In Open Discussion Track Proceedings – International Joint Conference
IBERAMIA-SBIA 2000 - Atibaia, Brazil - 2000 - pp. 12 - 21

User profiling with Case-Based Reasoning and Bayesian Networks

Silvia N. Schiaffino[1] and Analía Amandi

ISISTAN Research Institute, Facultad de Ciencias Exactas
Universidad Nacional del Centro de la Pcia. de Buenos Aires
Campus Universitario - Paraje Arroyo Seco - B7001BBO - Tandil, Bs. As., Argentina
[1] also CONICET
{sschia, amandi}@exa.unicen.edu.ar

Abstract. Agent technology provides many services to users. The tasks in which agents are involved include information filtering, information retrieval, user's tasks automation, browsing assistance and so on. In order to assist users, agents have to learn their preferences. These preferences are represented by user profiles. Many techniques have been developed for user profiling, which vary from statistical keyword analysis to social filtering algorithms and different machine learning techniques. This paper presents a technique that integrates Case-Based Reasoning and Bayesian Networks to build user profiles incrementally. Case-Based Reasoning provides a mechanism to acquire knowledge about user actions that are worth recording to determine his habits and preferences. Bayesian Networks provide a tool to model quantitative and qualitative relationships between items of interest. Information needed to build the BN is taken from cases stored in the case base. This technique supports particularly users' routines and changes of interests over time.

1 Introduction

The idea of personal assistants supporting people to do their work has emerged in recent years. Interface agents are computer programs that employ Artificial Intelligence techniques to provide active assistance to a user with computer-based tasks [14].

In order to help users, agents need some knowledge of the tasks they have to perform, and they have to be aware of the interests, habits and preferences of the user. A prerequisite for developing systems providing personalized services is to rely on user profiles, i.e. a representation of the preferences of an individual user [4].

User profiles vary in content, in acquisition mechanism and in its usage from one agent to another. The type of information that constitutes a profile is commonly application dependent and most of this information is simply a user's set of interests. Some agents also consider the dislikes of a user, personal data about the user and some other kind of information explicitly required by the task the agent is engaged in.

Many techniques have been developed to build profiles. Most profiles are constructed either directly by users supplying items of interest or by automatic methods in which an agent is able to learn user preferences. As regards automatic

profiling mechanisms, they can be classified in three main paradigms: statistical keyword analysis, social filtering algorithms and machine learning techniques [18]. The first method is very common and relies on standard information retrieval techniques. In this method, as keywords are analyzed in isolation, there are some losses of contextual information that affects the accuracy of profiles. Social filtering algorithms generally need a large community of users to operate effectively. The third method employs machine learning algorithms to derive user profiles. The most common approach within this method consists of agents that learn from users' behavior what items of interest are relevant for them. Users provide feedback about the accuracy of the derived profile. Relevance feedback is used to guide further learning. Techniques such as memory-based reasoning, Bayesian classifiers, neural networks and genetic evolution have been used within this approach [19].

This paper presents an alternative technique to build user profiles that integrates two well-known techniques that belong to the machine learning paradigm: Bayesian Networks (BN) and Case-Based Reasoning (CBR).

A BN is a graphical model for probabilistic relationships among a set of variables [10]. In the proposed technique, BN are used to model qualitative and quantitative relationships among the different elements the user is interested in. The network structure and probabilistic values associated to variables are modified with information obtained via CBR.

CBR is a problem-solving paradigm that is able to utilize the specific knowledge of previously experienced, concrete problem situations: cases. Basically it solves a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation [2]. Each computer-based task performed by a given user represents an experience that provides an agent information about the user's habits and preferences. Tasks performed in previous situations can offer some indications about the behavior that a user would have in a similar new situation. Information stored in the form of cases is used to gradually update the BN, which models a user's interests. Cases also provide information used to detect patterns in a user's behavior and determine his routine.

This work is organized in five sections. Section 2 describes the construction of user profiles using the integrated technique. Section 3 shows an experience with the proposed technique. Section 4 describes some related work. Finally, section 5 presents the conclusions of the paper and some future work.

2 User profile construction

2.1 Technique overview

This work presents a technique that integrates CBR and BN to gradually build a user profile. In order to exemplify the use of this technique, we will consider a university that has a database among its information resources. A user, for instance a student or a professor, queries the database and performs different tasks over the database system to get the information he needs, namely information about departments, careers,

students, courses, students' marks and so on. We will focus our attention on users who need data stored in the database to fulfil their everyday tasks, or at least that often use the database. Querying the database may become a repetitive and time-consuming task for such users. The goal of an agent assisting these users is to determine their information needs and help them by providing them their items of interests.

According to the selected approach, an agent learns a user's preferences by observing his behavior while he is querying the database and recording data obtained from this observation. Information is stored in the form of cases, from a CBR point of view. Each case records the attributes or keywords used by a given user to perform queries and data related to the moment in which the query was executed (e.g. day, month, time). Queries are classified according to its similarity with previous recorded queries. Classification is done by means of similarity metrics that consider types of attributes, attribute values and temporal information similarity. Relationships between attributes are modeled using BN, and the strength of each relationship is determined by the associated probability values.

Information stored in the case base and in the BN of a given user, is used to build the user's interest profile. A profile contains information about the types of queries frequently made by a given user and the situations in which these queries are performed. It consists of both statistic and inferred information. The statistic part of the profile contains, for example, the occurrence frequency of each attribute in queries. Inferred information is basically obtained via Bayesian inference mechanisms. The user's routine belongs to the inferred profile and comprises a series of situations in which a user makes queries. Each situation, in turn, has a set of queries associated to it. However, there can be some queries that are not made in a particular situation, but at any moment. Suggested queries are obtained combining attributes and attribute values inferred relevant while building the user profile. A profile is used to suggest the execution of relevant queries to a user at an appropriate moment.

Figure 1 shows the general process of building a user's profile using our technique.

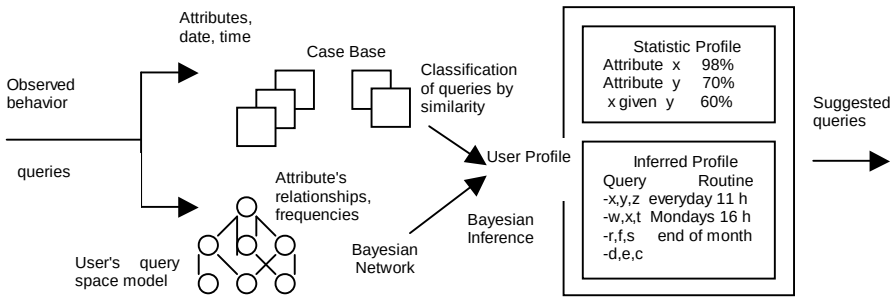

Fig. 1. User profiling integrating CBR and BN

The following subsections describe in detail the process of building user profiles using our technique.

2.2 Using Case-Based Reasoning

In CBR, a reasoner remembers previous situations similar to the current one and uses them to help solve the new problem. CBR considers reasoning as a process of remembering one or a small set of concrete instances or cases and basing decisions on comparisons between the new situation and old ones [12].

In the context of a user who often makes queries to a database system, and since CBR is based on previous experiences, queries made by a given user in past situations provide us some information about the queries that the user would possibly submit to the database in a future time.

In order to make the process description clear, we will consider an example in which a user, John Smith, frequently queries a university database looking for information. This university database stores information about departments, careers, professors, students, courses, attendance to courses, students' marks, rooms, and so on. John Smith is a professor at the Computer Science Department and teaches Logic Programming to Computer Science Engineering students. After every class, he always sends some reading material to the ones who attended his class. He queries the database to get the students' information and then he sends them some homework via email.

In this example, information about previous queries made by John Smith after a Logic Programming course gives some clues about queries he would make in similar situations. If user actions are consistent and he usually queries the database in a similar manner, his information needs can be easily inferred using our technique.

An agent whose goal is assisting users who make queries to this university database system should detect the information needs of each user. Using the proposed technique, an agent can build a user's interest profile and then perform the queries inferred relevant for him in advance. In this way, when John Smith enters into the system after a Logic Programming class, he will have the information of his students available.

Each query made by a user is represented in the form of a case. A case is an in-context piece of knowledge representing an experience, and any case worth recording in a case library teaches a lesson which is fundamental to achieve the goals of the reasoner who will use it. Each user query becomes a case that will help the reasoner to acquire information about his information needs. The attributes or features used to perform a query are very helpful for determining the topics a user is interested in.

A case has three main parts: the description of the situation or problem, the solution, and the outcome or results of applying the solution to the problem. In the chosen application domain, the description of the situation includes the attributes used to make a query (commonly named filters), the user goals, information about the user and data related to temporal aspects. This latter item is useful to determine the user's routine and includes information such as date and time when a query has been performed. The solution is, in this example, a code that identifies a certain topic of interest. This code is determined by comparing the new query with previous recorded ones and is used to group together similar queries. The outcome describes in some way the query results. Figure 2 shows a case representing a query made by John Smith.

In order to detect different preferences of a user, cases are classified according to the similarity of the queries they represent. Similar queries are assigned a code that identifies them as part of the same topic of interest. Similarity metrics compare the type of attributes involved in the queries and their values, to verify their correspondence.

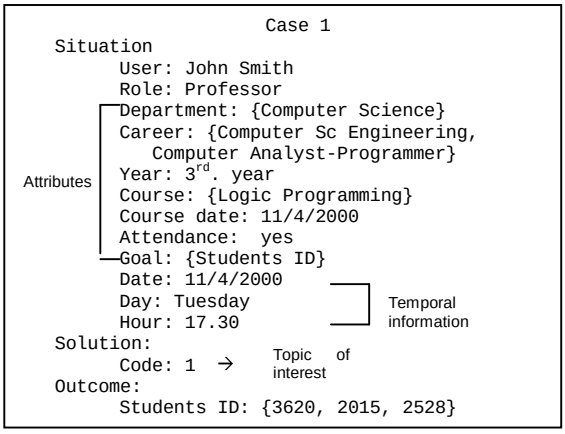

Fig. 2. A case representing a query

For example, if John Smith also taught Logic at the department of Mathematics, and he used the same methodology to send material to his students, he would make similar queries with different attribute values. Similarity metrics would associate a code to Logic Programming queries and a different one to Logic ones, because of the difference in the values of attributes such as department, career, course, course date. Each metric computes a similarity function that gives as result a score representing how similar the queries are. If this score is higher than a threshold value specified by the metric, then the queries are similar and they are assigned the same code. If not, a new code is assigned to the new query.

There are also other metrics that compare temporal aspects of queries to determine a user's routine. For example, queries can be classified according to the similarity between the day and hour of execution (e.g. Tuesdays, 6:30), or according to the time of the month (e.g. beginning or end of the month).

2.3 Using Bayesian Networks

A BN is a compact, expressive representation of uncertain relationships among parameters in a domain [7]. In particular, a BN can model the relationships between attributes involved in user's queries and also between subsequent queries.

A BN is a directed acyclic graph that represents a probability distribution. Nodes represent random variables and arcs represent probabilistic correlation between variables. Conditional probability tables specify quantitative probability information. For each node, a table specifies the probability of each possible state of the node given each possible combination of states of its parents. Tables for root nodes just contain unconditional probabilities [9].

In our technique, BN are used to model information needs of a given user. Each node represents an attribute or feature used by the user to query a database. Arcs correspond to existing relationships between the attributes used as filters. These relationships are domain dependent. Probability values are obtained by determining how many times a certain attribute is involved in user's queries. The frequency in which each attribute or feature appears in queries submitted by a particular user represents the importance of that feature for the user. This frequency is obtained analyzing the cases stored in the case base.

In the example application domain, we can point out some existing relationships between features that reflect dependencies between them. For example, for each department a user can ask about careers belonging to it. Each career has certain courses, students and professors associated to it. These restrictions can be modeled using BN by establishing relationships between the correspondent nodes. These relationships model the probability that a child node (e.g. a career) is used as filter given that a parent node is used as a filter too (e.g. a department).

Generally, a domain expert determines relationships existing between attributes, but they can also be learned as a user makes queries applying learning algorithms [10]. Relationships between attributes and the associated conditional probabilities can give information about, for example, the amount of times an attribute corresponding to a course is involved in a query when an attribute representing a particular career is also involved.

A BN is built gradually as a given user queries the database. When a user submits a query, the query is stored in the form of a case and a node is added to the network for each attribute involved in the query. Arcs are drawn between the correspondent nodes, considering the relationships established for the particular domain. Probability values are updated as attributes frequencies in queries are modified with each new query. Each variable can have only two values: true, representing that the attribute is present in the query, and false, indicating that the attribute is absent.

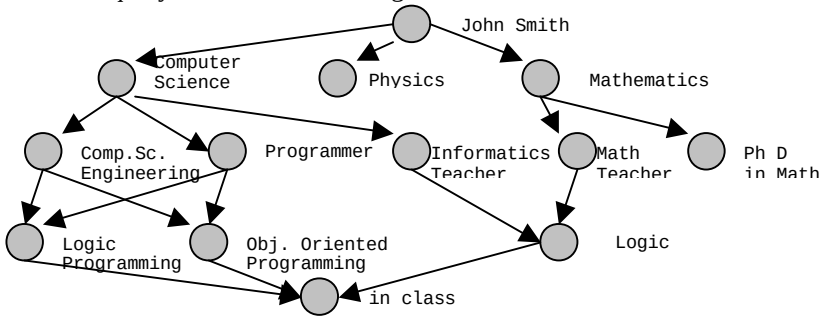

Fig. 3. BN representing a user's query space model

Figure 3 shows an example of a BN that models John Smith's query space in the example application domain. This network is the result of a sequence of queries made by the user after his teaching classes.

The model is completed by establishing the probability values associated to each node of the graph. Table 1 shows an example of simple probabilities associated to

departments. Simple probabilities are those associated to variables that do not depend on other variables (except on the user); departments in our example.

Table 1. Simple probability values

Computer Science	Mathematics	Physics
0.6	0.2	0.2

The previous values indicate that 60% of queries made by John Smith are related to Computer Science Department, 20% to Mathematics Department and 20% to Physics Department. Table 2 shows an example of conditional probability values.

Table 2. Conditional probability values

	Logic Prog OO Prog	¬Logic Prog OO Prog	Logic Prog ¬OO Prog	¬Logic Prog ¬OO Prog
Comp Sc Eng.	0.375	0.5	0.125	0
¬Comp Sc Eng	0	0	0	0

The cell whose value is 0.375 means that in queries made by John Smith where the Computer Science Engineering appears, the Logic Programming and Object Oriented Programming courses will also appear with a probability of 37,5%.

2.4 Using both techniques together

Integration of BN and CBR can be achieved with either of the methods as the master and the other as the slave, depending on which method uses information provided by the other [8]. Our technique uses CBR as the slave. Cases recorded in the case base are used to calculate the probability values associated to each node of the BN.

Efficient inference algorithms exist for deriving answers to queries given a probability model expressed as a BN. Considering the relationships between features modeled by the network, these inference mechanisms are used to derive which the most probable queries are. Cases are used to filter queries that were suggested using Bayesian inference mechanisms.

A user profile is built with information stored in the case base and information stored in the BN. A profile contains statistic data about queries made by the user and inferred information about his topics of interest. Statistic information includes items such as: most frequent queried attributes; for each attribute it contains the most frequent queried values; most frequent goals; most frequent requested query codes; most frequent values of an attribute given values of another attribute. Inferred information contains items such as attributes inferred relevant for the user, values inferred relevant for each relevant attribute; values inferred relevant given a value of another attribute; combinations of relevant inferred attributes. The user's routine is also part of the profile. This routine consists of a set of situations in which a user makes queries to a given database. Some queries are general, which means that they are not made at a particular moment and other queries are performed in certain situations specified in the routine. Figure 4 shows an example of John Smith’s profile.

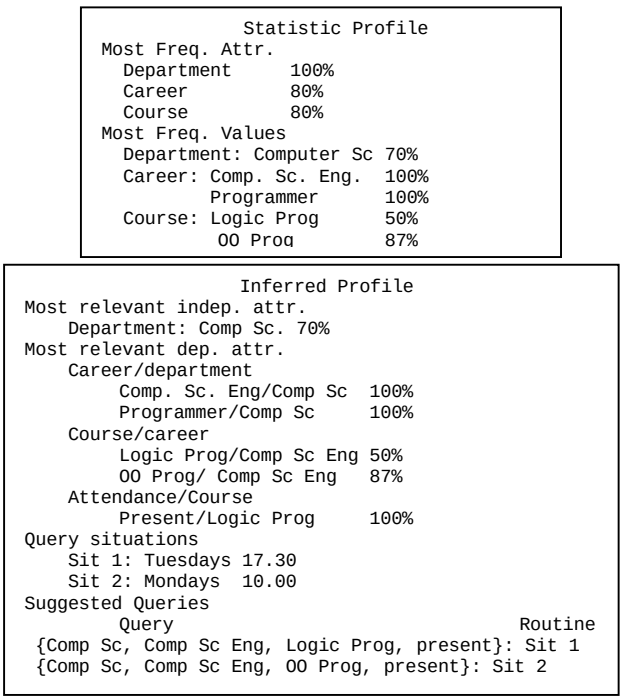

Fig. 4. User profile

The most important item of the inferred profile is ‘suggested queries’. Suggested queries are formulated combining attribute values inferred relevant for the user. An independent attribute (one whose node does not have parents, but the user name) is inferred relevant if its simple probability value is higher than a specified threshold. To determine the importance of a dependent value, the technique first sets as evidence relevant values of its parents. Then, via inference Bayesian mechanisms, it determines which of the values of the child attribute have higher probability values. Combining the obtained values for each attribute possible relevant queries are built. Possible queries are filtered using cases in order to suggest only queries that make sense for the user.

3 Experiences

The proposed technique has been incorporated in the development of an intelligent agent that assists a user who operates the subsystem in charge of sample tracking within a LIMS (Laboratory Information Management System). A user of such system makes queries to the LIMS's distributed database to obtain information about samples according to his work needs. The goal of the agent is to detect users' information needs and offer them relevant data at the right time.

In this domain, users make queries considering attributes such as sample types, product or material names, sample states within the system, department in which samples were requested, sample points and so on.

Experiments made so far have demonstrated the usefulness of the proposed technique to model user preferences and interests. According to profiles built integrating CBR and BN, the agent performs relevant queries in advance. Then, users have the information they need ready for them to use it without having to request it.

4 Related work

There are several machine learning approaches that can be used to learn a user profile, such as Bayesian classifier, nearest neighbor, PEBLS, decision trees, TF-IDF, neural networks and genetic algorithms [11,13,15,16,17]. Most learning approaches also include relevance feedback analysis. Social filtering algorithms instead of learning profiles, they compare different users' profiles. Other existing mechanisms, require users to supply items of interest.

Our approach belongs to the machine learning paradigm and combines the characteristics of CBR and BN to gradually build user profiles. The chosen application domain is an example of a domain in which existing techniques, which are most task and profile content dependent, are not viable. Most of the existing user profiling techniques were developed in the area of browsing assistants, filtering and searching agents, and rely on some kind keyword representation of documents and topics of interest.

Integrations of CBR and BN have been used in the development of different systems, but the authors have not found applications in user profiling. In Microsoft Research it was developed a system for fault diagnosis tasks in MS Word and MS NT [5]. Croft and Turtle address document retrieval using CBR and BN [6]. In [3] the domain model built combining semantic and Bayesian nets is used to support CBR processes. Other integrated systems are described in [1,3,8].

5 Conclusions and future work

This paper presents a technique to build user profiles that integrates CBR and BN. Our technique enables an agent to learn a user profile incrementally and continuously. The proposed approach combines the capability of BN to model the relationships between items of interests to a user, both in a quantitative and a qualitative way, and the utilization of knowledge stored in the form of cases to modify the structure and strength of these dependencies, according to user actions. Bayesian inference mechanisms are also a key concept in this profile building approach.

Experiments made so far have showed that the integrated technique is a viable alternative to build user profiles. Further experiments have to be made in order to demonstrate the usefulness of the technique over time and its capability to model changing users' interests.

Future work includes considering user feedback in profile construction. User feedback will be used to modify information contained in the BN.

Acknowledgements: We would like to thank Analyte - Lab Information Technologies - for providing us information to test our technique in a LIMS domain.

References

1. Aha, D., Chang L. W.: Cooperative Bayesian And Case-Based Reasoning for Solving Multiagent Planning Tasks - Technical Report - NCARAI, Washington DC, USA - Number AIC-96-005 - (1996)
2. Aamodt, A., Plaza E.: Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. AI Communications, v. 7, n.1, (1994) 39-59
3. Aamodt, A., Langseth, H.: Integrating Bayesian networks into knowledge-intensive CBR. In American Association for Artificial Intelligence, Case-based reasoning integrations; Papers from the AAAI workshop. Technical Report WS-98-15. AAAI Press - (1998) 1-6
4. Amato, G., Straccia, U.: User Profile Modeling and Applications to Digital Libraries - In Proceedings of ECDL 99 - LNCS 1696 - (1999) 184 - 197
5. Breese, J., Heckerman D.: Decision-Theoretic Case-Based Reasoning - In Proceedings of Fifth International Workshop on Artificial Intelligence and Statistics - (1995) 56 - 63
6. Croft W.B., Turtle H.R.: Retrieval Strategies for hypertext - Information Processing and Management, (1993) 29, 313-324
7. D'Ambrosio, B: Inference in Bayesian Networks - AI Magazine, Summer - (1999) 21 - 35
8. Dingsoyr, T.: Integration of Data Mining and Case-Based Reasoning - Master Thesis - Department of Computer and Information Science - Norwegian University of Science and Technology (1998)
9. Haddawy, P: An overview of Some Recent Developments in Bayesian Problem-Solving Techniques - Introduction to this Special Issue - AI Magazine, SUMMER - (1999) 11-19
10. Heckerman, D.: A tutorial on Learning with Bayesian Networks - Technical Report MSR-TR-9506 - Advanced Technology Division, Microsoft Research - (1995)
11. Joachims, T., Freitag, D., Mithcell, T.: WebWatcher: A tour Guide for the World Wide Web - In Proceedings, IJCAI 97 (1997)
12. Kolodner, J.: Case-Based Reasoning - Morgan Kaufmann Publishers (1993)
13. Lieberman, H.: Letizia: An Agent That Assists Web Browsing - In Proceedings of International Joint Conference on Artificial Intelligence - Montreal (1995)
14. Maes , P.: Agents that Reduce Work and Information Overload - Communications of the ACM (1994) -
15. Pazzani, M.,Muramatsu J., Billsus, D.: Syskill&Webert: Identifying interesting Web sites - Proceedings of the National Conference on Artificial Intelligence, Portland - (1996) 54-61
16. Pazzani, M., Billsus, D.: Learning and Revising User Profiles: The Identification of Interesting Web Sites - In Machine Learning 27, (1997) 313-331
17. Sheth, B.: A Learning Approach to Personalized Information Filtering - Master Thesis of Science in Computer Science and Engineering - MIT (1994)
18. Soltysiak, S., Crabtree, B: Knowing me, Knowing you: Practical Issues in the Personalization of Agent Technology - In Proceedings, PAAM98 - London (1998)
19. Soltysiak, S., Crabtree, B: Automatic learning of user profiles - towards the personalisation of agent services - BT Technol J, Vol 16 No 3 July 1998