The Development of the Inductive Database System VINLEN: A Review of Current Research

Kenneth A. Kaufman[1] and Ryszard S. Michalski[1,2]

[1] Machine Learning and Inference Laboratory, George Mason University, Fairfax VA 22030, USA

[2] Institute of Computer Science, Polish Academy of Scinces, Warsaw, Poland

Abstract. Current research on the VINLEN inductive database system is briefly reviewed and illustrated by selected results. The goal of research on VINLEN is to develop a methodology for deeply integrating a wide range of *knowledge generation operators* with a relational database and a knowledge base. The current system has already integrated an AQ learning system for generating *attributional rules* in two modes: *theory formation*, in which generated rules are consistent and complete with regard to data, and *pattern discovery*, in which generated rules represent strong patterns, not necessarily consistent or complete. It also has integrated a conceptual clustering module for splitting data into conceptual classes, and providing descriptions of those classes. Preliminary data management and knowledge visualization operators, such as the *intelligent target data generator* (ITG) and *concept association graph* display, have also been integrated. To facilitate an easy interaction with the system, a user-oriented visual interface has been implemented. An example of results from applying VINLEN to a medical problem domain is presented to illustrate VINLEN knowledge discovery and representation capabilities.

1 Introduction

The field of databases is in the midst of an extraordinary growth, and databases are becoming omnipresent and globally connected. In this context, a new research direction has been recently proposed to deeply integrate database technology with modern methods for inductive knowledge generation and for storing and using the knowledge so created. Such integrated systems are called *inductive databases*. In contrast to conventional databases, inductive databases can answer not only queries for which answers are stored in the database, but also queries that require synthesizing and applying *plausible knowledge*, generated by inductive inference from the facts in the database and prior knowledge. Inductive databases can be viewed as a natural next step in the development of the database technology.

This paper presents a brief review of research on the inductive database system VINLEN, which is being developed at the Machine Learning and Inference Laboratory at George Mason University. In VINLEN, inductive inference capabilities, combined with standard relational database operators implemented through an SQL client, are implemented by developing a new type of database operators, called *knowledge generation operators (KGOs).* KGOs operate on *knowledge segments*, consisting of a combination of one or more relational tables and related knowledge in the knowledge base. A KGO takes one or more input knowledge segments and generates an output knowledge segment.

Two important constraints have been imposed on knowledge generation operators in VINLEN: (1) that their results be in a form that is easy to understand and interpret by users, and (2) that knowledge generated can be expressed compactly and efficiently. These capabilities are achieved by applying ideas and methods for *natural induction*, in which inductive methods create data descriptions in the forms that appear natural to people, by employing *attributional calculus* as a representation language [8]. Attributional calculus is a logic system that combines elements of propositional, first order predicate, and multiple-valued logics. It serves both as an inference system and as a knowledge representation language. Attributional rules, the primary form of knowledge representation in VINLEN, are more expressive than conventional decision rules that use <attribute-relation-value> conditions.

2 VINLEN System

2.1 An Overview

Research on the VINLEN system grows out of our previous efforts on the development of INLEN, an early system for integrating databases and machine learning and inference mechanisms for the purpose of multistrategy learning, data mining, and decision support [12,9]. INLEN included multiple learning and discovery operators, the high-level knowledge generation language KGL-1 [4,10], and an advisory system. It did not, however, integrate an actual database system, instead being constrained to relatively small tables located in reserved files. VINLEN is an entirely new implementation that combines and extends INLEN with a wide range of new or improved knowledge generation operators and an advanced visual interface that provides an easy access to all system operators and components.

Researchers have started to acknowledge the advantages in integrating inductive inference capabilities in a database system. Among the approaches somewhat similar in philosophy to the one presented here are ones presented by Morik and Brockhausen [14], Mannila [7], and Roddick and Rice [15]. In general, VINLEN differs from these in terms of its much wider variety and complexity of knowledge generation operators, its knowledge storage mechanism, and its visual interface.

Fig. 1 presents a general schema for VINLEN. The top part presents a database, which can be local or distributed. VINLEN's database can be one of the widely used commercial databases, for example, ORACLE or ACCESS. The *Target Knowledge Specification* is a generalization of a database query;

specifically, it is a user's request for a knowledge segment to be created by the system.

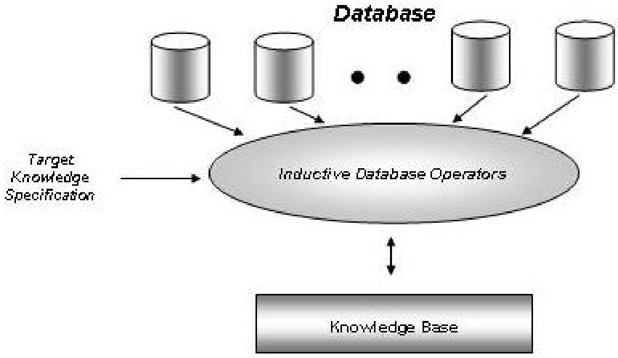

Fig. 1. A general schema of the VINLEN inductive database

The central part, *Inductive Database Operators*, contains a collection of operators that call upon various programs for knowledge generation (e.g., rule learning, conceptual clustering, target data generation, etc.) as well as SQL functions. These operators can be invoked by a user directly, or through a *knowledge query language (KQL).*

The KQL is an extension of the SQL database query language that integrates SQL with the KGL-1 capabilities. In addition to conventional data management operators, it thus includes operators for conducting inductive and deductive inference, statistical analysis, and many supportive functions. KQL is quite different from traditional high-level languages for data exploration, which have generally been Prolog-based. Among the exceptions to the Prolog-based approach, M-SQL [3] is philosophically similar to KQL in that it builds upon the SQL data query language, integrating it with one inductive operator. KQML [2] allows the querying for specific pieces of knowledge, although it does not support the abstract templates and multiple discovery operators supported by KQL.

The VINLEN knowledge base contains definitions of the domains and types of attributes in the database, data constraints, value hierarchies of structured attributes, known relationships binding attributes, and any other background knowledge that users may have represented in the system. During the operation of an inductive database, the knowledge base is populated by newly generated data descriptions, hypothetical patterns, data classifications, statistical information, results from hypothesis testing, etc.

To provide a general overview and easy access to all VINLEN operators, we have developed a visual interface that consists of VINLEN views at different abstraction levels. Fig. 2 presents the most abstract view of the main panel of VINLEN. The central part contains icons for invoking databases (DB), knowledge bases (KB), and knowledge systems (KS) currently implemented in the system. The term knowledge system stands for a system integrating a database and a relevant knowledge base to support knowledge mining and knowledge use for a specific application problem. By clicking on DB, KB, or KS, the user can select and access available to VINLEN databases, knowledge bases, and knowledge systems, respectively.

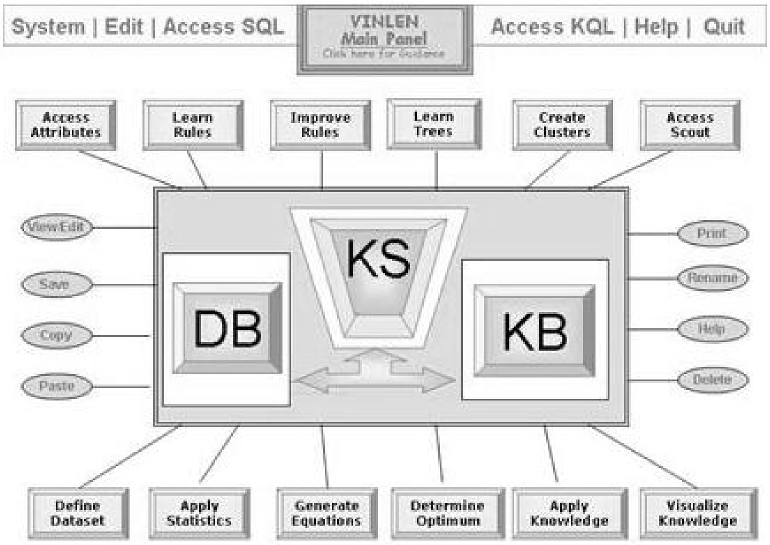

Fig. 2. VINLEN inductive database visual interface (main panel, abstract view)

The twelve rectangular buttons above and below the central region represent diverse classes of knowledge generation operators, and data and knowledge management operators. Each button invokes a pull-down menu whose options represent different programs for performing these classes of tasks, or different modes of operation of the same program. At the top of the interface is a set of pull-down menus consisting both of standard operators and mechanisms for direct access to an SQL client (for data management) and a KQL interpreter for invoking *knowledge scouts*, personal agents whose function is to synthesize and manage target knowledge [6].

The data in each VINLEN knowledge system are stored internally in relational tables, as are other system components. Prior knowledge and generated knowledge are stored in a hierarchy of relational tables. Parameter sets for individual operators are stored in system-specific tables. This storage methodology facilitates an efficient access to components by the system through a standard SQL interface.

2.2 Knowledge Representation

To illustrate the knowledge storage mechanism, consider a selection of attributional rules representing patterns discovered in experiments on a medical database containing information about men age 50-65, their diseases and their lifestyles (Fig. 3). These patterns were learned from 2063 cases describing patients with high blood pressure and 5288 patients without it (these total numbers of positive and negative examples are denoted P and N, respectively). The first pattern (A) represents a very simple and approximate regularity stating that patients with high or greater *rotundity* (a function of their weight and height) tend to have high blood pressure (the raw support of the rule, p, is 689 patients out of 2063, or 33%). The value $n = 1058$ indicates the number of patients, 20% of the 5288, that have high or higher rotundity, but do not have high blood pressure.

```
High_Blood_Pressure is present in patient-set if:
A  1  [Rotundity >= high] (p:689, n:1058)
     (p:689, n:1058, q:0.23)
B  1  [Heart_Disease = present](p:284, n:355)
   2  [Rotundity <= average] (p:1328, n:4120)
     (p:205, n:282, q:0.14)
High_Blood_Pressure is absent if:
A  1  [education >= vocational] (p:3691, n:1333)
   2 [Diabetes = absent] (p:5089, n:1884)
   3 [Rectal_Polyps = absent] (p:5072, n:1967)
     (p:3396, n:1167, q:0.26)
B  1  [Rotundity <= average] (p:4120, n:1328)
   2 [Heart_Disease = absent] (p:4933, n:1779)
   3 [Diabetes = absent] (p:5089, n:1884)
   4 [Exercise >= medium] (p:4219, n:1571)
   5 [Stroke = absent] (p:5249, n:2018)
   6 [Kidney_Disease = absent] (p:5228, n:2013)
   7 [Colon_Polyps = absent] (p:5191, n:2025)
   8 [education <= hs_grad] (p:1550, n:709)
     (p:851, n:238, q:0.18)
```

Fig. 3. Attributional rules indicating the likely presence or absence of high blood pressure

Thus, the above pattern indicates only a tendency, but not a very strong relationship. The parameter q denotes a measure of *pattern quality*, defined as the weighted product of the *completeness* (support percentage) and *consistency gain*, which is defined as the ratio:

$$q = (1 - C_p) / (1 - C_g)$$

where C_p is the consistency of the pattern (p / (p + n)), and C_g is the degree of confidence in randomly guessing a positive diagnosis (P / (P + N)). Thus, a rule with q=1 will be perfectly complete and consistent with regard to the training data, and a rule with q=0 will have a consistency equal to that of a random guess. Detailed explanations of the q measure can be found in [5,11].

The second pattern (B) found in the data indicates that presence of high blood pressure is associated with a history of heart disease and low to average rotundity. Of the cases that satisfied the heart disease condition, 284 had high blood pressure, and 355 did not. Of the cases that satisfied the rotundity condition, 1328 had high blood pressure, and 4120 did not. Of the cases that satisfied both conditions, 205 had high blood pressure (out of 2063, that is, 10%), and 282 did not (out of 5288, that is, 5%).

The above attributional rules are conjunctions of between one and eight *conditions*, which take the form `[<attribute(s)> <relation> <value(s)>]`. The rules and conditions are annotated by weights indicating their positive and negative coverage. Each diagnosis/decision is represented by a *ruleset*; in this example, the two rulesets each consist of two rules. The two rulesets comprise what we call a *ruleset family*, a collection of knowledge relevant to the task of determining the value of some attribute(s), in this case high blood pressure. Thus, while the knowledge is stored in relational tables, the attributional calculus representation allows it to be organized in such a way that components can be accessed and analyzed by the system. Specifically, VINLEN organizes the knowledge in a hierarchical set of tables: **Ruleset**, which identifies the ruleset's consequent (decision) and the numbers of positive and negative examples from which the ruleset was generated; **Complex**, which includes information on specific rules, including their rank in their rulesets (ordered by support), and their positive and negative training example coverage; **Selector**, which contains the details of the individual conditions that comprise rules' antecedents, including the attribute, relation and value portions of the conditions, and their positive and negative training example coverage levels; and **Learn**, which archives the parameters under which the rulesets were learned.

Fig. 4 depicts this knowledge organization, which allows the user to employ knowledge scouts that query for and access components of the knowledge base at any or all levels of the ruleset family hierarchy. For example, the user can request the number of *rules* in a given *ruleset* that have exactly two *conditions* with at least 80% support and 80% confidence over the training data. Fig. 5 instantiates this knowledge schema for the two rules with decision High_Blood_Pressure is present.

2.3 Knowledge Visualization via Concept Association Graphs

Users often rely on visualization methods to provide a better understanding of the patterns in the data and knowledge. Visual representations provide more succinctly and comprehensibly the ideas that are important to the

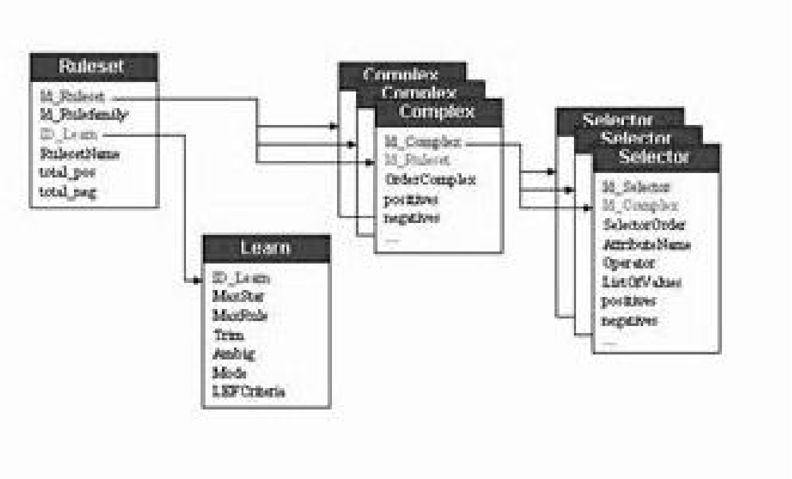

Fig. 4. Schema for the VINLEN inductive database visual interface

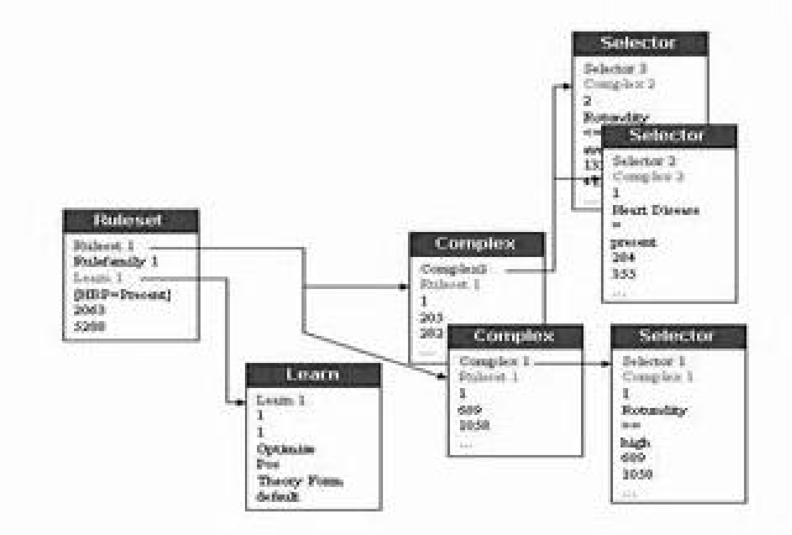

Fig. 5. Instantiated database structure of two rules for the decision High Blood Pressure is present

users. Hence, visualization technology is very much in the spirit of inductive databases. We have developed visualization operators based on the method of *diagrammatic visualization* [9], and on *association graphs* [1,6].

The latter method provides a novel approach to concept visualization. In it, the elements may represent attributes or high-level concepts, with annotated links showing details of the relationships. Fig. 6, for example, presents an association graph built from the rules listed in Fig. 3. The output attributes and decisions are represented by the darker ovals. The input attributes and values are represented by lighter ovals and rectangles, respectively. The square numbered boxes represent rules linking inputs to an output. Each link's width is an indication of the strength of the associated condition (if it connects a rule box to an input node) or of the rule as a whole (if it connects the rule box to a decision node). The strength is based on the consistency (C_p) of the condition or rule.

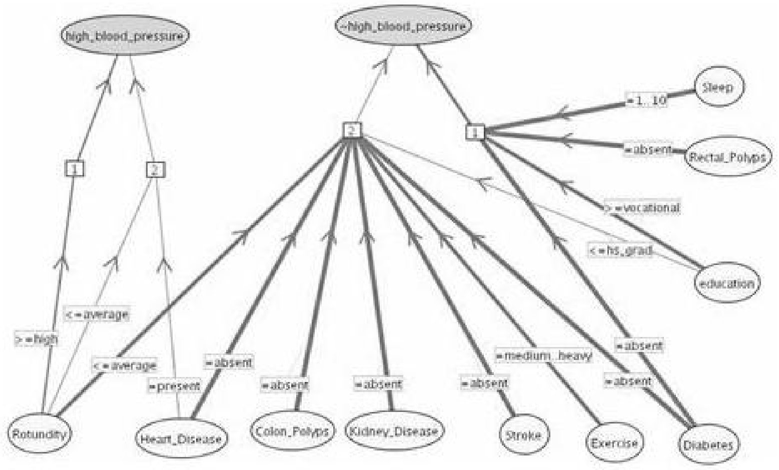

Fig. 6. An association graph representing rules for diagnosing high blood pressure

3 Conclusion

The VINLEN system is still under development, and is planned to include a very wide range of knowledge generation operators and related capabilities. Due to space limitations, this paper concentrated only on the rule generation and concept association graph visualization operators implemented in the current VINLEN system. Other major features already integrated in VINLEN include conceptual clustering, intelligent target data generation (selecting the most promising attributes and data records for completing a specified task in a given amount of time), database manipulation through an SQL client, and a mechanism for creating knowledge query language scripts to guide data exploration tasks.

Initial results indicate the great potential of this research for applications in many areas. Many questions and research issues remain, such as how to integrate symbolic and statistical reasoning capabilities in VINLEN, how to develop a knowledge query language that facilitates the creation of scripts for automated knowledge discovery, how to visualize complex regularities, and how to implement knowledge mining in temporal datasets. Potential approaches to this problem include, but are not limited to the construction of new temporal-oriented attributes that better characterize the target concepts, the characterization of sequential processes through the analysis of their components, and prediction methods based on previous events. To this end, we plan to conduct research on how to introduce the capability of *qualitative prediction* to a temporal database. Such a capability requires a method for inductively generating qualitative hypotheses about future events (e.g., [13]).

An inductive database can be used to build *knowledge scouts*, which are scripts for synthesis and management of target knowledge. During the course of its existence, a knowledge scout builds a model of interests and experiences of the user, and employs that model in synthesizing target knowledge. An important research issue for future study here is how to cope with changes in the users' interests (concept drift).

Summarizing, the main contribution of the VINLEN project is the development of a methodology for a tight integration of a data base, knowledge base, data management operators, knowledge generation operators, a general knowledge query language, and a user-oriented visual interface. Continuing research and development promises to achieve advances in the field of knowledge mining and solutions to real-world problems.

Acknowledgments

The authors thank Michal Draminski and Guido Cervone for their contributions to the implementation of VINLEN, the development of some of the component operators. This research was performed at the Machine Learning and Inference Laboratory at George Mason University. The Laboratory's research activities are supported in part by the National Science Foundation under Grants No. IIS-9906858 and IIS-0097476, and in part by the UMBC/LUCITE #32 grant. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors, and do not necessarily reflect the views of the National Science Foundation (NSF) or the UMBC/LUCITE sponsor.

References

1. Cervone, G., Michalski, R.S. (2003) Concept Association Graphs: CAG1. *Reports of the Machine Learning Laboratory*, George Mason University
2. Finan, T., Fritzson,R., McKay, D., McEntire, R. (1994) KQML as an Agent Communication Language. *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM '94)*, ACM Press
3. Imielinski, T., Virmani, A., Abdulghani, A. (1996) DataMine: Application Programming Interface and Query Language for Data Mining. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 256-261
4. Kaufman, K.A., Michalski, R.S. (1998) Multistrategy Data Mining via the KGL Metalanguage. *Proceedings of the Seventh Symposium on Intelligent Information Systems (IIS '98)*, Marlbork, Poland, pp. 39-48
5. Kaufman, K.A., Michalski, R.S. (1999) Learning From Inconsistent and Noisy Data: The AQ18 Approach. *Proceedings of the Eleventh International Symposium on Methodologies for Intelligent Systems (ISMIS '99)*, Warsaw, pp. 411-419
6. Kaufman, K.A., Michalski, R.S. (2000) A Knowledge Scout for Discovering Medical Patterns: Methodology and System SCAMP. *Proceedings of the Fourth International Conference on Flexible Query Answering Systems (FQAS'2000)*, Warsaw, pp. 485-496
7. Mannila, H. (1997) Inductive Databases and Condensed Representations for Data Mining. *Proceedings of the International Logic Programming Symposium (ILPS'97)*, pp. 21-30
8. Michalski, R.S. (2001) Attributional Calculus: A Representation System and Logic for Deriving Human Knowledge from Computer Data. *Reports of the Machine Learning and Inference Laboratory*, MLI 01-1, George Mason University
9. Michalski, R.S., Kaufman, K.A. (1998) Data Mining and Knowledge Discovery: A Review of Issues and a Multistrategy Approach. In Michalski, R.S., Bratko, I., Kubat, M. (eds.), *Machine Learning and Data Mining: Methods and Applications*, London, John Wiley & Sons, pp. 71-112
10. Michalski, R.S., Kaufman, K. (2000) Building Knowledge Scouts Using KGL Metalanguage. *Fundamenta Informaticae* **40**, pp. 433-447
11. Michalski, R.S., Kaufman, K. (2001) Learning Patterns in Noisy Data: The AQ Approach. In Paliouras, G., Karkaletsis, V., Spyropoulos, C. (eds.), *Machine Learning and its Applications*, Springer-Verlag, pp. 22-38
12. Michalski, R.S., Kerschberg, L., Kaufman, K.A., Ribeiro, J.S. (1992) Mining for Knowledge in Databases: The INLEN Architecture, Initial Implementation and first results. *Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies* **1**, pp. 85-113
13. Michalski, R.S., Ko, H., Chen, K. (1988) Qualitative Prediction: SPARC/G Methodology for Inductively Describing and Predicting Discrete Processes. In Van Lamsweerde, A., Dufour, O. (eds.), *Current Issues in Expert Systems*
14. Morik, K., Brockhausen, P. (1996) A Multistrategy Approach to Relational Knowledge Discovery in Databases. *Proceedings of the Third International Workshop on Multistrategy Learning (MSL-96)*, pp. 17-27
15. Rice, S.P., Roddick, J.F. (2000) Lattice-Structured Domains, Imperfect Data and Inductive Queries. *Proceedings of the Eleventh International Conference on Database and Expert Systems Applications (DEXA 2000)*