Improving Case-Based Recommendations using Implicit Feedback

Deepak Khemani, Mohamed A.K. Sadiq, Rakesh Bangani, Delip Rao

Department of Computer Science and Engineering,
Indian Institute of Technology Madras,
Chennai – 600 036, India.

khemani@iitm.ac.in, {sadiq, rbangani, delip}@cse.iitm.ernet.in

Abstract

A recommender system suggests items to a user for a given query by personalizing the recommendations based on the user interests. User personalization is usually done by asking users either to rate items or specify their interests. Generally users do not like to rate items; an alternative approach would be to implicitly track user's behaviour by observing their actions. In this paper, we build a recommender system by using case-based reasoning to remember past interactions with the user. We incrementally improve the system recommendations by tracking user's behaviour. User preferences captured during each interaction with the system are used to recommend items even in case of a partial query. We demonstrate the proposed recommender system in a travel domain that adapts to different kinds of users.

Introduction

Case Based Reasoning (CBR) can be traced back to storing episodes in memory (Schank 1999; Kolodner 1993). From its more cognitive origins, CBR gradually emerged and stabilized as a form of instance based learning (Mitchell 1997). This change was spurred on by a large number of successful industrial applications (Lenz et al. 1998; Watson 1997; Leake 1996). In the course of this transformation, CBR became an approach that was focused on inexact retrieval using a notion of similarity. Thus CBR evolved as a technique in which problem-solution descriptions are stored in a case base and the notion of inexact retrieval is deployed to implement the heuristic "similar problems have similar solutions". The fact that CBR provides a methodology for storing and reusing experience has even led to the emergence of an area called Experience Management (Bergmann 2002). The notion of dynamic memory for a problem-solving agent pertains to some form of learning from experience. Learning in CBR is instance based viz. every new experience is stored as an instance. This aspect of CBR was somewhat lost when CBR was used as a retrieval mechanism in recommender systems (Burke 2000), in which given a query, the most similar product or package is retrieved from a database.

In this paper we explore how a recommender system can learn from its experience with the user and maintain usage information to anticipate the requirements of similar users in subsequent interactions. The motivation is to emulate a human agent who often makes recommendations like "This tour package to Gangotri is very popular" or "The chicken tikka here has been selling like hot cakes". We demonstrate that a system that keeps track of the values of attributes that occur in successful cases can then pick cases that are tuned to the user preferences.

In our experiments, the weights used to aggregate local similarity to a global measure are chosen arbitrarily, and yet the system orders cases such that the case selected by the user from the retrieval set is increasingly higher in rank. Furthermore, one can design systems that track different kinds of users if they are cherished clients, and anticipate their requirements. We demonstrate this by using queries of decreasing specificity and show that even with a query containing few attributes the system still retrieves cases with feature values that the users had earlier wanted.

Related Work

A recommender system suggests products or services in response to a query, based on the personal interests of the users. It helps the user to overcome the problem of information overload by providing personalized recommendations. Examples of recommender systems include recommending books, CDs, and other products at Amazon.com (Linden, Smith and York 2003), movies by MovieLens (Miller et al. 2003) and news at VERSIFI Technologies (Billsus et al. 2002). The personalization information is obtained by building user profiles describing the characteristics of the user such as age, gender, income, marital status and/or their interests. The user information is mostly collected by explicitly asking the users to either rate a (partial) set of items or select their interests in the list provided.

The existing recommendation approaches could be classified as *content-based filtering*, *collaborative filtering*, *knowledge-based approach* and *hybrid approaches* (Burke 2002). In content-based filtering approach, an item is recommended to the user by estimating the rating of the non-rated items based on the description of items rated by the user in the past. These ratings could be estimated using various machine learning techniques such as Bayesian classification, clustering, decision trees and artificial neural networks (Pazzani and Billsus 1997). In collaborative filtering approach, an item is recommended to the user by estimating the ratings of the non-rated items based on the ratings given to the items by users with similar interest. In this case, the estimation of the rating is done by using either some ad hoc heuristic rules (Delgado and Ishii 1999) or a model learnt from the underlying data using machine learning techniques (Breese, Heckerman and Kadie 1998). Knowledge-based recommender systems (Burke 2000) recommend items to a user based on the available knowledge about the user and items to satisfy user requirements. (Towle and Quinn 2000) discuss the significance of detailed explicit user and product model representation for better recommendations. Hybrid systems can be built by combining some of the approaches based on the system's requirements. A detailed survey of such systems can be found in Adomavicius and Tuzhilin (2005) and Burke (2002).

Case-based reasoning has been shown to be useful in hybrid systems. PTV, a personalized TV recommender system (O'Sullivan, Wilson and Smyth 2002) uses CBR in combination with collaborative filtering approach. In this paper we present a mechanism to learn user preferences by incorporating the experience with the user into the case-based retrieval process.

User Modeling

From the CBR point of view, we extend the notion of a case base from being a collection of <description, lesson> pairs to that of a knowledge base with <description, lesson, usage> triplets. The usage component measures the strength of a case from its utility perspective. We approximate utility by how successfully the case was used in the past. The idea is that such usage information can be used towards ossification of cases as described in (Schank 1999). In a more dynamic environment in which new solutions are proposed, they could enter the case base as tentative solutions. These solutions would be available for recall but would need to earn their spurs through successful usage. Such new candidate solutions could be a modified production plan in a manufacturing industry, a new product introduced into the market in a recommender system, or a solution found through a process of adaptation in traditional CBR. In such environments it would be meaningful to characterize cases not just by their semantics, that is applicability, but also by their demonstrated usefulness.

The proposed system keeps track of the values that occur in successful cases. This information becomes a distinct knowledge container, which can be thought of as a user model. In other words, the user model here is an episodic memory of the user's preferences, maintained as usage component. The usage component is represented in terms of the occurrence frequencies of attribute values in the retrieved cases and that in the cases selected by users. For every user, a separate user model is maintained which acts as an episodic memory that keeps track of successful usage. On the other hand, the case base of attribute-value pairs is like a semantic memory that describes the problems and associated solutions.

Personalized Retrieval

We propose the Similarity-Usage Based Retrieval (SUBR) approach, which incorporates the notion of usage into retrieval. The description d of a case can be characterized as a set of attribute-value pairs, $d = \{(a_1, v_1), (a_2, v_2), \ldots (a_n, v_n)\}$. We define $attrs(d) = \{a_1, a_2, \ldots a_n\}$ and $d(a_i) = v_i$. Similarly, a query q can also be characterized as a set of attribute-value pairs, $q = \{(a_1, v_1), (a_m, v_m), \ldots (a_p, v_p)\}$. We also define $attrs(q) = \{a_1, a_m, \ldots a_p\}$ and $q(a_i) = v_i$. Let m be the number of attributes whose values "match" with the query. This can be formally stated as follows:

$$m = \sum_{a \in attrs(d) \cap attrs(q)} \mu(d(a), q(a)) \quad (1)$$

where μ is defined as follows:

$$\mu(v_i, v_j) = \begin{cases} 1 & \textit{if } v_i \textit{ and } v_j \textit{ are nominal and } v_i = v_j \\ 1 & \textit{if } v_i \textit{ and } v_j \textit{ are cardinal and close}(v_i, v_j) \\ 0 & \textit{Otherwise} \end{cases}$$

The function *close* is defined as *close*: R × R → [0, 1]. The definition of *close* is domain dependent and returns true if two numeric values from the domain are "close" enough.

Consider a case $c \in$ CB, where CB is the case base. Let $sim(q, c)$ be the similarity of a query q with the case c. Any similarity function can be used here. We strengthen this measure with a usage factor (Khemani et al. 2005) as follows

$$sim_{usage}(q, c, \textit{user model}) = sim(q, c) \times u_c \quad (2)$$

where u_c estimates the utility of the case from past history. Let r_j be the frequency of v_j's presence in a retrieved case and s_j be the frequency of it occurring in a case selected by users. Let n be the frequency of the case c's retrieval. We define u_c as follows

$$u_c = \frac{\sum_{j=1}^{m}\left[\frac{s_j}{r_j} + (1 - \frac{r_j - s_j}{n})\right]}{m} \tag{3}$$

where, $s_j \leq r_j \leq n$. Observe that u_c lies in the range [0, 2]. Initially, the utility value for all the cases in the case base is set to neutral (such that u_c is 1) in order to avoid any bias. The utility of the case, u_c is greater than one when a case was previously selected by the user and this results in $sim_{usage} > sim$. Whenever a case was previously rejected, i.e. retrieved and not selected, u_c becomes less than one thereby reducing the sim_{usage} score. It is to be noted that the attribute values of the selected case are rewarded in terms of their usage or utility, while the attribute values of the rejected cases are penalized. Given the measure sim_{usage} (*q, c, user-model*), the SUBR procedure constructs a retrieval set, by retrieving the *k* cases with highest sim_{usage} values.

System Evaluation

The architecture of the proposed system is depicted in Fig. 1. The system consists mainly of three components namely Query Generator, Retriever and User Simulator. The case base represents the semantic memory while the user model represents the episodic memory.

Query Generator

A query generator was designed to generate a large set of queries uniformly spanning the query space (all possible queries). It can also generate partial queries, which were used for experiments. For such queries, the attributes to be present in a query were randomly chosen and their values were chosen uniformly from the attribute space.

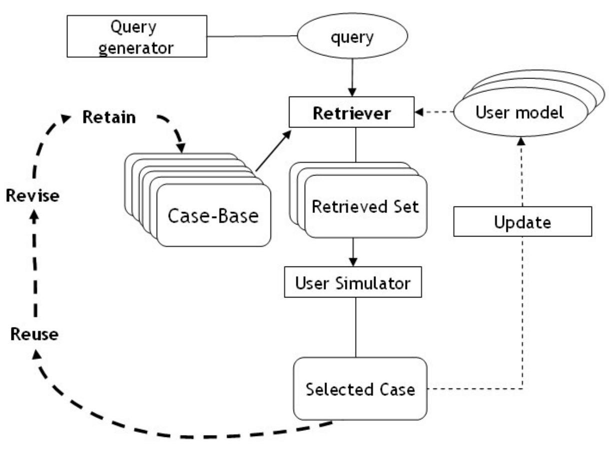

Fig. 1. System Architecture

Retriever

The retriever, as described in the previous section, retrieves cases (items) from the case base using the similarity measure and the usage information from the user model. Note that the other stages of CBR like Reuse, Revise and Retain (Aamodt and Plaza 1994) are not discussed because they are not relevant to the experiment.

User Simulator

The user simulator attempts to act as the real world user. It serves as a synthetic user to select cases for system evaluation. We maintain user preferences as probability distributions over the values of each attribute. This is collectively referred as user-preference-data. The user-preference-data is hidden from the retriever and user model. In our experiments, we handcrafted the user-preference-data for two users – a typical graduate student and a typical corporate executive. Hence the task of this simulator would be to select a case (from the retrieved set of cases) on behalf of the real user based on the probability values already set for the particular user. The case, which maximizes the joint probabilities of its matching attribute values, is selected. In the event of a tie, we resolve it by considering the joint probabilities of *all* the attribute values in those cases. This is analogous to a situation where more than one case is of our interest and we look at other characteristics of the items to make our choice. We use the corresponding user-preference-data of a user for experimental evaluation of the system.

User Model Update

The case selected by the User Simulator is taken as a feedback by the system and its usage information is updated to reflect user preferences in that case. The preferences of attribute values in the selected case are bolstered against the rejected ones.

Experimental Results

We use the case base from the travel agents domain containing 1470 cases for our experiments. Originally Lenz collected this data set for experiments in CBR (Lenz 1994). The symbolic attributes in this case base are tour type, location, season, accommodation and transport type. The numeric attributes are duration, cost and number of persons.

In the experiments reported here, we simulated two kinds of users namely the student and executive, by designing their user-preference-data over different attributes to build corresponding user simulators. For example, a student may prefer a low cost accommodation to stay in; prefer a trekking holiday; or like to travel by a train. The experiments demonstrate that the system learns the choices made during selection and personalizes the results based on the user preferences learnt.

The system was given a set of thousand queries, starting with complete queries and gradually reducing to partial queries with decreasing specificity. This was done to test the learning of user preferences even in the absence of certain attribute values. However, since the queries were distinct in nature, the performance of our system improves gradually. In case, the queries were repeated, the performance would be more like traditional CBR, retrieving the same cases after a few episodes.

The performance of our system's learning with usage is compared against unbiased system – using mere similarity measure *sim*. The size *k* of the retrieval set was restricted to ten for all the experiments discussed below. The plots show the average rank of the selected cases for the given set of thousand queries for both the systems. The figures also show error bars with 95% confidence interval. The error bars represent the variance of the ranks in the results during the ten runs of the experiments.

Fig. 2.1 plots the system's performance in learning student preferences and that of executive user in Fig. 2.2 for the same set of queries. Both the plots also show the results using the unbiased system. It can be seen that in both scenarios, the learning system progressively ranks the selected case higher than the unbiased system. The two plots demonstrate that the user model acquired from experience does result in learning the user preferences. It may also be noted that the error bars also reduced during the learning process for our system as compared to the unbiased system.

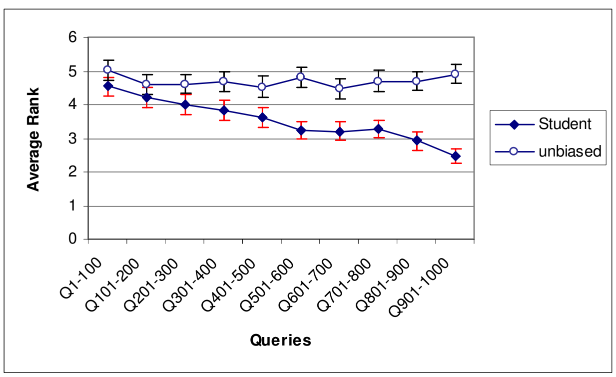

Fig. 2.1. System learning student preferences

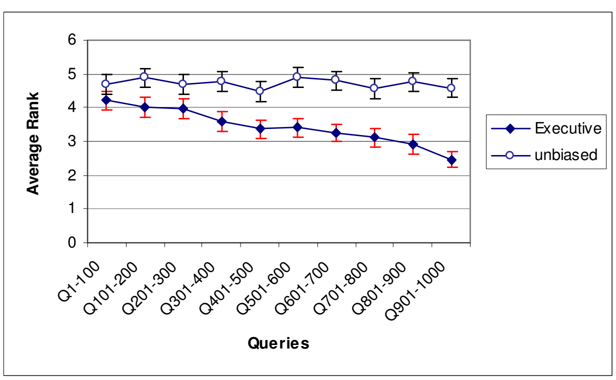

Fig. 2.2. System learning executive preferences

To demonstrate that the two user models that evolve during learning process are different, we query the system using the usage information from the student user model and select a case using the two different user simulators. Each user simulator selects a case according to its own preferences as defined in the user-preference-data but from the same retrieval set, the one tuned to the student. As expected the case selected by the executive ranked consistently lower than the case selected by the student. The two plots are compared in Fig. 3.

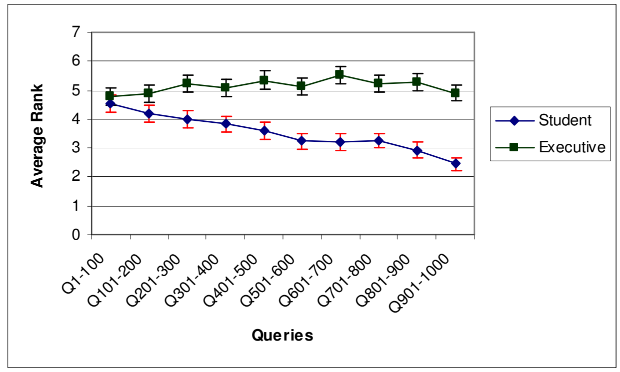

Fig. 3. Average rank of selected case by different user simulators over same retrieval set

Finally, to show that the user model that evolves for a user is significantly different from other user models, we looked for the specific case selected by the student simulator (student-best-case) from the retrieval set of the system while using the student user model, in the retrieval set of the system while using the executive user model as well as in the retrieval set of unbiased system. As expected, the student-best-case was not present most of the times in the other two retrieval sets. Fig. 4.1 shows the number of times this case was present in the three retrieval sets. By definition, it was always present in its own retrieval set but only less than half the times in the other two retrieval sets. In other words, more often than not the systems not attuned to the student did not even retrieve the student-best-case.

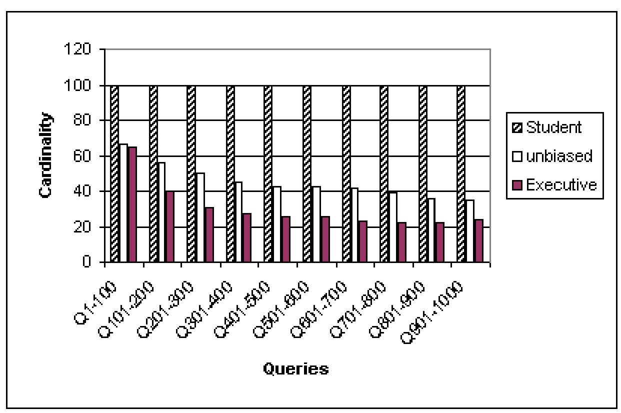

Fig. 4.1. Cardinality of presence of student-best-case in the retrieval sets

It was also observed that even when the student-best-case *was* present in the other two retrieval sets, its rank was always lower than that in the student retrieval set. Fig. 4.2

shows the average rank of the student-best-case in the three retrieved sets, whenever it was present. The student performance is the same as in earlier plots. The other two plots in the graph are ranks averaged only when the case was present in the retrieval set.

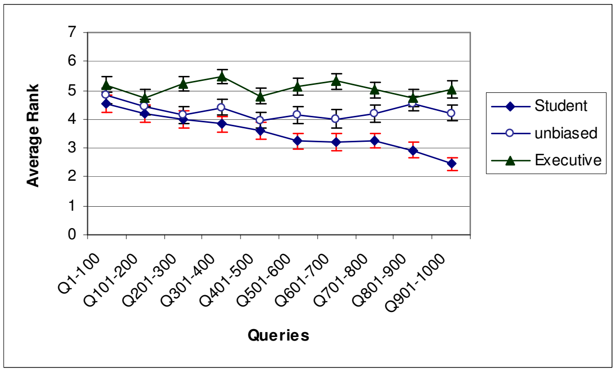

Fig. 4.2. Average rank of the student-best-case in the retrieval sets when it was present.

We conclude this section by considering sample queries from the execution. We presented two queries Q1 and Q2 to the system before and after the learning, such that Q1 is a superset of Q2. The query Q1 is a more specific query than Q2. The query Q2 specifies that eight people want to wander around for six days within a budget of $1100, while the query Q1 specifies in addition that the above holiday should be during May and they would prefer a 3-star hotel. Before learning, different cases are selected for queries Q1 and Q2 using student's preferences as shown in Table 1. After learning, the queries Q1 and Q2 both result in the same case being selected, as shown in Table 2. This case is different from the ones selected before training and is also ranked high. It should also be noted that after learning, even though Q2 was less precise than Q1, the system was able to recall what the user desired. This demonstrates that the system already had an idea of what the user wanted and a detailed query does not make a difference. The description of the cases selected is shown in Table 3.

Conclusions and Future Work

We demonstrated that by using the feedback of cases selected by users from retrieval sets, a recommender system could learn to anticipate the cases that a user or a group of users may want. The system does this by keeping track of the values of the attributes that occur in the successful cases. The similarity based retrieval method is augmented to use this statistical information to induce a new ordering on the cases where the cases with more frequently occurring values are placed higher. Observe that this value learning is different from weight learning that has been reported (Aha and Wettschereck 1997).

In the experiments it is shown that when a case base accrues usage information for a user, then it tends to pick those cases, the user had preferred earlier. In an extreme situation even with hardly any information in the query, such a system can pick a case that would satisfy the user. This is akin to the situation when you walk into your local pub, and the barman comes up to you with "The usual?", without you having specified anything. The system demonstrated here uses frequency information of values of attributes that occur in successful cases. It does not however keep track of combinations of preferred values. The above work assumes that the user preferences are static. It would also be interesting to design systems that adapt to changing user preferences.

Table 1. Sample query execution before learning

Attributes	Tour Type	No. of Days	Cost	No. of Persons	Month	Accommo dation	Selected Case ID	Rank
Query Q1	Wandering	6	$1100	8	May	3-star	105	2
Query Q2	Wandering	6	$1100	8			84	7

Table 2. Sample query execution after learning

Attributes	Tour Type	No. of Days	Cost	No. of Persons	Month	Accommo dation	Selected Case ID	Rank
Query Q1	Wandering	6	$1100	8	May	3-star	435	2
Query Q2	Wandering	6	$1100	8			435	2

Table 3. Description of the selected cases

ATTRIBUTES	TOUR TYPE	NO. OF DAYS	COST	NO. OF PERSONS	LOCATION	MONTH	ACCOMMODATION	TRANSPORT TYPE
CASE ID 105	WANDERING	7	$1172	3	HARZ	MAY	3-STAR	CAR
CASE ID 84	WANDERING	7	$944	8	BAVARIA	AUGUST	FLAT	CAR
CASE ID 435	WANDERING	14	$1008	2	CARINTHIA	MAY	2-STAR	CAR

References

Aamodt, A., and Plaza, E. 1994. "Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches". *AI-Communications* 7(1), pp. 39-52.

Adomavicius, G., and Tuzhilin, A. 2005. "Towards 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*, vol. 17, no. 6, pp. 734-749.

Aha, D. W., and Wettschereck, D. 1997. "Case-based learning: Beyond classification of feature vectors", In *Proceedings of the Ninth European Conference on Machine Learning*, pp. 329-336.

Bergmann, R. 2002. "Experience Management: Foundations, Development Methodology, and Internet-Based Applications", *Lecture Notes in Artificial Intelligence*, Vol. 2432, Springer Verlag.

Billsus, D.; Brunk, C.A.; Evans, C.; Gladish, B., and Pazzani M. 2002. "Adaptive Interfaces for Ubiquitous Web Access", In *Communications of ACM*, vol. 45, no. 5, pp. 34-38.

Breese, J. S.; Heckerman, D.; and Kadie, C. 1998. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", In *Proceedings of 14th Conference on Uncertainty in Artificial Intelligence*, pp. 43-52.

Burke, R. 2000. "Knowledge-based Recommender Systems. In A. Kent (ed.), *Encyclopedia of Library and Information Systems*. Vol. 69, Supplement 32. New York: Marcel Dekker.

Burke, R. 2002. "Hybrid recommender systems: Survey and experiments". *User Modeling and User Adapted Interaction*, pp. 331-370.

Delgado, J., and Ishii, N. 1999. "Memory-Based Weighted-Majority Prediction for Recommender Systems", In *Proceedings of ACM SIGIR Workshop Recommender Systems: Algorithms and Evaluation*.

Khemani D., Sadiq M., Bangani R., and Rao D. 2005, "What You Get is What You Wanted", poster at *13th International Conference on ADCOM*, India.

Kolodner, J. 1993. "Case Based Reasoning", *Morgan Kaufmann, San Mateo*.

Leake, D. 1996. "Case Based Reasoning: Experiences, Lessons, and Future Directions", *AAAI Press/MIT Press, Menlo Park*.

Lenz, M. 1994. "Case-Based Reasoning for Holiday Planning", (Eds.) Schertler, W., Schmid, B., Tjoa, H. Werthner, In *Information and Communications Technologies in Tourism*, pp. 126-132, Springer Verlag.

Lenz, M.; Bartsch-Sporl, B.; Burkhard, H.D.; and Wess, S. (Eds.) 1998. "Case Based Reasoning Technology", *Lecture Notes in Artificial Intelligence* 1400, Springer Verlag.

Linden, G.; Smith, B.; and York, J. 2003, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering", In *IEEE Internet Computing*, pp. 76-80.

Mitchell, T., 1997. "Machine Learning", *McGraw Hill*.

Miller, B.N.; Albert, I.; Lam, S.K.; Konstan, J.A.; and Riedl, J., 2003. "MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System", In *Proceedings of International Conference Intelligent User Interfaces*.

O'Sullivan, D.; Wilson, D.C.; and Smyth, B., 2002. "Improving Case-Based Recommendation: A Collaborative Filtering Approach", In *Proceedings of European Conference on Case-Based Reasoning*, pp. 278-291.

Pazzani, M., and Billsus, D., 1997. "Learning and Revising User Profiles: The Identification of Interesting Web Sites", *Machine Learning*, vol. 27, pp. 313-331.

Schank, R. 1999. "Dynamic Memory Revisited", *Cambridge University Press*.

Towle, B. and Quinn, C. 2000, "Knowledge Based Recommender Systems Using Explicit User Models", *In Knowledge-Based Electronic Markets, Papers from the AAAI Workshop*, pp. 74-77.

Watson, I. 1997. "Applying Case-Based Reasoning; Techniques for Enterprise Systems", *Morgan Kaufmann Publishers*.