USING USER BEHAVIOR SIMILARITY FOR RECOMMENDATION COMPUTATION: THE BROADWAY APPROACH

B. Trousse, M. Jaczynski R. Kanawati
INRIA Sophia-Antipolis, France

1 The Broadway Recommendation Approach

Recommendation systems have gained a lot of attention recently especially in the field of on-line information retrieval (IR) systems (e.g. searching documents on the web). Recommenders are traditionally partitioned into two main families: *content-based* recommenders and *collaborative filtering* ones (Resnick et al 1997). Systems of the first type recommend items or actions to the user depending on an evaluation of the user own past actions, while those belonging to the second family recommend to a user items positively evaluated by other *similar* users. In this work we describe a new recommendation approach, called the Broadway approach, where the system recommends to a user what have *satisfied* other users (or eventually *similar* users) that have *behaved similarly* to that user. Other systems in IR field use user behavior similarity as a basis for recommendation computation (Yan et al 1996). However, the Broadway approach has the particularity to model user behaviors by observation variables rather than matching user actions to a pre-specified behavior model.

Following the Broadway approach, the user interactions with the application are saved in a log-like file. This log file contains a set of *time series*, each holds the evolution with time of a variable that is said to be relevant to describe the user behavior. Obviously the choice of these variables depends on the application field (cf. Section 2). Time series are grouped into *records* that correspond to a user session with the system within a well specified period of time which has a well defined semantic in the application.

Now CBR methodology is used (cf. Case-Based reasoning (Kolodner, 1993)): it is a problem solving methodology where, in order to find the solution to a current problem, one looks for a similar problem in an experience base, takes the solution from the past and use it for a starting point to find a solution to the current problem. A *case* is a contextualized piece of knowledge representing an experience. It is generally composed of two main parts : the problem and the solution.

Also, from records, we extract potential useful experiences, called *potential cases,* by using a *case template* issued from domain experts. Such a template identifies useful *situations* from saved sessions (or records) as well as the solutions (i.e. recommendations) suggested by these situations. The situation, formed of a sub-history of the user session, is used to retrieve past situations that match the most the current user behavior and could explain the current behavior in that session. The definition of a situation (i.e. indices that are said to be relevant to *explain* the user behavior) depends on the nature of recommendations we want to provide (cf. Section 2). Figure 1 illustrates the classical CRB system cycle applied by the Broadway approach.

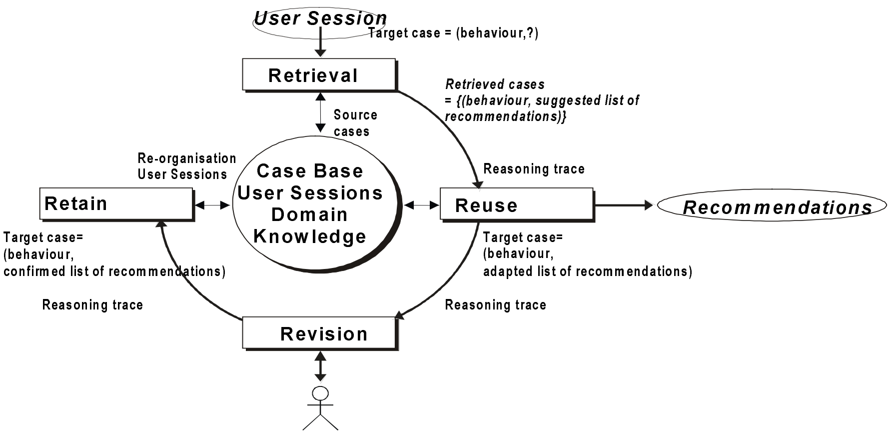

Figure 1. The Broadway recommendation computation approach cycle

Using CBR technology to implementing the Broadway approach needs to deal with cases with temporal indices (Jaczynski, 1997). We have developed an object-oriented framework called CBR*Tools (written in Java) that allows the manipulation of such cases. To sum up, the application of the Broadway recommendation computation approach requires the following steps: 1/ Identification of variables to be used to model the user behavior, 2/ Definition of records and associated semantics, 3/ Definition of case templates to use, 4/ Definition of reasoning steps (retrieve, reuse , revise and retain).

2 Applications for the Web

Next, two applications of the Broadway approach are briefly described : a web navigation advisor and a query refinement recommender.

2.1 Supporting Browsing on the Web : Broadway-V1

Applying the Broadway approach in the context of web browsing (Jaczynski and Trousse, 1998) aims at reusing browsing choices made by past users in order to help a user engaged in a browsing process. The user behavior is modelled by two types of variables :

1. *Browsing description variables* that describe the visited Web pages. A visited page is characterised by its address (i.e. URL) and a content description variable (i.e list of keywords for example).
2. *User satisfaction variables*. These variables are needed to measure the user satisfaction from visited pages. User satisfaction can be expressed explicitly by the user or implicitly estimated by the system (ex. the time spent on displaying a page over the page size could be used as an implicit satisfaction metric).

The evolution of the above variables are saved into records (i.e the navigation base), each representing a browsing session. More precisely in Broadway-V1, such a record is based on four variables chosen to observe the navigation process (cf. Figure 2): URL, title keywords, user evaluation, reading time ratio. The time series associated to each variable are sampled since the unit of the chosen model of time is a change of HTML pages.

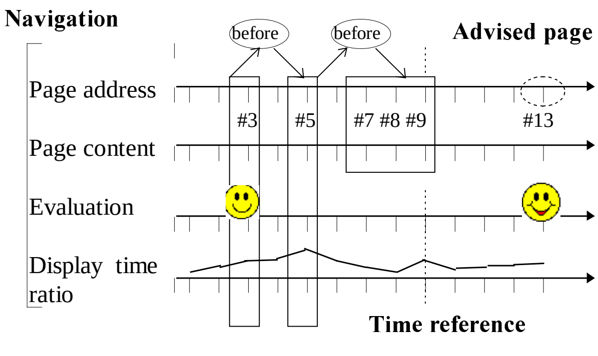

Figure 2. Example of a record and a case

Cases can be identified by a direct search inside the records according to a *potential case template* defining typical situation constraints. Such a *template is based on* a restriction composed of the last three visited pages and by different browsing events well evaluated (explicitly or implicitly) as shown in Figure 1. Such cases could become concrete cases if they are used during the reasoning of the broadway-V1 engine i.e. enough similar to the current browsing session and adding some new information the concrete case base. This application has been experimentally evaluated by ergonomists and ten students in Cognitive Psychology. Results summarised in (Jaczynski and Trousse, 1998) show that the Broadway approach helps to reduce the time and the navigation length.

2.2 Supporting query refinement

Querying is known to be an iterative process : a user submits a query, evaluates the adequacy of obtained results and if s/he is not satisfied with these results the query is modified and re-submitted. Applying the Broadway approach for helping in query refinement aims at reusing *good* query refinement decisions made by past users of a search engine in order to help current users in refining their queries. The user behavior is modelled by three types of variables :

3. *Query configuration variables* that describe the submitted query. A query syntax can vary from simple list of keywords, as in the basic search mode in most of web search engines, to a complex structure (i.e SQL like queries). What is important, regarding our application is to be able to define a reliable query similarity measurement.

4. *Raw results variables* that describe results returned by the search engine in answer to a given query.

5. *User satisfaction variables* that describe the user satisfaction from answers returned by the search engine. The user satisfaction can be simply taken to be the classical query precision measurement.

The evolution of the above variables are saved into records corresponding *to information searching sessions.* In a very natural way, a useful experience in query refinement process is taken to be the *p* past query refinement steps. As additional indices that could *explain* query refinement decisions, we take the set of past results that has been positively evaluated by the user (these could describe the user goal), the set of results that have been rejected by the user and the set of negatively evaluated queries (these could describe the query refinement motivation). Similarity between current and past situations is computed using the above cited sets of variables. CBR technology allows us to retrieve past situations that are similar to a current one, queries that are positively evaluated in past situations are then proposed to the current user. This approach has been applied in a concrete application, the BeCBKB application, which integrates a Broadway-based query refinement recommender in the context of the CBKB meta-search engine developed by Xerox Research Center

Europe. Detailed description of the BeCBKB application can be found in (Kanawati et al, 1999).

3 Conclusion

In this paper, we have briefly presented a new approach for recommendation computation in general, called the Broadway approach. The originality of our approach relies mainly on using the CBR methodology, on extracting potential cases from user sessions and on managing cases with temporal indices. Furthermore our approach is implemented using the CBR*Tools framework which facilitates the development of such recommender systems. Two examples of Broadway-based recommender systems in information retrieval on the Web have quickly described. It is worth to say that in information retrieval, the Broadway approach based on user session cases complements the other classical recommendation approaches (e.g. content-based and collaborative filtering recommenders) rather than replacing them. More generally, the Broadway approach proposes a new type of computerised assistants based on reusing user sessions cases.

4 References

Jaczynski, M (1997). A framework for the management of past experiences with time-extended situations. In *Proceedings of CIKM'97*, pages 32-39. Las Vegas, Nevada

Jaczynski, M., Trousse, B. (1998). WWW assisted browsing by reusing past navigations of a group of users. *In: Proceedings of EWCBR'98*, volume 1488 of LNAI, Springer Verlag, pages 160-171 - Dublin, Irland,.

Kanawati, R., Jaczynski, M., Trousse, B., Andreoli, J-M.. (1999). Applying the Broadway recommendation computation approach for implementing a query refinement service in the context of the CBKB meta-search engine. In *Proceedings of RàPC'99*, Palaiseau, June, 1999.

Kolodner, J. (1993) *Case-Based Reasoning*. Morgan Kaufmann, 1993.

Resnick, P., Varian, H.R. (1997). Recommender systems. *Communications of the ACM*, 40 (3).

Yan, T.W., Jacobsen, M., Garcia-Molina, H. & Dayal, U. (1996). From user access patterns to dynamic hypertext linking. *Computer Network and ISDN systems*, vol. 28, 1007-1014.