AUTOMATED PRODUCT RECOMMENDATION BY EMPLOYING CASE-BASED REASONING AGENTS

Ozgur Baykal[a] Reda Alhajj[b] Faruk Polat[a]
[a]*Dept of Computer Engineering, Middle East Technical University, Ankara, Turkey,*
[b]*ASDA Lab, Dept of Computer Science, University of Calgary, Calgary, AB, Canada,*

Keywords: Multiagent Systems, Case-Based Reasoning, E-Commerce, Product Recommendation.

Abstract: This paper proposes a cooperation framework for multiple role-based case-based reasoning (CBR) agents to handle the product recommendation problem for e-commerce applications. Each agent has different case structure with intersecting features and agents exploit all information related to the problem by cooperation, which is accomplished through the merge of distributed cases. The role-based CBR agents merge the distributed cases by introducing a global heuristic function, which exploits the relevancy of each merged case within the viewpoint of each agent and the satisfied/unsatisfied problem constraints. Finally, the proposed framework has been tested for elective course recommendation.

1 INTRODUCTION

In this paper, we propose a cooperative method that improves solution quality of CBR agents by the cooperative retrieval of distributed cases. An agent constitutes an autonomous node of the respective distributed case-base. The cooperation of agents is accomplished by merging distributed cases to form *overall cases* having better representation of the problem. This way, a problem is cooperatively solved by a society of autonomous agents in order to improve solution quality (Sycara 1998). Further, the cooperation of agents results in having each agent uses its expertise to retrieve distributed cases and to evaluate the overall cases that will be used in the solution. They negotiate on possible results of merging distributed cases to locate erroneous ones according to a *heuristic function*, which is used for directing the result of merge in increasing the relevancy of overall cases and for handling noisy distributed case-bases. So, the most salient property of the work described in this paper is the merge of distributed cases by autonomous agents guided by heuristic functions in order to introduce efficient evaluation of the resulting overall cases by considering the limited viewpoint of each agent, and to provide a robust way of merging distributed case-bases with noisy cases, by negotiation between agents over the conflicting feature values. The rest of the paper is organized as follows. The proposed multiagent CBR framework is described in Section 2. Experimental results for elective course recommendation are discussed in Section 3. Section 4 includes summary and conclusions.

2 PROPOSED COOPERATION FRAMEWORK

The overall case structure consists of the following values: 1) *feature-agent(s) tuple*, denoting agents related to a problem feature; it holds the values of problem features; 2) set of *problem constraints*, includes constraints of the current problem satisfied and unsatisfied by the overall case; 3) *value of information* of the overall case within the limited viewpoint of each agent; 4) *value of coherency* of the overall case, denoting the consistency of a component within the view point of the relevant agent. Finally, the relevance of the overall cases can be determined from the value of information, value of consistency, and the satisfied constraints.

The overall cases are stored in an *overall case-base* (OCB) shared by all agents. Every time a problem is presented, a new OCB is generated to be used by the agent working on the problem for a global case-base in the context of the current problem. Since the structure and platform of the running agents are mostly different, OCB has been implemented as an agent with communication capabilities. It is controlled by an agent called OCB-interface.

Consider an agent a_i and let e_{a_i} be its *evaluation*

Baykal O., Alhajj R. and Polat F. (2004).
AUTOMATED PRODUCT RECOMMENDATION BY EMPLOYING CASE-BASED REASONING AGENTS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 515-518
DOI: 10.5220/0002620105150518

function; α_{a_i} be the set of all feature-value pairs of overall case α that can be evaluated within the viewpoint of a_i, i.e., the values of features in the agent's local case structure; and g_{a_i} be a distance function defined over distributed cases of agent a_i. An evaluation function should be fast and simple because it will be used after each merge on the overall case. Distance functions are percentage functions defined via the ratio between the values of features of a distributed case; they are simple and fast.

Satisfied constraints have the value 1, and unsatisfied constraints have the value 0. The semantics of this heuristic function are used in constructing overall cases. The overall value shows the relevancy of an overall case to the problem. The overall value of a case is the evaluation of the following points within the viewpoint of the system: 1) the solution quality within the viewpoint of each agent (value of information); 2) the percentage of the change of the originating distributed case by the sequence of merge operations (value of consistency); 3) the set of satisfied/unsatisfied constraints. The overall evaluation function of the agent working on the problem is transmitted to other agents as expressions. This way, agents share the overall evaluation function in order to evaluate overall cases.

Each agent uses its limited viewpoint to retrieve from its case-base the most promising local cases to be used as initial seeds in initiating the merge of distributed cases. When all the distributed cases are not used as initial seeds, the combination of initial seeds with the distributed cases may not lead to the best overall cases. However, initiating all cases will significantly decrease the response time of the application when the number of cases is very high.

Retrieved initial seeds are inserted into the overall case base. The value of the information features of the overall case are requested from the relevant agents. The request is only possible when more than one agent share the same case-structure. Since each agent is capable of solving the problem individually, they manipulate the constraints available in their viewpoints for checking. So, the satisfied/unsatisfied constraints of the initial seeds are set by the agent retrieving the initial seeds.

Each agent selects some local cases per overall case using its selection function. Selection is a unification of the set of intersecting features of an overall case and the agent's local case as well as differing features of agent's problem and the overall case. When distributed cases are noisy or have missing parts, the merge operation results in inconsistent cases.

Each possible result of merge, called *temporary overall case*, is stored in the agent's temporary overall case table, denoted temp-OC. Constraints of the temporary overall cases are checked by the agent performing the merge. Then each relevant agent is informed about the changes of the temporary overall case for the evaluation of the value of information and value of consistency. Selecting the temporary overall case with the highest evaluation value for the result of the merge makes the search climbs through hills to some local minima. The selected merge result is only inserted in the overall case-base when the overall evaluation value of the selected overall case is higher than the overall evaluation value of the unmerged overall case (*merge if promising* principle). Therefore, merge of overall cases and distributed cases may not lead to complete cases.

The revision of cooperative retrieval can be accomplished by the revision done in the retrieval of initial seeds and the selection of distributed cases for merge. One of the possible ways of accomplishing the revision is restarting the cooperative retrieval by removing the distributed components of the overall cases which are used by selected distributed cases for merge and by retrieved distributed cases as initial seeds.

The overall case-base is implemented using OCB-interface module, which is instantiated by the agent that presents the problem. An overall case-base is generated per active problem using the OCB-interface by the agent to whom the problem has been submitted. It has communication capabilities with the agents and has own control over the overall cases in order to provide safe computations on them.

An agent holds the problem it is working on, a list of registered agents, a message queue for storing messages, retrieved cases used for initiating merge operations, a temp-OC table for holding temporary results of merge operations it is currently working on, and selected cases for the possible merges of an overall case with distributed case-base. Since more than one problem is to be solved by the system, the problems are distinguished with a unique label referring to the agent to whom the problem is presented. The list of registered agents enables the communication with the other agents, and each agent needs to register itself in order to communicate using messages. The message queue provides asynchronous computations of the agents, such that they can continue their execution after placing their messages into the queue.

Except announcing the problem and the overall evaluation function, most of the time agents communicate with each other in order to request information or as a response to a request. The information transmitted between agents is the value of information and value of coherency of the overall cases. If a change on the features that are related with agents occurs after the merge operation, agents request the value of information and value of coherency through transmitting relevant distributed components. Agents check constraints using their available information, since all distributed components of an overall case are checked for satisfaction by the relevant agents. These processes need to be synchronized because overall cases are merged with selected distributed cases and distributed cases are selected per overall case.

The information passing mechanism in the system is peer-to-peer, although the OCB-interface is considered as a shared memory for holding overall cases. Since all overall cases are maintained by OCB-interface, it becomes bottleneck in the communication infrastructure. Despite that OCB-interface controls overall cases without any overhead, it proposes no solution in the case of failure.

The messages used in the system for communication are *KQML* messages (Finin, McKay and Fritson 1992). This way, all the information related with the content of the message and with ontology of the problem could be sent along with appropriate message format. KQML provides separation of semantics of the message from the communication protocol. Therefore, the use of KQML in the system enables different structured agents to cooperate.

3 EXPERIMENTS

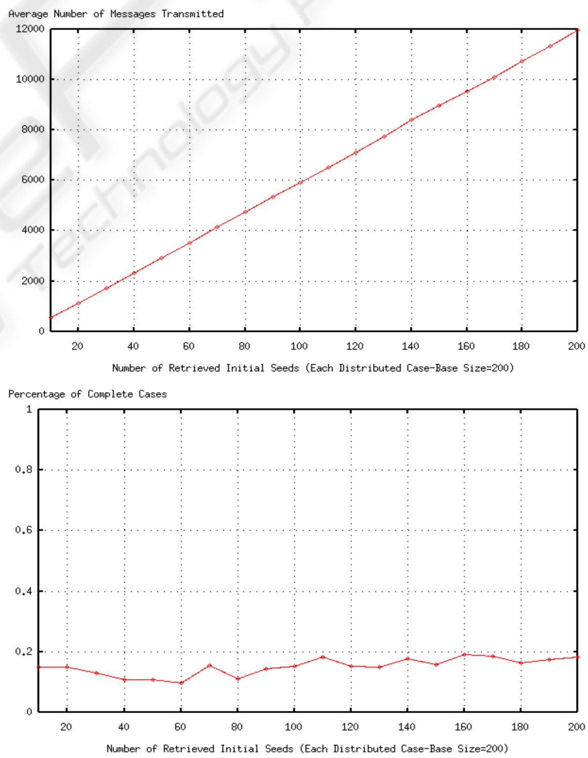
The *elective course recommendation* problem has been selected for evaluating the proposed framework. The environment on which agents operate is a subset of the Student Affairs database of Middle East Technical University. Experimental results and the elective recommendation problem with hypothetically generated cases are presented in relation with the product recommendation problem.

The elective course recommendation problem is a classification problem. The superset of the set of offered courses and the set of all students constitute the domain of the problem, and the set of courses that can be taken as electives constitutes the range of the problem. There are also inter-dependencies between problem features that contribute to the potential solution; these are cumulative grade point average (CGPA), list of previously taken elective courses, and ÖSS[1] scores of the student. The experimental system consists of three agents, namely CGPA-Agent, ELIST-Agent and ÖSS-Agent. Each agent controls the cases determined by its role at the respective distributed node. The distributed cases of each agent are constructed by the concrete enrollments of a student. Note that distributed cases may be redundant, and redundant cases are subsumed by the more powerful cases (Yang and Racine 1998).

A student can take an elective course according to the preferences of CGPA, previously selected courses, and ÖSS scores using CGPA-Agent, ELIST-Agent and ÖSS-Agent, respectively. One of the preferences determines the agent to solve the problem, and the selected agent completes the following processes to solve the problem using CBR: 1) cooperative retrieval of its cases in order to exploit all problem relevant information; 2) adaptation of selected case(s) of cooperatively retrieved cases to generate an answer; 3) revision of cooperative retrieval in case of failure.

We tested the number of messages and the percentage of complete cases with respect to the change in distributed case-base size and number of retrieved cases for 3-agent scenario. Each time, the obtained experiment results are averaged over randomly selected 100 problems, and one distributed case is selected with each overall case by each agent. The following heuristic function h is used in the experiments, where n is the number of agents in the system.

$$h(VI_0, VC_0, ..., VI_n, VC_n) = \sum_{i=0}^{n} (VI_i * VC_i)$$

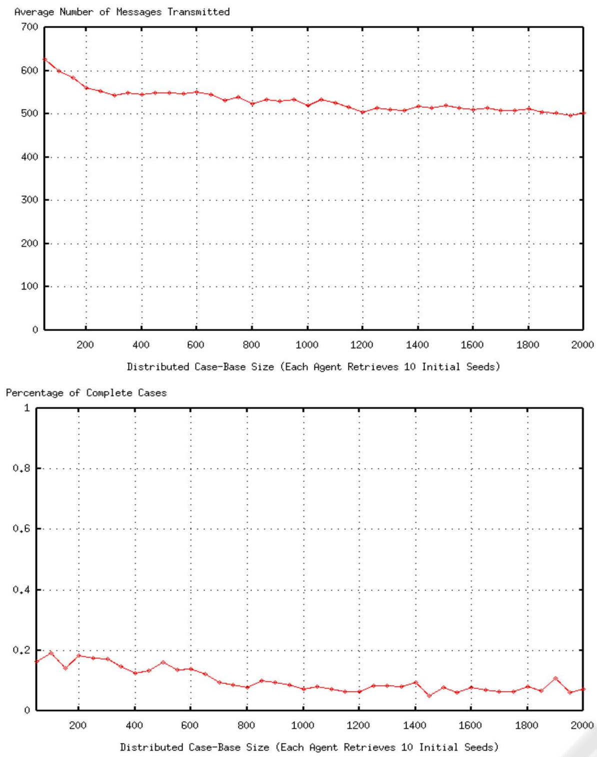


(a) Number of messages (b) Percentage of complete cases

Figure 1: 3-Agent Scenario with increasing number of retrieved initial seeds

The results for increasing number of retrieved initial seeds and 200 distributed case-base size are given in Figure 1. Figure 1-a shows the average number

[1] ÖSS is a university entrance exam that measures verbal and analytic skills of each high school student in Turkey.



(a) Number of messages (b) Percentage of complete cases

Figure 2: 3-Agent Scenario with increasing size of distributed case-bases

of messages transmitted between the agents with respect to the number of initial seeds. From Figure 1-a, it can be easily seen that the ratio of the number of messages transmitted to the number of retrieved initial seeds is constant. Figure 1-b shows the percentage of complete cases with respect to the number of initial seeds. The percentage of complete cases decreases in Figure 1-b because the more distributed cases are used, it is more probable to have distributed cases with higher relevance, so that the possible result of the merge operation mostly has lower heuristic value. In other words, the result of the merge operation has lower heuristic value most of the time.

The results for increasing size of distributed case-bases and 10 retrieved initial seeds are given in Figure 2. Figure 2-a shows the average number of messages transmitted between the agents with respect to the size of distributed case-bases. It can be easily seen that the number of messages transmitted tend to decrease with increasing number of messages. Figure 2-b shows the percentage of complete cases with respect to the size of distributed case-bases. According to Figure 2-b, the percentage of complete cases decreases slightly with the increasing size of distributed cases. Note that agents can exploit subset of the problem relevant information for partial cases because agents can solve problems for non-complete cases.

It has been observed from the experiments that when the size of distributed cases is huge, the number of messages is quite admissible with respect to the size of distributed case-base. Therefore, the proposed framework can be efficiently used with physically distributed case-bases. Finally, the percentage of complete cases is low because they are not used in the generation of distributed cases, and the distribution of the evaluation function (for evaluating the value of information) and distance function (for evaluating the value of consistency) of each agent are different.

4 SUMMARY AND CONCLUSIONS

The proposed approach is simple and provides computationally efficient cooperation framework for retrieval of distributed case-bases. It is applicable to systems having small number of agents in which the distributed case-bases are physically distributed and in which the number of retrieved initial seeds and the number of selected distributed cases for merge are small. Since role-based CBR agents have limited description of the problem, they need to cooperate in order to improve the solution quality. The retrieval from distributed case-bases results in sub-optimal retrievals because of the fact that the construction of overall cases highly depends on the initiated cases and the progressing heuristic values of the merged overall cases. The use of heuristics in the merge of distributed cases makes the search as an application of hill climbing search in broadening hyper-planes. The merge of distributed cases handles noisy distributed cases by negotiation of agents on the possible results of merge. Finally, the proposed cooperation framework can be extended with cooperative application of adaptation functions within the multiagent framework.

REFERENCES

Finin T., McKay D. and Fritson R., "An overview of KQML: A Knowledge Query and Manipulation Language," *Technical Report,* University of Maryland, 1992.

Yang Q. and Racine K., "Redundancy and Consistency Detection in Large and Semi-Structured Case-Bases," *Technical Report,* 1998.

Sycara K., "Multiagent Systems," *AI Magazine,* Vol.19, Vol.2, 1998.

Wooldridge M., "Intelligent Agents," *in Multiagent Systems,* Weiss G. (Ed.), MIT Press, 1999.