Knowledge Discovery from Multiple Databases

James S. Ribeiro, Kenneth A. Kaufman and Larry Kerschberg

George Mason University
4400 University Drive
Fairfax, Virginia 22030-4444
ribeiro@osf1.gmu.edu; kaufman@aic.gmu.edu; kersch@gmu.edu

Abstract

Knowledge discovery systems for databases are employed to provide valuable insights into characteristics and relationships that may exist in the data, but are unknown to the user. This paper describes a methodology and system for performing knowledge discovery across multiple databases. These enhancements have been integrated into the prototype knowledge discovery system called INLEN. The enhancements include the incorporation of primary and foreign keys as well as the development and processing of knowledge segments.

Introduction

During the past several decades, considerable effort has been placed on building computerized databases and on writing applications which exploit that data. An automated system that assists the user in the discovery of knowledge from a database is referred to as a *knowledge discovery system* (KDS). A KDS can include research prototypes such as INLEN (Michalski et al, 1992), AURORA (INIS, 1988), Recon (Simoudis, Livezey, & Kerber, 1994), or recent commercial offerings such as ReMind™, IXL™, or Database Mining Workstation™.

Without a KDS, discovering knowledge from large or complex databases often proves difficult or, in most cases, impossible for the user, due to the amount of data that must be analyzed. Many potential discoveries go unnoticed for lack of appropriate knowledge discovery tools. For example, the term "pre-discovery" was introduced by scientists who had data on a supernova that went undetected by their group, but was later discovered by another group (Jones, 1991). Thus, the ultimate goal of knowledge discovery systems is to minimize pre-discovery by providing the tools to mine ever-increasing and expanding databases in the ongoing quest for knowledge.

Recently, attempts have been made to automate the *knowledge discovery process* by applying techniques from Artificial Intelligence (AI) to databases, forming the interdisciplinary field referred to as Knowledge Discovery in Databases (KDD) (e.g., Piatetsky-Shapiro & Frawley, 1991). A number of algorithms developed within the Machine Learning field of AI to detect patterns in data have been employed in knowledge discovery systems. We will refer to these algorithms as *knowledge discovery algorithms*. Two of the better known algorithms are AQ (Michalski, 1983; Michalski et al, 1986) and ID3 (Quinlan, 1986). AQ presents its knowledge in the form of IF-THEN rules while ID3 uses decision trees as a knowledge representation structure.

One prototype knowledge discovery system, developed at George Mason University, is INLEN (Michalski et al, 1992). INLEN, whose name is derived from **in**ference and **le**ar**n**ing, contains the AQ algorithm as one of its knowledge discovery algorithms. It can provide valuable insights into characteristics and relationships that may exist in the data, but are unknown to the user.

To date, INLEN has been used only for discovery in small single databases. Typically, the user hand-crafts a database for discovery from an existing database or multiple databases by selecting a subset of the data for discovery and defining preference criteria for knowledge discovery to be performed on this dataset. We are extending this approach to knowledge discovery in multiple databases by applying INLEN's methodology to individual relations or databases, and then further processing this discovered knowledge. Our approach increases the effectiveness of the overall knowledge discovery process for the following reasons:

1. Knowledge discovery on existing operational databases avoids the set-up time required to hand-craft a database from multiple databases. This approach is time consuming, especially in large database environments which contain many attributes, records, and tables. Additionally, the various owners of the data may not be willing to release their data in full for such recombination.

2. We avoid the computational cost of performing knowledge discovery on a universal relation which will have many more attributes and records than required by the proposed approach. Many of the attributes in a larger set will be irrelevant to the current knowledge

™ ReMind is a trademark of Cognitive Systems, Inc.
™ IXL is a trademark of IntelligenceWare, Inc.
™ Database Mining Workstation is a trademark of HNC, Inc.

discovery task, and as such will make the extraction of useful knowledge more difficult.

3. Increasingly, very large databases will be "published" on the Internet through the World Wide Web or by comparable means. These will provide more data than the KDS systems will be capable of processing. Therefore, new techniques are needed that allow the KDS to query, retrieve, process, and learn from subsets of the entire database. The techniques presented in this paper are a step in this direction.

The next section provides some background information on the INLEN system and its principal knowledge discovery algorithm AQ. An example problem domain is discussed followed by our methodology for knowledge discovery from multiple databases. A detailed example is provided and this is followed by a conclusions section.

Background

INLEN is a knowledge discovery system whose principal knowledge discovery algorithm, AQ, uses inductive inference to learn *decision rules* from examples. AQ also supports rules being integrated into other knowledge forms. In this section we provide an overview of the overall INLEN architecture and briefly review the output of the AQ algorithm.

INLEN Architecture

Figure 1 illustrates the high-level architecture of the INLEN system and is a more refined design than the original INLEN design presented in (Kaufman, Michalski, & Kerschberg, 1991).

Data is stored in the relational database (DB) and can be manipulated by any of the *data management operators* (DMOs). The knowledge base (KB) is used for storing *knowledge segments* and can be manipulated by the *knowledge management operators* (KMOs). Knowledge segments include rulesets, constraints, and schema information. The *knowledge generation operators* (KGOs) interact with both the knowledge and data bases. They invoke various machine learning programs to discover new knowledge. One such operator, GENRULE, can apply the AQ algorithm in order to produce knowledge in the form of predicate calculus-like expressions which are then stored in the knowledge base. The discovered knowledge is also displayed to the user in the form of IF-THEN rules. The discovered rules may cause the user to end the knowledge discovery process and take some action or to begin a new knowledge discovery session, possibly modifying the original data in the database by using DMOs or KGOs.

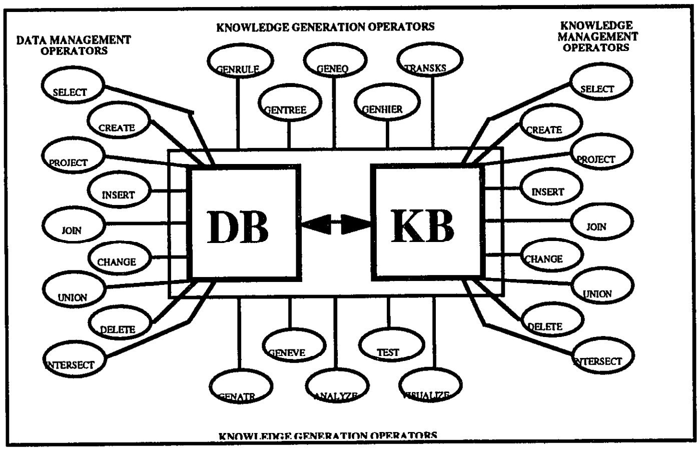

Figure 1: High-level architecture of INLEN

Within the INLEN architecture of Figure 1, the work described in this paper falls within the ANALYZE knowledge generation operator. Specifically, it implements the RELKS (Relate Knowledge Segments) operator within the ANALYZE KGO group (Michalski et al, 1992).

AQ

AQ is an inductive learning program with capabilities for incremental learning and constructive induction. *Incremental learning* allows AQ to add new learning examples and modify or generate new rules based upon the existing knowledge and these new examples. *Constructive induction* allows AQ to generate new variables (also referred to as attributes) not present in the input data and to use these new variables to produce better rules.

AQ learns decision rules by performing inductive inference over a set of training examples. Training examples are given to AQ in the form of *events* that belong to different decision classes. A *decision class* is a concept based upon the value of a *decision variable*. Events belonging to a given decision class are termed *positive examples* of that class while those that do not belong are termed *negative examples*. For example, in earlier work (Michalski et al, 1992) we analyzed a database of scientific publications written by scientists in the Commonwealth of Independent States (CIS). This database was referred to as the CIS Authors database, or CISA. In the CISA database we had a decision variable called INSTITUTE, which represented the research institute affiliation of the publishing author. Thus, one decision class would be authors whose research institute was LITMO (Leningrad Institute of Materials and Optics). The positive examples of this class would be all tuples with values of "LITMO" for decision variable INSTITUTE, with all other events with a known institute being negative examples. Other attributes from the CISA database that will be referred to in this example include PUBYR, which is simply the publication year of the paper.

All of these concepts are expressed in the variable-valued logic language called VL_1 (Michalski, 1975). In VL_1, a *selector* is an expression of the form:

<term> <rel> <reference>

where <term> is a variable, an arithmetic expression of constants and variables, or a conjunction of terms; <rel> is one of the relational symbols <, <=, =, <>, >=, >; and <reference> is a constant value, a disjunction of values, or a range of values.

For example, the selector:

[INSTITUTE = LITMO, IREE],

identifies all publications from authors with research institute affiliation of LITMO or IREE.

A *complex* is simply a conjunction of selectors, while a *cover* is a disjunction of complexes. A cover of a class is a set of rules that is satisfied by all positive examples of the class and by no negative ones.

Again, in the CISA database, the complex:

[INSTITUTE = LITMO, IREE] & [PUBYR = 1985],

specifies all publications from authors with research institute affiliation of LITMO or IREE and published in the year 1985.

Problem Domain

The motivation for this research lies in the authors' desire to apply knowledge discovered in one database to enhance the process of knowledge discovery in other databases. Our earlier work involved single databases and proceeded by first grouping all attributes into a universal relation and then beginning the knowledge discovery process. However, there are many problem domains that involve multiple databases with many more attributes and records than the databases used in our earlier experimental work. Such a domain is discussed below.

CIA World Factbook

This work uses three databases of facts created from the 1993 Central Intelligence Agency (CIA) World Factbook (CIA, 1993). We included in our domain the 182 countries belonging to the United Nations (U.N.) plus the eight other independent countries not belonging to the U.N. (Andorra, Holy See, Kiribati, Nauru, Serbia and Montenegro, Switzerland, Tonga, and Tuvalu). The three databases we used for this research were PEOPLE, GEOGRAPHY, and ECONOMY, all linked by the common attribute of CountryName.

The PEOPLE database describes demographic characteristics in the 190 countries and contains the following attributes:

1. Population growth rate (PopGrRate) : Annual population growth rate as a percentage.
2. Birth rate (BirthRate) : Number of births per 1000 population.
3. Death rate (DeathRate) : Number of deaths per 1000 population.
4. Net migration rate (NetMigRate) : Number of immigrants minus emigrants per 1000 population.
5. Infant mortality rate (InfMortRate) : Number of infant deaths per 1000 population.
6. Life expectancy (LifeExp) : Average life expectancy at birth in years.
7. Total fertility rate (FertRate) : Number of children born per woman.
8. Literacy rate (Literacy) : Percentage of total population who are literate.
9. Religion (Religion) : Country's predominant religion (50% or greater), otherwise encoded as mixed.

The GEOGRAPHY database describes various geographic features of the 190 countries and contains the following attributes:

1. Climate (Climate) : The climate of the country (e.g., arid, temperate).
2. Arable land (ArableLand) : Percentage of the country's land mass that is arable.
3. Meadows and pastures (MdwsPstrs) : Percentage of the country's land mass that consists of meadows or pastures.
4. Forests and woodlands (FrstWdlnds) : Percentage of the country's land mass that consists of forests and woodlands.

The ECONOMY database contains information describing the economies of the 190 countries and consists of the following attributes:

1. National product real growth rate (NPGrowthRate) : The country's national product real growth rate as a percentage.
2. National product per capita (NPPerCapita) : The country's national product per capita expressed as a dollar amount.
3. Inflation rate (InfRate) : Inflation rate for consumer prices as an annual percentage.
4. Unemployment rate (UnEmpRate) : Unemployment rate for the country expressed as a percentage.

Methodology

In order to extend INLEN for knowledge discovery across multiple databases we addressed the following three areas:

1. Include information on *primary* and *foreign keys*. A primary key field is designated by the database designer and uniquely identifies each record in the relation. A foreign key field occurs in another relation and has a domain equivalent to that of the primary key field, allowing it to refer back to the primary key field relation.
2. Integrate all discovered knowledge for each database into *knowledge segments.*
3. Apply the information from the first two areas in order to enable the discovery of knowledge from multiple

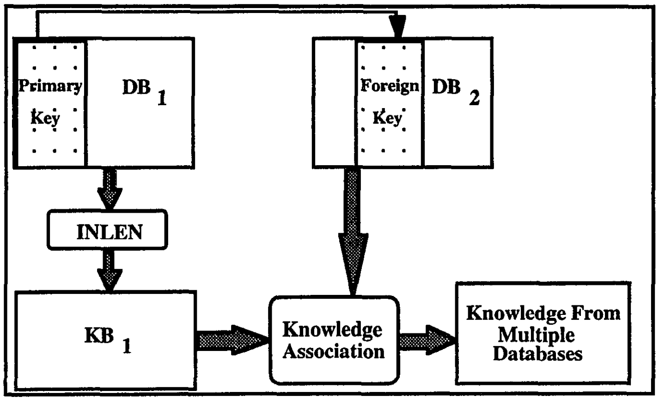

Figure 2: The architecture of this method

databases. It should be noted that while we present this technique as a method for making discoveries across multiple databases, it can also be applied to discoveries across different relations within a single database.

Figure 2 shows the architecture of the core of this method (INLEN and other knowledge discovery methods can also be applied at intermediate stages other than the one shown) The enhancements in each of the three areas are addressed in further detail in the next three sections.

Primary and Foreign Keys

Data sets in earlier experiments with INLEN did not include attributes that were primary keys. This was due to the nature of the programs that make up INLEN's knowledge generation operators. For example, the AQ algorithm discovers rules to discriminate between classes of examples based on the features provided as input. Hence if a key field is included, AQ will be apt to find the rule: Class is <class-name> if <key-attribute> is <set-of-key-values-in-that-class>. This rule performs extremely well according to the program's biases toward completeness, consistency and simplicity, but it is also completely useless from a knowledge discovery viewpoint, for it tells us nothing we did not already know. A feature of INLEN allows a user to set parameters so that the key field will almost never appear in rules, but it is easier and more natural simply to omit keys from the input data

Advanced AI learning programs are tuned to classify collections of related data by lumping the information together, this being *the most convenient form for the discovery system.* The key fields of a relation are therefore not included in the set of learning attributes. However, from the database management point of view, key fields uniquely identify tuples in a relation, so that one may correlate information across multiple databases by joining relations on the key attributes. Database systems will often distribute the information into a form that is convenient for *user access and understanding* . These two properties of AI learning algorithms and database join algorithms, when taken together, form the basis for this novel approach.

In order to perform knowledge discovery across multiple databases, primary and foreign keys must be included, since they serve as the links across the databases. INLEN's knowledge segments (see next section) include references to the particular records in the INLEN database on which they were based. The INLEN database (more precisely, the view of the data that INLEN uses) has been enhanced to include references to the key fields in the host database. By traversing these links, one can access the original records from which a piece of knowledge was generated and integrate them with the knowledge learned from other databases for further discovery. A detailed example is given in the next section.

Knowledge Segments

The discovered knowledge generated by INLEN from a particular database is stored in knowledge segments. Knowledge segments include information such as database name, decision variable, decision value, rule head, rule body, and number of covered examples. A knowledge segment can represent a complex, and link the complex to primary keys in the database. The knowledge segments generated for one database are grouped together into a knowledge base. The CLIPS expert system development tool (Giarratano, 1993) is used to store and manipulate these knowledge segments.

Figure 3 shows four components of a knowledge segment, Database, Class, Complex, and Selector. The items to the left contain more general information, while the items to the right are more specific. For example, each Database usually contains information on many Classes and each Class usually contains information on many Complexes. Similarly, many Selectors usually make up a Complex.

Knowledge segments are typically processed based on decision variables and the various decision values. For example, the decision variable Religion within the PEOPLE database can be accessed by any of its values (e.g., Protestant, mixed). At this level we can also access discovered knowledge, knowledge acquisition method, rule strength (based on Class, Complex and Selector coverages), and other background information important to further knowledge processing.

Discovery from Multiple Databases

Our approach is to combine the information about foreign and primary keys from the database schemas with the knowledge segments for discovery from multiple databases. This method enables knowledge discovery from

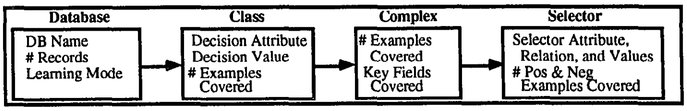

Figure 3: Knowledge segment components

multiple databases without the need to construct new databases (e.g., universal relations) and then perform knowledge discovery within them.

In order to perform knowledge discovery on multiple databases, our method must first have a database that contains an attribute which is a foreign key that maps to a primary key attribute in another database. The foreign key values extracted from this database form a *target class* for association with the primary key database. At this point we can proceed by examining all knowledge segments in the knowledge base of the databases(s) containing the primary key, and selecting those knowledge segments which satisfy a multiple database association criterion for the foreign key values. New knowledge segments can be formed from this information and stored in the knowledge base for the database containing the foreign key.

We define as a new measure a multiple database association criterion, the *knowledge association coefficient* (KAC), for selecting knowledge segments as:

$$KAC = coverage \times specificity$$

where: *coverage* is defined as the percentage of the keys in the target class that are covered by the complex, and

specificity is defined as the percentage of keys covered by the complex that belong to the target class.

Example of KDD in Multiple Databases

To demonstrate the method we will use representative knowledge generated from the GEOGRAPHY database and illustrative data from a hypothetical oil spills database. The following complex is representative of the knowledge INLEN discovered from GEOGRAPHY:

Climate is cold if: (3 examples)

A. 1. ArableLand < 10% *(Percentages of arable land*
2. MdwsPstrs < 10% *and meadows and pastures*
3. FrstWdlnds > 60% *are both under 10%, while the country is at least 60% forested)*

KEYS: Canada Russia Finland

The OILSPILLS database contains the fields: Id, CountryName, Year, and Tonnage. Projecting on CountryName we find only three countries: Canada, Russia, and Finland.

A traditional way to learn about relationships between oil spills and geography would be through a constructive induction approach, in which one would add a new attribute to the GEOGRAPHY database, a Boolean variable called HasOilSpills representing whether there were any oil spills in a country, and then repeat the learning from the modified database. In this example, relationships involving spill potential would be found, but this would be relatively computationally costly, and might necessitate the replacement of the entire GEOGRAPHY knowledge base in order to incorporate the effects of the new attribute.

Our approach is to examine all knowledge segments in the GEOGRAPHY knowledge base, in search of primary key coverage similar to that found in the OILSPILLS database. In the example shown, we find that the complex for cold climate perfectly covers all three of the countries involved in oil spills, and covers no others, thereby obtaining a KAC score of 1 (coverage = 3/3 and specificity = 3/3). Linking the countries with oil spills to this complex is a *deductive generalization*, suggesting that either the outcome (Climate is cold) or the three conditions that predicate this outcome may serve to better explain a country's propensity for oil spills.

Now assume that the Climate complex above covered 4 countries – Norway as well as Canada, Russia and Finland for a KAC of 0.75 (coverage = 3/3 while specificity = 3/4). If this complex still satisfactorily covers the oil spill countries according to our KAC threshold, then we can express our knowledge with an exception:

HasOilSpills is true if

A. 1. ArableLand < 10%
2. MdwsPstrs < 10%
3. FrstWdlnds > 60%
4. CountryName ≠ Norway.

Several inferences can be made in an attempt to explain Norway's absence from the list of countries in the OILSPILLS database: 1) Norway could actually have had spills, but managed to suppress public information about these occurrences; 2) This finding could represent a prediction that Norway might suffer an oil spill in the near future; 3) There could be a further reason why Norway has avoided oil spills, one that could perhaps be uncovered by further data exploration and could possibly be applied by the three countries with similar climactic conditions in order to reduce their likelihood of having oil spills; 4) It might be the case that there is really no causal relationship between cold climate and oil spills – the location of the spills could have been due to some other factor (such as drilling location) or purely coincidental. It should be emphasized that discovered knowledge may suggest a causal relationship that does not exist. In any case, an oil industry data analyst would determine the likely conclusion and optimal course of action.

This example demonstrates a situation in which this method can be especially useful, in which the database with the primary key is a repository for general background knowledge applicable to many domains, while the database containing the foreign keys holds information on a specific relevant field. The process of comparing the foreign keys to the knowledge learned from the general database selects the factors in the general data that are likely to be relevant to the information in the specific domain.

Conclusions and Future Work

When the primary key in one database appears as a field in another database, it is possible to discover knowledge linking the two databases without having to actually combine their data. Our method uses the values of the first database's key field present in the second database as selection criteria in a knowledge base discovered from the first database alone. Using this method, deductive generalization and abductive reasoning are applied instead of constructive induction.

The methodology presented has been incorporated into a large-scale knowledge discovery system as one of a suite of many knowledge generation operators. As was described above, the World Factbook databases provide general information that can be combined with the domain-specific information from various databases that use CountryName as a foreign key. We are in the process of accessing such databases to validate the methodology. We also intend to refine our usage of the knowledge association coefficient.

The example presented here involved a case in which all of the records in the foreign key relational table were used to define a class (in this case countries with oil spills). An extension of this method would allow a subset of the records in that table, determined by a query or description, to define that class, while the remainder of the records are used as negative examples, a technique that was explored in (Yoon & Kerschberg, 1993). For instance, to learn about "high-tonnage" oil spills, we would look for rules that covered countries that not only closely matched the countries with high-tonnage spills, but also had a low KAC with respect to the low-tonnage oil spill countries.

A second area for future research involves the logical combination of knowledge segments into larger pieces of knowledge that may better correlate with the second database than its individual component parts. To thwart the potential combinatorial explosion, heuristics must be developed to select the operators and knowledge segments that show the most promise of fruitful combination.

Finally, we plan to investigate loosening the primary key/foreign key requirement and to develop techniques which can be applied to databases with domain equivalent attributes. An ultimate goal is to be able to navigate networks of interrelated databases in order to discover patterns only visible using knowledge from each of the components. For instance, prescription fraud in the medical domain may only become apparent when data sets pertaining to doctors, patients, clinics and pharmacies are viewed together. The discovery of low-level knowledge in each of these databases and some relationships among them may make early detection significantly easier.

Acknowledgments

The authors thank Eric Bloedorn and Stan Wozniak for their comments and criticism of the earlier drafts of this paper.

This research was conducted in the Center for Machine Learning and Inference (MLIC) and the Center for Information Systems Integration and Evolution (CISIE) at George Mason University. MLIC research is supported in part by the National Science Foundation under Grants No. CDA-9309725, IRI-9020266, and DMI-9496192, in part by the Advanced Research Projects Agency under Grant No. N00014-91-J-1854, administered by the Office of Naval Research, and the Grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research, and in part by the Office of Naval Research under Grant No. N00014-91-J-1351. CISIE research is supported in part by the Advanced Research Projects Agency under Grant No. N00014-92-J-4038, administered by the Office of Naval Research.

References

Central Intelligence Agency. 1993.*1993 World Factbook*.

Giarratano, J. C. 1993.*CLIPS Users Guide*, Artificial Intelligence Section, Johnson Space Center.

International Intelligent Systems, Inc. 1988.*User's Guide to AURORA 2.0: A Discovery System*, Fairfax, Virginia.

Jones, A. K. 1991. The scientific data decade, *IEEE Computer*, 24,(9):102-103, September 1991.

Kaufman, K.; Michalski, R. S.; Kerschberg, L. 1991. Mining For Knowledge in Data: Goals and General Description of the INLEN System, in Piatetsky-Shapiro, G. and Frawley, W. J. (Eds.), *Knowledge Discovery in Databases*, 449-462, AAAI Press.

Michalski, R.S. 1975. Variable-Valued Logic and its Application to Pattern Recognition and Machine Learning, *Computer Science and Multiple-Valued Logic Theory and Application*, D. C. Rine (Ed.), North-Holland Publishing Company, 506-534.

Michalski, R.S. 1983. A theory and methodology of inductive learning, in *Machine Learning: An Artificial Intelligence Approach*, Michalski, R. S.; Mitchell, T. M.; Carbonell, J. G. (Eds.), Morgan Kaufmann, San Mateo, CA, 83-129.

Michalski, R. S.; Mozetic, I.; Hong, J.; and Lavrac, N. 1986. The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, *Proceedings of AAAI-86*, 1041-1045.

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, *Journal of Intelligent Information Systems: Integrating AI and Database Technologies*, 1(1): 85-113, August.

Quinlan, J. R. 1986. Induction of Decision Trees, *Machine Learning*, 1(1):81-106.

Piatetsky-Shapiro, G.; and Frawley, W. J. (Eds.). 1991. *Knowledge Discovery in Databases*, AAAI Press, Menlo Park, CA.

Simoudis, E.; Livezey, B.; and Kerber, R. 1994. Integrating Inductive and Deductive Reasoning for Database Mining, *Proceedings of KDD-94*, Seattle, Washington, 37-48.

Yoon, J.P. and Kerschberg, L. 1993. A Framework for Knowledge Discovery and Evolution in Databases, *IEEE Transactions on Knowledge and Data Engineering*, 5(6).