Case-Based Recommender Systems

Fabiana Lorenzi
Universidade Luterana do Brasil, Brazil

Francesco Ricci
eCommerce and Tourism Research Laboratory, ITC-irst, Italy

INTRODUCTION

Recommender systems are being used in e-commerce web sites to help the customers in selecting products more suitable to their needs. The growth of Internet and the business to consumer e-Commerce has brought the need for such a new technology (Schafer, Konstan, & Riedl., 2001).

BACKGROUND

In the past years, a number of research projects have focused on recommender systems. These systems implement various learning strategies to collect and induce user preferences over time and automatically suggest products that fit the learned user model.

The most popular recommendation methodology is collaborative filtering (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994) that aggregates data about customer's preferences (ratings) to recommend new products to the customers. Content-based filtering (Burke, 2000) is another approach that builds a model of user interests, one for each user, by analyzing the specific customer behavior. In collaborative filtering the recommendation depends on the previous customers' information, and a large number of previous user/system interactions are required to build reliable recommendations. In content-based systems only the data of the current user are exploited and it requires either explicit information about user interest, or a record of implicit feedback to build a model of user interests. Content-based systems are usually implemented as classifier systems based on machine learning research (Witten & Frank, 2000). In general, both approaches do not exploit specific knowledge of the domain. For instance, if the domain is computer recommendation, the two above approaches, in building the recommendation for a specific customer, will not exploit knowledge about how a computer works and what is the function of a computer component.

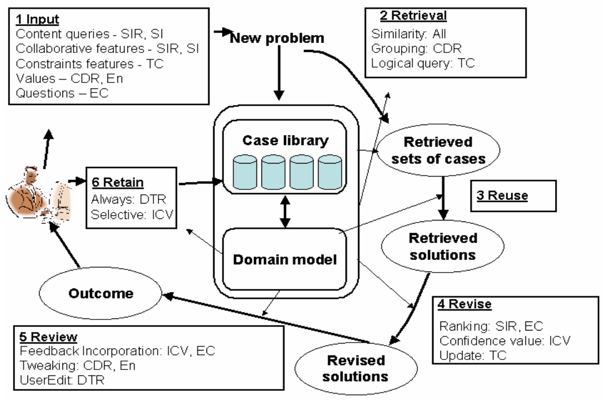
Conversely, in a third approach called knowledge-based, specific domain knowledge is used to reason about what products fit the customer's preferences (Burke, 2000). The most important advantage is that knowledge can be expressed as a detailed user model, a model of the selection process or a description of the items that will be suggested. Knowledge-based recommenders can exploit the knowledge contained in case or encoded in a similarity metric. Case-Based Reasoning (CBR) is one of the methodologies used in the knowledge-based approach. CBR is a problem solving methodology that faces a new problem by first retrieving a past, already solved similar case, and then reusing that case for solving the current problem (Aaamodt & Plaza, 1994). In a CBR recommender system (CBR-RS) a set of suggested products is retrieved from the case base by searching for cases similar to a case described by the user (Burke, 2000). In the simplest application of CBR to recommendation problem solving, the user is supposed to look for some product to purchase. He/she inputs some requirements about the product and the system searches in the case base for similar products (by means of a similarity metric) that match the user requirements. A set of cases is retrieved from the case base and these cases can be recommender to the user. If the user is not satisfied with the recommendation he/she can modify the requirements, i.e. build another query, and a new cycle of the recommendation process is started.

In a CBR-RS the effectiveness of the recommendation is based on: the ability to match user preferences with product description; the tools used to explain the match and to enforce the validity of the suggestion; the function provided for navigating the information space. CBR can support the recommendation process in a number of ways. In the simplest approach the CBR retrieval is called taking in input a partial case defined by a set of user preferences (attribute-value pairs) and a set of products matching these preferences are returned to the user.

MAIN THRUST

CBR systems implement a problem solving cycle very similar to the recommendation process. It starts with a new problem, retrieves similar cases from the case base and shows to the user an old solution or adapts it to better

Figure 1. CBR-RS' framework (Lorenzi & Ricci, 2004)



solve the new problem and finishes retaining the new case in the case base. Considering the classic CBR cycle (see Aamodt & Plaza, 1994) we specialized this general framework to the specific tasks of product recommendation. In Figure 1 the boxes, corresponding to the classical CBR steps (retrieve, reuse, revise, review, and retain), contain references to systems or functionalities (acronyms) that will be described in the next sections.

We now provide a general description of the framework by making some references to systems that will be better described in the rest of the paper. The first user-system interaction in the recommendation cycle occurs in the input stage. According to Bergmann, Richter, Schmitt, Stahl, and Vollrath (2001), there are different strategies to interact with the user, depending on the level of customer assistance offered during the input. The most popular strategy is the dialog-based, where the system offers guidance to the user by asking questions and presenting products alternatives, to help the user to decide. Several CBR recommender systems ask the user for input requirements to have an idea of what the user is looking for. In the First Case system (McSherry, 2003a), for instance, the user provides the features of a personal computer that he/she is looking for, such as, type, price, processor or speed. Expertclerk (Shimazu, 2002) asks the user to answer some questions instead of provide requirements. And with the set of the answered questions the system creates the query.

In CBR-RSs, the knowledge is stored in the case base. A case is a piece of knowledge related to a particular context and representing an experience that teaches an essential lesson to reach the goal of the problem-solving activity. Case modeling deals with the problem of determining which information should be represented and which formalism of representation would be suitable. In CBR-RSs a case should represent a real experience of solving a user recommendation problem. In a CBR-RS, our analysis has identified a general internal structure of the case base: $CB = [X \times U \times S \times E]$.

This means that a case $c = (x, u, s, e) \in CB$, generally consists of four (optional) sub-elements $x, u, s, e,$ which are elements of the spaces X, U, S, E respectively. Each CBR-RS adopts a particular model for the spaces X, U, S, E. These spaces could be empty, vector, set of document (textual), labeled graphs, etc.

- **Content model (X):** the content model describes the attributes of the product.
- **User profile (U):** the user profile models personal user information, such as, name, address, and age or also past information about the user, such as her preferred products.
- **Session model (S):** the session model is introduced to collect information about the recommendation session (problem solving loop). In DieToRecs, for instance, a case includes a tree-based model of the user interaction with the system and it is built incrementally during the recommendation session.
- **Evaluation model (E):** the evaluation model describe the outcome of the recommendation, i.e., if the suggestion was appropriate or not. This could be a user a-posteriori evaluation, or, as in (Montaner, Lopez, & la Rosa, 2002), the outcome of an evaluation algorithm that guesses the goodness of the recommendation (exploiting the case base of previous recommendations).

Actually, in CBR-RSs there is a large variability in what a case really models and therefore what components are really implemented. There are systems that use only the content model, i.e., they consider a case as a product, and other systems that focus on the perspective of cases as recommendation sessions.

The first step of the recommendation cycle is the retrieval phase. This is typically the main phase of the CBR cycle and the majority of CBR-RSs can be described as sophisticated retrieval engines. For example, in the Compromise-Driven Retrieval (McSherry, 2003b) the system retrieves similar cases from the case base but also groups the cases, putting together those offering to the user the same compromise, and presents to the user just a representative case for each group.

After the retrieval, the reuse stage decides if the case solution can be reused in the current problem. In the simplest CBR-RSs, the system reuses the retrieved cases showing them to the user. In more advanced solutions, such as (Montaner, Lopez, & la Rosa, 2002) or (Ricci et al., 2003), the retrieved cases are not recommended but used to rank candidate products identified with other approaches (e.g. Ricci et al., 2003) with an interactive query management component.