Utilizing Goal-Directed Data Mining For Incompleteness Repair In Knowledge Bases

Daniel J. Stein III, Sheila B. Banks, Eugene Santos, Jr., and Michael L. Talbert
Artificial Intelligence Laboratory
Air Force Institute of Technology
Wright-Patterson Air Force Base, OH 45433
sbanks, esantos@afit.af.mil

Abstract

In this paper we present a methodology for goal-directed data mining of association rules and incorporation of these rules into a probabilistic knowledge base. The purpose of the data mining and rule extraction process is to repair knowledge base incompleteness uncovered during validation. We discuss how this incompleteness is uncovered and show the fundamental forms this incompleteness can take. We describe how association rules can be extracted from databases in order to address excluded information and to express missing relationships in a probabilistic knowledge base. The current implementation of this goal-directed data mining within an integrated generic expert system tool is also described. Our methodology can benefit many data intensive and imprecise domains such as stock market analysis, intelligence analysis, and operational management.

Introduction

The relationship between the efficiency of a reasoning algorithm and the flexibility of its knowledge representation scheme is an inverse one. In order to implement a realistic, real-world application, both of these properties must not only be balanced, but maximized as well. Bayesian Knowledge Bases (BKBs) provide the needed blend of efficiency and flexibility, while also providing an ease of understanding lacking in many representation schemes (Santos, Jr., & Santos 1996). For these reasons, the BKB is the representation for the Probabilities, Expert System, Knowledge, and Inference (PESKI) environment, an integrated framework for expert system development (Santos, Jr., & Santos 1996).

Validation of a BKB is performed by submitting test cases and comparing the expected solutions with the actual ones. Incompleteness in a BKB is encountered during testing whenever the inference engine cannot reach one or more elements in the expected solution. This normally happens because one or more relationships are missing from the BKB. This paper addresses a method for automatically extracting the necessary relationships uncovered as lacking during testing and incorporating those relationships back into the BKB.

Presently, incompleteness in BKBs is repaired by applying the same knowledge acquisition techniques that created the BKB. Whenever incompleteness is encountered, the PESKI user must manually augment the BKB to fill in the missing areas. In this work, we show that it is possible to repair each of the primary forms of incompleteness using data mining techniques. In addition, typical data mining approaches can become bogged down by an overabundance of patterns. We show how our approach, *goal-directed* data mining, can help bound the scope of data mining operations and make those operations more feasible. In general, our methodology can benefit many data intensive and imprecise domains such as stock market analysis, intelligence analysis, and operational management.

Background

PESKI is the physical realization of an integrated knowledge-based system framework that combines the functions of natural language interface, inferencing, explanation and interpretation, and knowledge acquisition and maintenance into a single, consolidated application (Santos, Jr., & Santos 1996). PESKI is the combination of the following closely interrelated, yet specialized tools: (1) Intelligent Graphical User Interface, (2) Inference Engine, (3) Knowledge Acquisition, (4) Verification and Validation, and (5) Data Mining. PESKI is currently in the prototype stage, where each of the fully functioning components are operating within a single cohesive whole.

PESKI uses the Bayesian Knowledge Base (BKB) knowledge representation scheme. BKBs depend on Bayesian probabilities to represent uncertain information in a knowledge base. This probabilistic aspect of BKBs makes them almost ideal for operating in an uncertain environment and enables BKB systems to make inferences using incomplete knowledge. Incompleteness is allowed in a BKB, but only as long as the requirements (i.e., the conclusions drawn based on given evidence) of the BKB are kept consistent. Incompleteness in BKBs occurs whenever essential connections are missing between nodes or when nodes lack necessary states. Problems can arise whenever inferencing is attempted on a knowledge base that is incomplete.

Incompleteness in a BKB cannot be determined simply by inspection. It must be determined based upon the knowledge base validation process. Validation guarantees the system produces the correct output and that it does what the users actually want it to do (Gonzalez & Dankel 1993), (O'Keefe, et. al. 1987). Validation is performed in PESKI by submitting a series of test cases and comparing the resulting solution to the one that was expected. Each

test case consists of a number of pieces of evidence (known events) and a set of expected answers (anticipated events). A test case is said to be valid if the expected answers are part of the overall solution set obtained after inferencing over the BKB based on the evidence. In PESKI, incompleteness is evident whenever it is impossible to conclude one or more elements of the answer set given the evidence. Incompleteness uncovered during validation can occur in one of three fundamental ways: (1) a relationship between two different states in the BKB is missing (i.e., missing probabilistic link, either direct or indirect, between some state in the evidence set and some other state in the answer set); (2) a given state may have insufficient support conditions (i.e. more evidence is needed to indicate the instantiation of a given state); (3) a component has one or more missing or unspecified states.

Data mining techniques offer a number of possible solutions to repair incompleteness in knowledge bases. Only recently has data mining been distinguishable from other knowledge gathering activities (Frawley, et. al. 1992), (Moulet & Kodtratoff 1995). A good working definition for data mining is *the automatic extraction of useful information from raw data* (Fayyad, Piatetsky-Shapiro, & Smyth 1996). Data mining usually refers to tools and methods used to extract meaningful information from data that is unformatted and either unstructured or partially structured (Kloesgen & Zytkow 1996). There are a number of different synonyms for data mining, including knowledge extraction, database exploration, information harvesting, and knowledge discovery in databases (KDD). In each case, the purpose of the activity is the nontrivial extraction of implicit, previously unknown, and potentially useful information. Extraction of association rules is just one of the many data mining techniques (Piatetsky-Shapiro, et. al. 1994). Association rules are those in which one or more items in the antecedent of an implication are correlated with one or more items in the consequent with some level of confidence and support. An example of an association rule would be, "If a supermarket customer buys bread and eggs, he will also buy milk with a 90% probability." In this rule, the purchase of bread and eggs comprise the antecedent, the purchase of milk the consequent, and the value 90% is the confidence (Agrawal, et. al. 1993). When searching for a rule of the form X $\Rightarrow$ Y (read "X implies Y"), that rule has a confidence value of C if C% of the database records containing X also contain Y. If S% of the records in the database contain both X and Y, then the rule has a support value of S (Agrawal and Ramakrishnan 1994).

Goal-directed Data Mining in PESKI

Of all the methods considered, association rules provided the best fit to our current BKB incompleteness problem. The basic algorithm for determining an association rule is relatively straightforward, and the resulting rule is already in the correct form for incorporation into a BKB (or, for that matter, in any other probabilistic knowledge base). When searching for a rule of the form X $\Rightarrow$ Y, we search the entire database and compute the percentage of records containing X that also contain Y. This value, called the confidence, is treated in our work, as in other similar research (Agrawal & Ramakrishnan 1994), (Srikant & Agrawal 1996), as the probabilistic strength of the association rule.

The support for a rule is the value that represents the frequency of co-occurrence of all the variables in that rule within the database. The support for the rule would be the percentage of records in the database that included both X and Y. It is important that the minimal support value threshold for a rule is well chosen: if the support is too low, unfounded rules with sufficient confidence values could be captured as apparently valid associations. If the only occurrences of X and Y were together in the same record out of a total 100,000 records, the confidence for the derived rule would be 1.0 (or 100%), since *every* instance of X would be correlated with an instance of Y. However, the support, only 0.00001, would be insufficient in almost any practical circumstance for the formation of a rule. A subtle but important point regarding confidence is that if A $\Rightarrow$ B with confidence C, we cannot automatically draw the conclusion that B $\Rightarrow$ A with the same confidence. Association rules discovered in this way are not necessarily invertible.

Our data mining approach is slightly different in that each mining operation is aimed at finding *specific* rules relating two or more database attributes instead of the traditional approach that attempts to derive *all possible* rules meeting minimum support and confidence criteria for all possible combinations of items in each itemset (Agrawal & Ramakrishnan 1994), (Houtsma & Swami 1995). We call these focused operations *goal-directed data mining*, the goal being the association of two specific states to each other, either directly or indirectly, and the search for specific rules (i.e., a search for associations between specific states), rather than the search for all possible rules. We enhance the effectiveness of data mining operations by performing the following: (1) always attempting to find an association rule involving a particular state; (2) eliminating attempts at rule formation whenever possible by considering the support value of each state involved; (3) preventing the same states from being compared more than once, thus avoiding circular or repetitive rules.

The data mining operation extracts information in one of three forms depending on the type of incompleteness encountered during knowledge base validation. These three forms correspond to the respective category of incompleteness (see Section 2) and are as follows: (1) a series of association rules relating an antecedent state to a consequent state; (2) a set of associations related to a single consequent; (3) a set of states for a particular component in the database. Incompleteness categories 1 and 2 can be directly solved by a goal-directed search for association rules. In incompleteness category 1, we try to find an association rule of the form X $\Rightarrow$ Y, where X and

Y are component states in need of an unspecified relationship. If both the confidence and support values of the rule $X \Rightarrow Y$ meet the minimum values specified by the user, then the relationship is considered to be a direct one. On the other hand, if the minimum values are not met, the data mining tool searches for all possible relationships of the form $X \Rightarrow Z_i$, i = 1, 2, .. *n* where *n* is the user-specified branching factor and the confidence of the rule $X \Rightarrow Z_j$ is greater than or equal to the confidence of $X \Rightarrow Z_{j+1}$. The tool then tries to associate each Z_i with Y by finding rules of the form $Z_i \Rightarrow Y$. If such associations are not possible, the process continues until either some associative relationship is found to Y, or until a user-specified *lookahead value* is exceeded. This lookahead value is used to specify the maximum number of intermediate relationships allowed between the original antecedent (X) and consequent (Y). For instance, a lookahead value of 1 would allow only a single intermediate state between X and Y (e.g. $X \Rightarrow Z \Rightarrow Y$). Note that it is feasible that each individual Z_i branch could lead to the consequent Y.

In category 2, we look for support conditions that are immediately related to a given node. In terms of data mining, this reduces to searching for all association rules of the form $X_i \Rightarrow Y$, i = 1, 2, .. *n* where Y is the state for which support is needed, *n* is the user-specified branching factor, and where the confidence of the rule $X_j \Rightarrow Y$ is greater than or equal to the confidence of $X_{j+1} \Rightarrow Y$.

Finally, in category 3, we are given a component (an attribute in database terms) for which new states are needed. The data mining tool searches the database and extracts all possible states for that component. This is a reasonably straightforward process for categorical (non-numerical) components. We simply examine each record and if we discover a previously unencountered state, we add the new state to a list. However, much more work is involved for numerical components. There are several problems concerned with choosing states, or intervals, for numeric components. If the number of states for the component is large (i.e. if the size of each interval is small), then the support for any state will probably be low. As a result, any potential rules involving the numeric component may never be discovered. Conversely, there is always some information lost whenever we partition numeric values into intervals. This loss increases as the size of the interval increases. Some rules may only have sufficient confidence when the numeric interval consists of a single value. When dealing with numeric database attributes, support and confidence values are inversely related (Srikant & Agrawal 1996). To balance these factors, we set the number of intervals, or states, for each numeric component to be $1/S$, where S is the user-specified support value (see Srikant and Agrawal (Srikant & Agrawal 1996) for a detailed proof).

The PESKI Data Mining Tool

The current implementation of the PESKI Data Mining tool is illustrated by control screen shown in Figure 1. In this example, a BKB (a goldfish diagnosis BKB *GF2*) has already been loaded into PESKI. One of the existing components, *Chlorine Level*, has been selected and the *Find States* mining operation is about to take place. None of the required numeric parameters (*Minimum Support*, etc.) have been set. The *Status* window, in the lower right-hand corner, is used to communicate interim messages to the user of the data mining tool. In this case, the user has been notified that *Chlorine Level* has been selected for this mining operation. In the upper right-hand corner of the screen, the user indicates the mining source in the *Search Location* window. This can be done either manually or by browsing the contents of any available disk drive. The *Results* window displays the outcome of the data mining operation in a textual format. The user is able to select these results individually for incorporation into the BKB. Note that the user has the ability to terminate any data mining operation by pressing the *Stop Mining* button in the upper right-hand corner.

Future Work and Conclusion

The immediate future of the PESKI data mining tool is to incorporate the ability to maneuver through various databases, each potentially having a different format. The ability to maneuver through various databases is utilized whenever a mining operation cannot be successfully completed and the model tool attempts to further the mining effort. The tool is then required to autonomously identify and prioritize a set of possibly heterogeneous sources and select its own data mine or mines. This is to be done using embedded heuristic rules. In the future this ability of the data mining tool will become truly intelligent and "learn," through trial-and-error experience, which sources were more applicable to the problem domain and which evaluation criteria were the better predictors of usefulness. The tool will then be able to contemplate new sources for exploitation, classify them based on their significance and relevance, and select the most promising ones to exploit. The tool should *consider* each database as being a potential source of knowledge, but it should only mine the most promising ones for the desired information.

This paper described a developed methodology and tool for mining association rules and incorporating those rules into a knowledge base. The results extracted by our tool are intended for incorporation into a BKB, but are in a form suitable for incorporation into any knowledge base with a probabilistic representation. In addition, our data mining tool is designed specifically for integration into PESKI, though the techniques we present are general enough for incorporation into other comparable systems.

References

Agrawal, Rakesh, et. al. 1993. Database Mining: A Performance Perspective. *IEEE Transactions on Knowledge and Data Engineering* 5(6): 914-925.

Agrawal, Rakesh and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules. In Proceedings of the 20th Very Large Data Bases Conference, 487-499. Santiago, Chile.

Frawley, William J., et. al. 1992. Knowledge Discovery in Databases: An Overview. *AI Magazine* Fall: 57-70.

Gonzalez, Avelino J. and Douglas D. Dankel. 1993. *The Engineering of Knowledge-Based Systems, Theory and Practice*, Englewood Cliffs, New Jersey: Prentice Hall.

Houtsma, Maurice and Arun Swami. 1995. Set-Oriented Mining for Association Rules in Relational Databases. In Proceedings of the International Conference on Data Engineering, 25-33. Taipei, Taiwan.

Kloesgen, Willi and Jan Zytkow. 1996. Machine Discovery Terminology. http://info.gte.com/~kdd/kdd-terms.html.

Moulet, M. and Y. Kodtratoff. 1995. From Machine Learning Towards Knowledge Discovery in Databases. In 1995 IEEE Colloquium on Knowledge Discovery in Databases, 5/1 - 5/3.

O'Keefe, Robert M., et. al, 1987. Validating Expert System Performance. *IEEE Expert* Winter: 81-89.

Piatetsky-Shapiro, Gregory, et. al., 1994. KDD-93: Progress and Challenges in Knowledge Discovery in Databases. *AI Magazine* 15(Fall): 77-82.

Santos, Jr., Eugene and Eugene S. Santos. 1996. Bayesian Knowledge-Bases, Technical Report, AFIT/EN/TR96-05, Department of Electrical and Computer Engineering, Air Force Institute of Technology.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P., 1996. The KDD Process for Extracting Useful Knowledge from Volumes of Data, Communications of the ACM, 39(11): 27-34.

Srikant, Ramakrishnan and Rakesh Agrawal. 1996. Mining Quantitative Association Rules in Large Relational Databases. In Proceedings of the ACM SIGMOD Conference on Management of Data, Montreal, Canada.

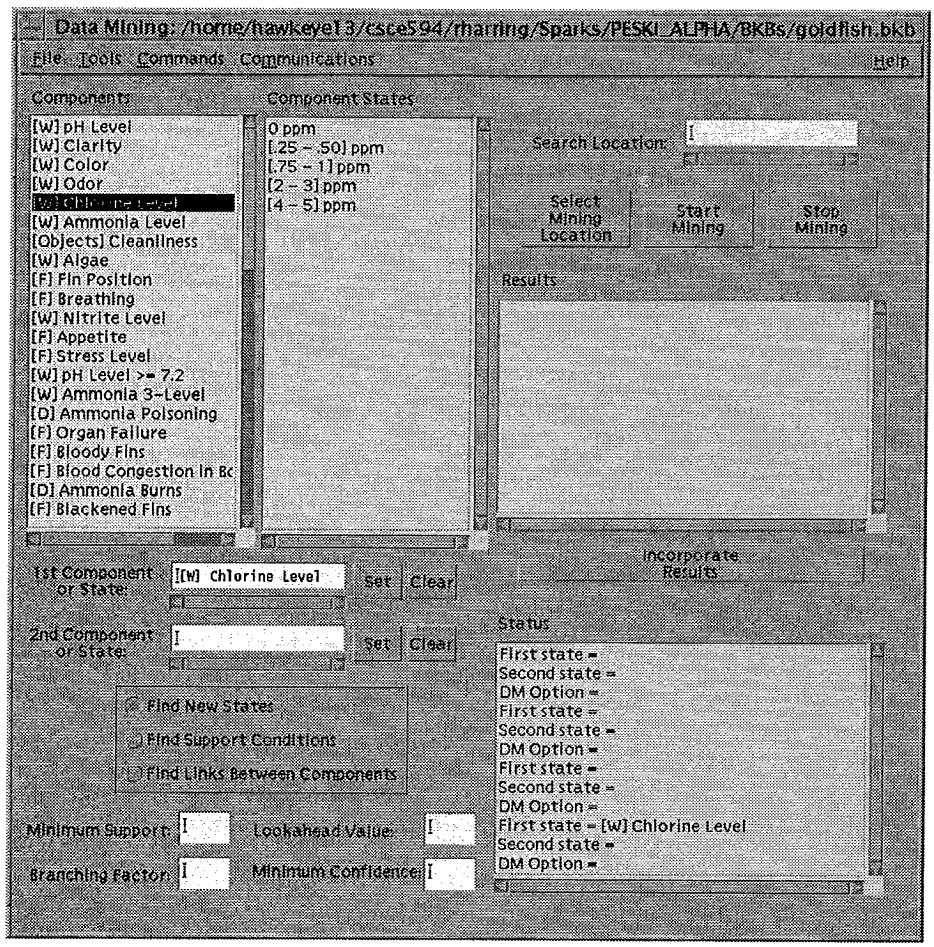

Figure 1. The PESKI Data Mining Tool