A Multiagent System using Associate Rule Mining (ARM), a collaborative filtering approach

Prakash Ranganathan, Juan Li, Kendall Nygard
Department of Computer Science
North Dakota State University, Fargo, ND 58102
email:prakashranganathan@mail.und.edu

Abstract—Agent Oriented Programming (AOP) is a recent promising software paradigm that brings concepts from the theories of artificial intelligence into the mainstream realm of distributed systems, and yet it is rather difficult to find a successful application of agent oriented system (specifically) when large-scale systems are considered. When adopting an agent-oriented approach to solve a problem, there are a number of domain independent issues that must always be solved, such as how to model agent behavior to predict future action and how to allow agents to communicate rather than expecting developers to develop this core infrastructure themselves. In our paper, we address several problems that exist in a socialized e-learning environment and provide solutions to these problems through smart and collaborative agent behavior modeling which learn and adapt themselves through prior experiences, thereby assisting in successful implementation of this large scale e-learning system. In this paper, the author (s) proposes an implementation of a complete distributed e-learning system based on Collaborative filtering (CF) method. The system has intelligent collaborative filtering based tutoring system (ICFTS) capabilities that allow contents, presentation and navigation to be adapted according to the learner's requirements. In order to achieve that development, two concepts were put together: multi-agent systems and data mining techniques (specifically, the ARM algorithm). All the implementation code is developed using MATLAB GUI environment.

To our best knowledge, very few literatures discusses a portion of e-learning environment using adaptive software agents, but none of the current literatures addresses a complete implementation of their learning system in detail. The goal of the paper is to implement one such multi-agent based e-learning system which learns from its prior user experiences on top of an agent-oriented middleware that provides the domain-independent infrastructure, allowing the developers to focus on building the key logic behind it. In this system,
the agents follow an adaptive cognitive learning approach, where the agent learns through user
behaviors via a collaborative filtering technique, or experiencing and then processing and remembering the information in an e-learning environment. The paper will utilize agent (a piece of code) based environment in our e-learning system using ARM [1][2]. The paper follows a learning approach based cognitive domain of Bloom's Taxonomy such as Analyze, Evaluate, Create, Apply, understand and remember.

Keywords: Collaborative filtering, intelligent agents

I. INTRODUCTION

The concept and practice of distance education has evolved tremendously at numerous educational institutions and are growing exponentially in recent years around the world and yet there is not much research done to employ adaptive learning agents or algorithms which can predict the behavior of student users to make their learning process easier in an e-learning environment. Current e-learning system in literatures does not cite any use of rigorous adaptive software agents in detail to study the behavior of student users in promoting their knowledge base or assist in their e-learning environment. Effective e-learning systems should include advanced functions with adaptive learning methods and their interface should hide their complexity to learners, providing an easy interaction grasping the student's interest. Despite of this, we often find a mere electronic transposition of traditional material, provided through rigid interaction schemes and awkward interfaces.

In our paper we follow a Market basket approach, but here the customers are students in e-learning environment and the products they shop are the query they visit in an online interface, So, we call our approach as Query Basket Analysis, as seen in Fig.1 below. Frequent item set mining leads to the discovery of associations and correlations among items in large transactional or relational data sets. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases. The discovery of interesting correlation relationships among huge amounts of business (query) transaction records can help in many e-learning (business) decision making processes, such as course design, cross marketing, and student shopping-behavior-analysis.

In our paper, we present an agent oriented e-learning framework employing adaptive software agents, which learns from user behavior (via a participation of rating scheme) who share similar experience in their learning using collaborative filtering technique [5][6]. In our method, the student users rates the difficulty or completion level of the questions in a course using a rating scale ranging from 0 to 5 with 0 (1 or 2 or 3 or 4) being higher difficulty level in understanding the problem signifying the student need more attention or help in solving the problem further and 5 being a rating where student user can easily or moderately understand the problem stated.

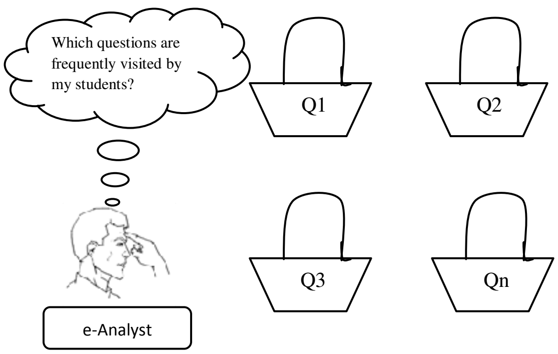

Fig.1 Query (Market) Basket Analysis

We focus on a 5 star rating scheme in our problem to filter out students who are not having trouble in solving the problem. These students experience are shared to determine and predict whether the next student will complete the query in the course interface. Then, a recommendation agent is deployed to the student interface for student users who are unable to complete the question, by providing more help to the student associated with the particular problem that he or she having difficulty in understanding the query or the problem. The recommendation agent deploys various e-assist tools on the problem to the student agent, such as an e-help window on the particular problem, an instructor – student chat interface to further assist the student [7] seen in Fig 2. Thus provide a high degree of adaptive intelligent learning mechanism according to their logged history of attended student user queries.

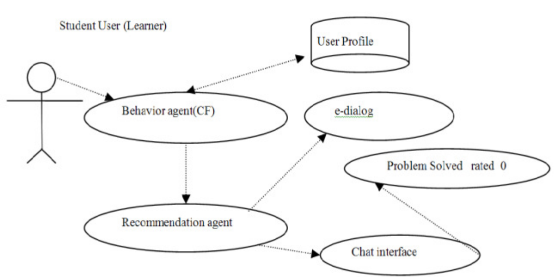

Fig.2 Use-case diagram of a Student User

Our collaborative filtering algorithm uses a collection of student user profiles to identify "interesting information" for these users. A particular student user gets a recommendation based on the user profiles of other, similar users. User profiles are commonly obtained by explicitly asking users to rate the items. Collaborative filtering has often been formulated as a
Self-contained problem, apart from the classic information retrieval problem (i.e. ad hoc text retrieval). The rest of the paper is divided into following sections: GUI Introduction and Learning Interface, Agent behavior model, Query Association Rule and Conclusion

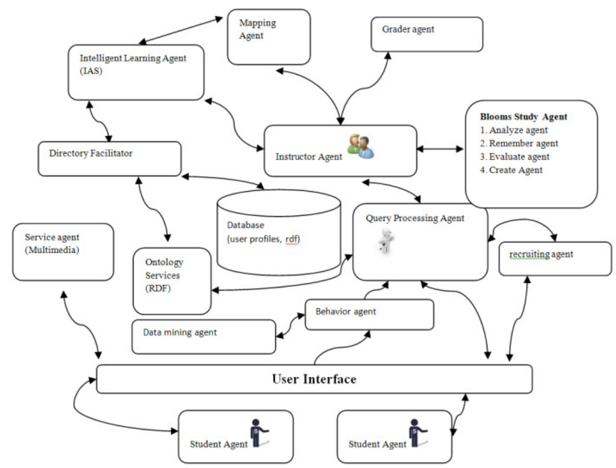

Fig.3 A generic e-learning system architecture

II. E-LEARNING INTERFACE

Our MATLAB based GUI interface has question set on the right side relating to a course and five agents at the entry login phase such as Instructor Agent, Behavior or Recommendation Agent, Prospective Student Agent, Directory agent, Grader agent and Service agent. We will primarily focus on Recommendation agent employing Collaborative filtering method to study the behavior of the users in the system.

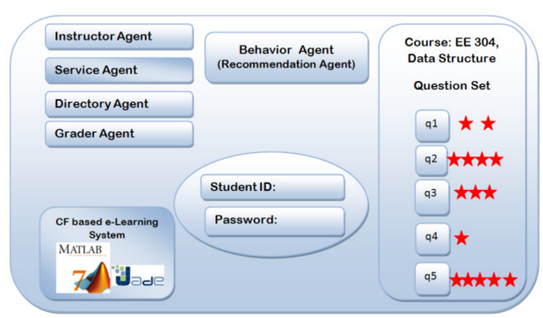

Fig.4 A model of MATLAB/JADE Interface representing multi-agent e-learning System

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications to support interoperability amongst agents and agent-based systems. JADE is a FIPA compliant agent platform and development framework. Agent platforms are responsible for dealing with agent services such as messaging (transport, encoding and parsing), scheduling, agent lifecycle management and other common resources. JADE agents are executed in a container where they share these services with other agents present in the container. Our main interest in the JADE platform is to build a adaptive learning agent and implement an multi-agent oriented framework[3][4].

III. III. AGENT BEHAVIOR MODEL USING COLLABORATIVE FILTERING (CF)

Collaborative-filtering algorithms aim to identify users that have relevant interests and preferences by calculating similarities and dissimilarities between user profiles (Herlocker et al., 2004) [9]. The idea behind this method is that, it may be of benefit to one's search for information to consult the behavior of other users who share the same or relevant interests and whose opinion can be trusted. So, our collaborative filtering technique assumes the following: 1. Student Users who were behaved (rated) similar in the past are likely to be behaving similar in the future. 2. Use similar users' behaviors to make recommendations [4].3. All student user must have a fixed timeframe (*say 15 minutes*) in responding to each question on the agent interface.4. All e-assist tools are deployed ahead (*say 3 minutes*) before the closing of each query session.5. We assume a predefined behavior through a behavior rating Table containing 4 users and 5 sample queries in a course.

The purpose of our e-learning system is to learn the behavior of each student user through their responses to a set number of questions and adapt to offer a personalized solution through agent oriented programming automatically, if the student had problem in understanding the question. We decide to use association rule mining (ARM), since it discovers associations between sets of virtual agent communities that are shared across many student users [2] [9]. By virtual community we mean "a group of agents who share characteristics and interact in essence or effect only". In other words, people in a virtual community influence each other *as though* they interacted but they *do not interact.* To carry out this agent based adaptive learning task, we developed the e-learning interface using MATLAB GUI environment. Our e-learning system assumes a pre-defined set of questions (five questions, in our example) pertaining to a problem in a course. For our simplicity, we have adapted a Data Structure course in C++ programming environment. Our paper describes an ARM implementation in MATLAB GUI/JADE using CF.

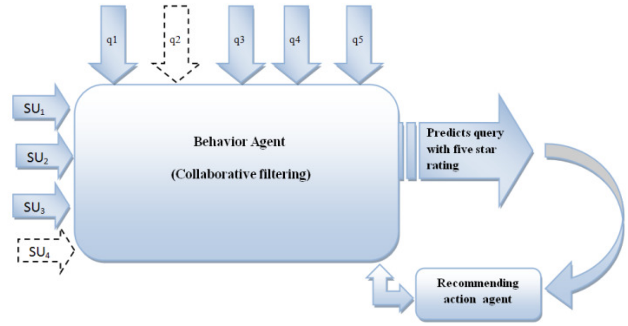

Fig.5 Behavior Prediction using Collaborative filtering

A. Association Rule Mining (ARM)

Association rule mining consists of first finding frequent item sets (set of items such as A and B, satisfying a minimum support threshold, or percentage of the task-relevant tuples) from which strong association rules in the form of A=>B are generated. These rules also satisfy a minimum confidence threshold (a pre-specified probability of satisfying B under the condition that A is satisfied). Associations can then be further analyzed to uncover correlation rules, which convey statistical correlations between item sets A and B. Recommendation agents need to employ efficient prediction algorithms so as to provide accurate recommendations to users. If a prediction is defined as a value that expresses the predicted likelihood that a user will ''like'' an item, then a recommendation is defined as the list of n items with respect to the top-n predictions from the set of items available. Improved prediction algorithms indicate better recommendations. This explains the essentiality of exploring and understanding the broad characteristics and potentials of prediction algorithms and the reason why this work concentrates on this research direction. In our paper, we use the ARM based CF technique to predict the student user behavior and make recommendation to support a student or not in their learning process.

IV. IV. QUERY ASSOCIATION RULE (QAR), SUPPORT AND CONFIDENCE RULE

We define some definitions for Associative Rule Mining on Student Collaborative Filtering (SCF) data, such as the entities U (i.e., the set of *Student Users*) and q (i.e., the set of queries or questions of multiple choice type in a course), and for any subset of rating values, $R \subset \{0,1,2,3,$

4, 5}, a R refers to a rating relationship relating user, u, to question q, iff u rates q with a rating from R, e.g., for R={0,1,2,3,4,5} in RT listed below. If a user rates to a question, then he has attended and successfully responded to the question and it is assigned a 1 in the Routing Table or else a zero as shown in the routing table. It is important to note that a 0 in the Routing Table implies that the student user either not visited this question on the GUI interface or visited, but failed to complete successfully or the user may have rated 1, 2, 3 or 4 rating for the query. Let A, be a set of questions, then the ***User set*** of A with respect to R, $Uset_R(A)$, is the set of users, u, such that $\forall a \in A$, u Rrates a [$(Uset_R(\{q_1,q_2\})= \{u_2,u_4\})$]. Similarly, the ***User count*** of A w.r.t R, $Ucount_R(A)$, is the count of u such that $\forall a \in A$, u Rrates a [$(Ucount_R(\{q_1,q_2\})=|\{u_2,u_4\}|=2)$]. ***User ratio*** of A w.r.t R, $Uratio_R(A)$, is the ratio of u such that $\forall a \in A$, u Rrates a. [$(Uratio_R\{q_1,q_2\}=Ucount_R\{q_1,q_2\}/4=2/4)$].

TABLE 1: BEHAVIOR ROUTING TABLE

u	**ratings for $q \in Qset_R(u)$**
U1	r(u1,q1)
U2	r(u2,q1) r(u2,q2) r(u2,q4)
U3	r(u3,q1) r(u3,q3)
U4	r(u4,q1) r(u4,q2) r(u4,q4)

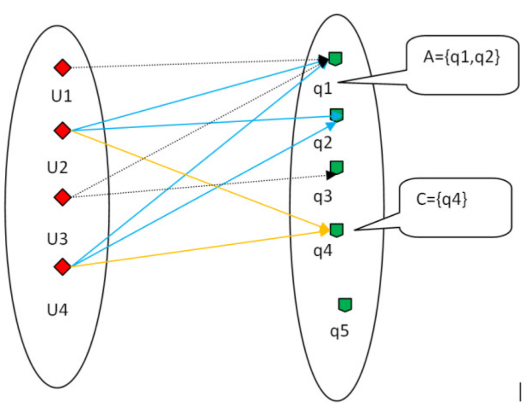

Fig.6

Routing Table Graph

We develop a new Query Association Rule (a QAR), where $A \rightarrow C$, associates two disjoint sets A and $C \subset Q$. We form new set A called the *antecedent*, because it is listed first in the RT graph and a set C, the *consequent*, as it listed last in RT graph.

	q1	q2	q3	q4	q5	U_i
U_1	1	0	0	0	0	**1**
U_2	1	1	0	1	0	**3**
U_3	1	0	1	0	0	**2**
U_4	1	**?**	0	1	0	**3**
m_j	**4**	**2**	**1**	**2**	**0**	

User rates 5* , System assigns a 1 in the BRT to represent query completed
User rates 0,1,2,3 or 4 - query not visited/ not completed or having difficulty, the system assigns 0 for all of the above ratings

Fig. 7 (a and b): User- Query Behavior Ratings Table (BRT)

Goal*: Predict the behavior of student user SU4 in visiting query q2 (i.e., rate) in the JADE/GUI interface?*

Notes on Fig 7.a: 1 means that the user has chosen this query and completed the query successfully in the JADE/GUI interface and a 0 means that the user has not chosen this query or not completed the question successfully in the JADE/GUI interface

We mainly rely on two factors to make a predictive decision using the minimum Support ($Support_{mini}$) and Confidence rule to understand the behavior of student learners in the JADE interface. The ***User set*** [count] {ratio=support} of QAR, $A \rightarrow C$, w.r.t R can be written as, $Uset_R(A \rightarrow C)$ [$Ucount_R(A \rightarrow C)$] or {$Uratio_R(A \rightarrow C)$}, is that of $A \cup C$. Note: In ARM, $Uratio_R(A \rightarrow C)$ is called "support" of $A \rightarrow C$, $Usup_R(A \rightarrow C)$. The ***User confidence*** of a QAR, $A \rightarrow C$, w.r.t R, $Uconf_R(A \rightarrow C)$ can be thought as $Usup_R(A \cup C)/Usup_R(A)$ or $count_R(A \cup C)/Ucount_R(A)$. ARM Data Miners typically want to find all *Rstrong Rules, $A \rightarrow C$,* i.e., those with $Usup_R(A \rightarrow C) > Support_{mini}$ and $Ucon_R(A \rightarrow C) > Confidence_{mini}$ (i.e. $Qsup_R(A \rightarrow C) > Support_{mini}$ $Qsup_R(A \rightarrow C) > Confidence_{mini}$) with A and C form a Rstrong Correlation, $A \leftrightarrow C$, if both $A \rightarrow C$ and $C \rightarrow A$ are Rstrong rules. So for a particular q, look for n such that $Ucount_R(m,n) > Ucount_R(n)^* \ Confidence_{mini}$. It is to be noted that

$Ucount_R(n) > Ucount_R(m,n)$ at always. So for the example at right, let's do ARM to predict $(u,q) = (u_4,q_2)$.

We look for $q \neq n \in Qset_R(u_4)$ such that $Ucount(q_2,n)>Ucount(n) * (7/8)$. We find the Ucount's values are taken with u_4 removed, so we get a user chosen number for $Confidence_{mini} = 7/8$.

Using the condition: Ucount(q_2,n)>Ucount(n) * (7/8)

$n=q_1$,	1	>	3	*	(7/8)	FALSE
$n=q_4$,	1	>	1	*	(7/8)	TRUE

Fig.8 User Confidence condition

So $\{q_4\}\rightarrow\{q_2\}$ is a confident rule and therefore, likely, since q_4 rates u_4, q_2 likely will too. As we know now that the student user U4 will visit and complete q2, then this user need not need any help. On the other hand, if the value turn to be 0 i.e., had the confidence rule failed or student not visited q2, then the recommendation agent will be automatically called up into the GUI interface to assist the student in wondering about query-2 and provide personalized assistance in completing the problem, thereby creating an adaptive learning environment using CF approach.

V. V. RESULTS

We have tested the query rules with smaller data set of 10 students and 5 queries (multiple runs), which is pretty small on MATLAB GUI environment using Associative Rule Mining (ARM), but our result show a confidence and support metric greater than 60% for each user participation reflecting the potential of using ARM based environment in future e-learning system.

VI. VI. CONCLUSION

It is evident that agent based learning object repositories are part of the future of e-Learning system, because of the great level of potential for cost reduction and personalized instruction[8]. The approach presented, could be expanded easily to each course being considered as a learning object, with a objective description of the course given by its title, number, list of pre-requisites and so on. We do see several challenges before learning objects can fulfill their potential, especially large data set for automated deployment of e-assist tools. We believe that collaborative filtering and rules has good potential in this kind of learning environment. We have demonstrated and discussed an implementation of ARM based Collaborative filtering in e-learning environment using MATLAB interface. A work in progress on similar implementation to test them in a JADE interface for verification with our preliminary confidence metric results large data set with detailed semantic rules.

REFERENCES

[1] FIPA, "FIPA Agent Message Transport Service Specification," 2002.

[2] Charu C. Aggarwal, Zheng Sun, and Philip S. Yu, Online Algorithms forFinding Profile Association Rules, *Proceedings of the ACM CIKM Conference,* 1998, pp 86-95

[3] FIPA, "FIPA Agent Message Transport Protocol for WAP Specification," 2002.

[4] FIPA, "FIPA Agent Message Transport Protocol for IIOP Specification," 2002.

[5] Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents, Manos Papagelisa, Dimitris Plexousakisa, ELSEVIER, Engineering Applications of Artificial Intelligence 18 (2005) 781–789.

[6] CSCI783/785, Dr. William Perizzo Notes in High Performance Computing, Computer Science, North Dakota State University

[7]. B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce, in: Proceedings of ACM ECommerce,2000

[8]. H. Nwana, D. Ndumu, A perspective on software agents research, Knowledge Engineering Review 14 (2) (1999) 1–18.

[9] Herlocker, J., Konstan, J.A., Terveen, L., Riedl, J., 2004. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS).