational content indexing to build content profiles, and Web usage mining techniques (clustering and association rule mining) to build user profiles. Hybrid recommendations (content based filtering and collaborative based filtering) were used in the recommendation phase. In the future, we will present our experimental results and metric, and explore in detail several techniques and strategies in the modeling and recommending phases.

Acknowledgments:

This research was partially supported by National Science Foundation CAREER Award IIS-0133948 to O. Nasraoui.

References

- Agrawal, R & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September 1994.
- Banerjee, A. & Ghosh, J. (2001). Clickstream Clustering using Weighted Longest Common Subsequences, in Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining (Chicago IL, April 2001), 33-40.
- Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6 (2-3), pp. 87-129. 1996.
- Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110. 2001.

- Chorfi, H. & Jemni, M. (2004). PERSO : Towards an adaptative e-learning system. *Journal of Interactive Learning Research*, 15 (4), pp 433-447.
- Khare, R., Cutting, D., Sitaker, K. & Rifkin, A., (2004) Nutch: A Flexible and Scalable Open-Source Web Search Engine. CommerceNet Labs Technical Report 04-04. November 2004.
- Koch, N. (2000) Software Engineering for Adaptive Hypermedia Systems. Phd Thesis, Ludwig-Maximilians-University, Munich/Germany, 2000.
- Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L. & Riedl, J. (1997) GroupLens : applying collaborative filtering to usenet news. *Communications of the ACM* (40) 3, 1997.
- Meteren, R. V. & Someren, M. V. (2000) "Using Content-Based Filtering for Recommendation". *MLnet / ECML2000 Workshop*, May 2000, Barcelona, Spain.
- Mobasher, B. (2006) Data Mining for Web Personalization. *The Adaptive Web: Methods and Strategies of Web Personalization*. Lecture Notes in Computer Science, Vol. Springer-Verlag, Berlin-Heidelberg New York, 2006.
- Mobasher, B., Cooley, R., & Srivastava, J. (2000) Automatic personalization based on Web usage mining. *Communications of the ACM*, Vol 43(8) 142-151. 2000.
- Mulvenna, M., Anand, S. S. & Buchner, A. G. (2000) Personalization on the net using Web mining. *Communication of ACM* 43(8). 2000.
- Nasraoui, O. & Krishnapuram, R. (2001) Robust Multi-resolution Web Usage Data Mining using Hierarchical Unsupervised Niche Clustering. Proc. ANNIE (Artificial Neural Network In Engineering) Conference, pp.369-374, St. Louis, MO, Nov. 2001, received Best paper award for theoretical developments in computational intelligence.
- Nasraoui, O., Zhang, Z., & Saka, E. (2006) Web Recommender System Implementations in Multiple Flavors: Fast and (Care) Free for All. In Proceedings of the ACM-SIGIR Open Source Information Retrieval, 2006.
- Nasraoui, O. (2005) World Wide Web Personalization, Invited chapter in "*Encyclopedia of Data Mining and Data Warehousing*", J. Wang, Ed, Idea Group, 2005.
- Paramythis, A & Loidl-Reisinger, S. (2004) Adaptive Learning Environments and e-Learning Standards. *Electronic Journal of e-Learning*, 2 (1), 181-194. 2004.
- Sarwar, B.M., Konstan, J., Borchers, A., Herlocker, G., Miller, B., & Riedl, J. (1998) Using filtering agents to improve prediction quality in the grouplens research collaborative filtering systems. In Proceedings of the ACM Conference on Computer Supported Cooperative Work, Seattle, Washington. 1998.
- Shahabi, C., Zarkesh, A.M., Adibi, J. & Shah, V. (1997) Knowledge Discovery from User's Web-page Navigation, in *Proc. 7th IEEE Intl. Conf. On Research Issues in Data Engineering* (1997), 20-29.
- Tanasa, D., Trousse, B. & Masségli, F. (2004) Application des techniques de fouille de données aux logs Web: Etat de l'art sur le Web Usage Mining, Projet AxIS, INRIA Sophia ANTIPOLIS, 2004.
- Yan, T., Jacobsen, M., Garcia-Molina, H. & Dayal, U. (1996) From user access patterns to dynamic hypertext linking. In Proceedings of the 5th International World Wide Web Conference, Paris, France, 1996.
- Yao, Y.Y., Hamilton, H.J. & Wang, X. (2002) PagePrompter: An Intelligent Agent for Web Navigation Created Using Data Mining Techniques. In Proceedings of the Third International Conference : Rough Sets and Current Trends in Computing, RSTC 2002, Malvern, PA, USA, October 14-16, 2002.
- Zaiane, OR. (2002) Building a Recommender Agent for e-Learning Systems, in Proc. of the 7th International Conference on Computers in Education (ICCE 2002) pp 55-59, Auckland, New Zealand, December 3 - 6, 2002.