SEEKING KNOWLEDGE IN THE DELUGE OF FACTS

by

R. S. Michalski

Fundamenta Informaticae, Vol. 30, pp. 283-297, 1997.

Fundamenta Informaticae 30 (1997) 283–297
IOS Press

Seeking Knowledge in the Deluge of Facts

Ryszard S. Michalski
George Mason University, Fairfax, VA and
Institute for Computer Science
Polish Academy of Sciences, Warsaw, Poland
michalski@gmu.edu

Abstract. An enormous proliferation of computer technology in modern societies has produced a severe information overload. The navigation through the masses of available information in order to derive desired knowledge is becoming increasingly difficult. This creates a demand for intelligent systems capable of assisting data analysts in extracting goal-oriented knowledge from large volumes of data. This paper presents a multistrategy methodology and a system, INLEN, for knowledge discovery in large relational databases. The system integrates data base, knowledge base and machine learning technologies. It offers a data analyst an integrated interface and a wide range of *knowledge generation* operators, as described in the Inferential Theory of Learning. Presented ideas are illustrated by results from experiments with INLEN.

1. Introduction

The current information age is characterized by an enormous proliferation of data generated and stored about all kinds of human activities. An increasing proportion of this data flow is recorded in the form of computer databases. This makes the data easily accessible and analyzable by computer technology. The explosive growth of databases has not, however, been matched with a parallel development of powerful methods for deriving knowledge from them. Although existing data analysis tools are very useful and important, they continue to be oriented primarily toward extraction of quantitative statistical characteristics. These tools include determination of statistical correlations, cluster analysis, numerical taxonomy, regression analysis, stochastic models, times series analysis, nonlinear estimation techniques, relaxation techniques, various curve-fitting methods, and other.

The above conventional techniques seem to be particularly useful for such tasks as creation of statistical data summaries, fitting equations to data, revealing data organization on the basis of given numerical measures, developing mathematical data models, etc. Their results facilitate useful data interpretations, and can help to gain important insights into the processes that generated the data. These interpretations and insights constitute the ultimate knowledge sought for by a data analyst. Yet, they have to be developed by a human data analyst. As the quantity of available data increases, the complexity of these processes may outstrip capabilities of a human data analyst.

Although, traditional techniques have important practical applications in data mining, they suffer from inherent limitations. For example, a statistical data analysis technique can discover a correlation between given variables, but it cannot produce a conceptual characterization or causal explanation why such a correlation exists. Neither it can develop a

justification of this correlation in terms of higher-level concepts or laws. A statistical analysis can determine a central tendency and variability of various properties, and a regression analysis can fit a complex curve to a set of data points. These techniques cannot, however, develop a qualitative characterization of the data points in abstract terms, or draw an analogy between this characterization and some regularity in another domain. They cannot generate knowledge from past experience and use it for solving new problems.

A numerical taxonomy technique can create a classification of entities, and specify a numerical similarity among the entities assembled into the same or different classes. It will not hypothesize, however, reasons for the entities being in the same class, or build qualitative descriptions of the classes created. Attributes that define the similarity, as well as the similarity measures, must be defined by a data analyst in advance. These techniques cannot generate relevant attributes and appropriate similarity measures by themselves. All the above problems require complex symbolic reasoning processes that relate high level concepts and goals of the analysis to available quantitative measures, and able to performs data transformations relevant to these goals.

In view of the above, there has been a fast growing interest in developing new tools for data mining and knowledge discovery, e.g., Michie, 1991; Pawlak, 1991; Piatetsky-Shapiro, 1991 and 1993; Michalski et al., 1992, Zembowicz and Zytkow, 1992; Zytkow, 1992; Ziarko,1993; Fayyad and Uthurusamy, 1995. This paper describes a new methodology for assisting data analyst in performing some of the above mentioned data exploration tasks.

2. Applying Machine Learning To Data Mining

2.1. Rule learning from examples

This paper discusses the application of symbolic methods of machine learning and discovery to problem of data mining. It shows that these methods are able to perform new type of operations on data, and therefore widen the scope of data exploration tasks that can be automated or semi-automated. In particular, they can perform *conceptual data analysis*, that is, derive high-level data descriptions and discover qualitative patterns in data. Below is a brief review some of these methods in the context of the data mining applications.

One class of machine learning methods that are potentially useful for data mining are based on methods for inductive learning from examples. Given a set of examples of different classes (or concepts), and problem relevant knowledge ("background knowledge"), an inductive learning method hypothesizes a general description of each class. The description is usually expressed as a set of decision rules or as a decision tree.

A decision rule can have different forms; here we will assume the following form:

$$\text{CLASS} \Longleftarrow \text{CONDITION},$$

where CLASS denotes a class or a concept that is assigned to an entity, if that entity satisfies the CONDITION. The CONDITION is a conjunction of elementary conditions on the values of single attributes, or a disjunction of such conjunctions (a DNF form). Here, we also assume that if the CLASS needs a disjunctive description, then several conjunctive rules are associated with the same CLASS. For example, Figure 1 gives an example of a disjunctive description of *Class 1* in the form of two rules.

Class 1 $\Longleftarrow$ jacket color is red, green or blue &
head shape is round or octagonal

Class 1 $\Longleftarrow$ head shape is square and jacket color is yellow

Figure 1. A two-rule description of Class 1

These rules characterize a class of robot-figures used in the EMERALD system of learning and discovery programs. Paraphrasing, "A robot belongs to Class 1, if the color of its jacket is red, green or blue, and his head is round or octagonal; or, alternatively, if the color of its jacket is yellow and its head is square."

In a decision tree representation, nodes correspond to attributes, branches stemming from the nodes correspond to attribute values, and leaves to individual classes (e.g., Quinlan, 1986). A decision tree can be simply transformed into an equivalent set of decision rules (a rule set) by traversing all paths from the root to individual leaves. The opposite process, that is, transforming a rule set into a decision tree is not so direct. The reason is that a rule representation is more powerful than a decision tree, in the sense that the decision tree that is equivalent to a given rule set may contain superfluous attributes and thus be more complex (e.g., Michalski, 1990). It should also be noted that the decision tree derived from decision rules is often simpler than the decision tree derived directly from examples (Imam and Michalski, 1993).

The EMERALD system, mentioned above, combines five programs exhibiting different learning and discovery capabilities (Kaufman, Michalski, and Schultz, 1989; and Kaufman, Schultz and Michalski, 1991). These capabilities include decision rule learning from examples, learning distinctions between structures, conceptual clustering, predicting object sequences, and deriving equations characterizing data about physical processes. The rules in Figure 1 were generated by the rule learning program (version AQ-15; Michalski, Hong and Mozetic, 1986) from a set of "positive" and "negative" examples of robot-figures.

Most inductive rule learning methods learn *attributional* descriptions of entities in a class, i.e., descriptions that involve only binary or multiple-valued attributes. Some methods learn *structural descriptions*, which characterize entities in terms of both, attribute values, as well as relationships that hold among components of the entities. Such relationships are represented by multi-place predicates (Michalski, 1983). For data mining, most directly applicable are programs for learning attributional descriptions, because typical databases characterize entities in terms of attributes.

The input to an attributional learning program include a set of examples for each decision class, and background knowledge relevant to the learning problem. The examples are in the form of vectors of attribute-value pairs associated with a given decision class. In many cases background knowledge (BK) is limited to the information about the legal values the attributes, their type (the scale of measurement), and the *preference criterion* for choosing among candidate hypotheses. Such a criterion is defined by the user in advance. In addition to BK, a learning method may have a *representational bias*, e.g., may constrain the form of descriptions to only a certain type of expressions, e.g., single conjunctions, decision trees, sets of conjunctive rules, DNF expressions, etc. In some methods, BK may include more information, e.g., constraints on the interrelationship between various attributes, rules for generating higher level concepts or attributes, and an initial hypothesis (e.g., Michalski, 1983). Learned rules are usually *consistent* and *complete* with regard to the input data. This means that they completely and correctly classify all the original "training" examples. Section 4 presents example solutions from the inductive concept learning program AQ 15. In some applications, especially those involving learning rules from noisy data or learning *flexible* concepts (Michalski, 1990), it may be advantageous to learn descriptions that may be incomplete and/or inconsistent (Bergadano et al, 1990).

Attributional descriptions can be easily visualized by mapping them into a set of cells in a certain diagram. Such diagram is a planar representation of a multidimensional space spanned over the set of attributes (Michalski, 1978; Wnek et al., 1990). For example, Figure 2, shows a diagrammatic visualization of the rules from Figure 1.

The diagram was generated by the visualization program DIAV (Wnek, et al., 1990). Each cell in the diagram represents one combination of values of the attributes. For example. the cell marked by an X represents the vector: (HeadShape=S, Holding=S, Jacket Color=R,

IsSmiling=F). The four darker-shaded areas, marked Class 1 (A), represent rule A, and the lighter-shaded area, marked Class 1 (B), represents rule B. In such a diagram, conjunctive rules correspond to regular arrangements of cells and can be easily recognized (Michalski, 1978). It should be noted that this example is used only for illustration. In real-world scenarios, the number of attributes and the size of the data set may be very large. However, the same regularities apply.

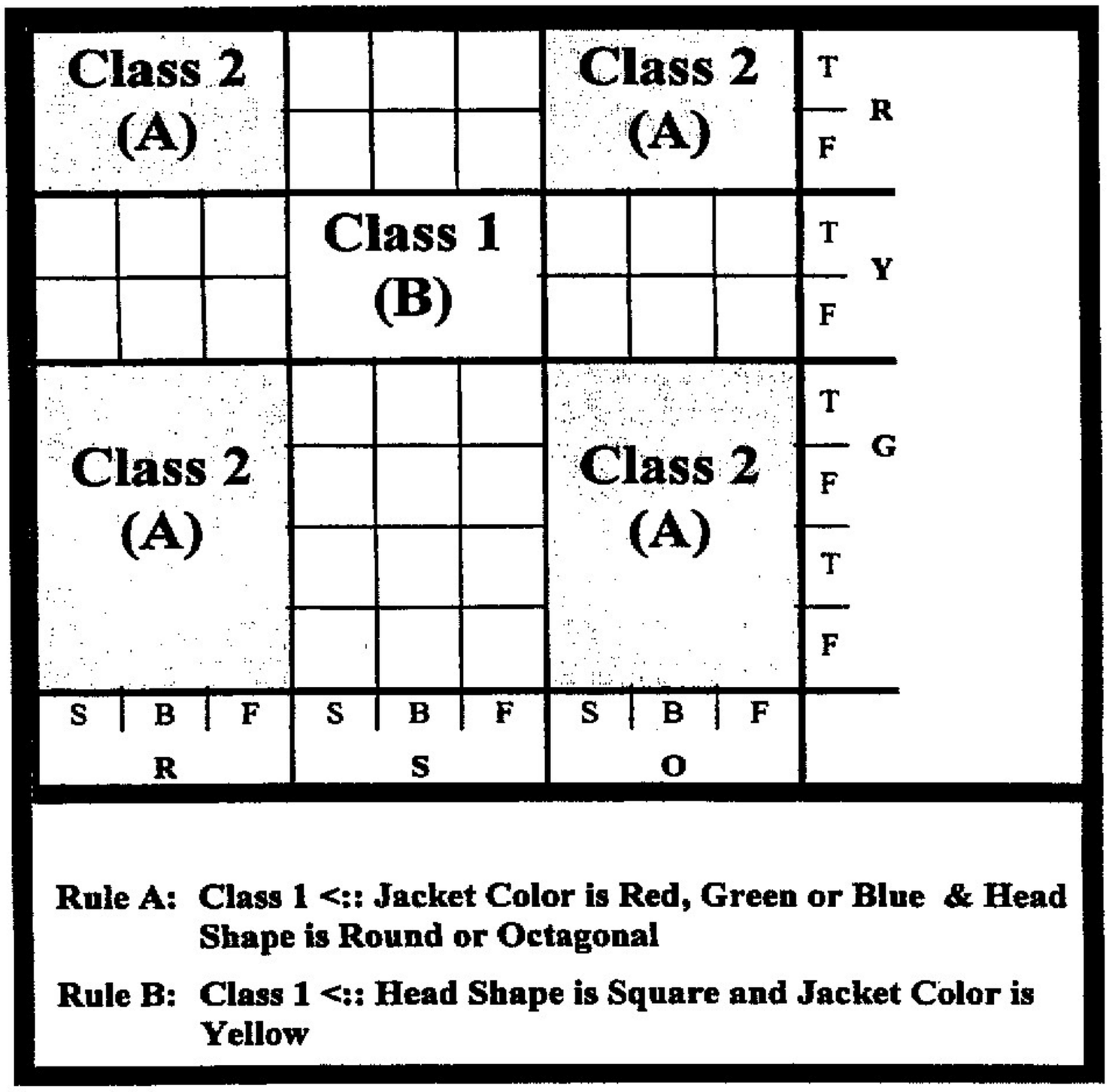

Figure 2. Visualization of rules from Figure 1

The diagrammatic visualization can be used for displaying the *target concept* (i.e., the concept to be learned), the training examples (the examples and counter-examples of the concept), and the actual concept learned by a method. By comparing the target concept with the learned concept, one can determine the *error area*, i.e., the area containing all examples that would be incorrectly classified by the learned concept. The diagrammatic visualization method can illustrate any kind of attributional learning process. Since rows of data tables can be viewed as points in a multidimensional space, this visualization technique can be useful for representing data and learned symbolic descriptions.

A program for learning concept descriptions from examples can be used for two classes of data mining problems:

- determining differences between different groups of entities in a data set (i.e., learning a *discriminate* concept description). Such differences are expressed as symbolic descriptions or rules.
- developing descriptions characterizing one or more groups of entities (i.e., learning a *characteristic* concept description).

Section 3 will illustrate these two types of descriptions. More advanced problems in the area of learning concepts from examples include:

- *Learning from incorrect data*, i.e., learning from examples that may contain of errors or noise.
- *Learning from incomplete data*, i.e., learning from examples in which values of some attributes are unknown.
- *Learning flexible concepts*, i.e., learning concepts that lack precise definition and are context-dependent (Michalski, 1990).

2.2. Conceptual clustering

Another class of methods developed in symbolic machine learning is concerned with developing a classification of a given set of entities. The problem is similar to that considered in traditional cluster analysis, but is defined in a more general way. Given a set of attributional descriptions of some entities, a language for characterizing classes of entities (concepts), and a cluster quality criterion, the task is to split entities to classes that maximize the cluster quality criterion, and generate symbolic descriptions of these classes. Thus, a conceptual clustering method seeks not only a classification (a dendrogram) but also a symbolic description of the proposed classes (clusters). In determining the quality of the classification, the properties of the class descriptions are taken into consideration.

A conventional ("similarity-based") clustering method groups objects into classes on the basis of a "similarity" measure that is a function of properties (attribute values) of the objects being compared O_1 and O_2:

$$Similarity(O_1, O_2) = \text{f}(\text{properties}(O_t), \text{properties}(O_2))$$

In contrast, a conceptual clustering program clusters objects on the basis of *conceptual cohesiveness* that is a function of not only properties of the objects, but also of the set of concepts C (specified by the language for characterizing classes of objects), and of the environment E (a set of neighboring examples):

$$Conceptual\_cohesiveness(O_1, O_2) = \text{f}(\text{properties}(O_1), \text{properties}(O_2), \text{C}, \text{E})$$

The conceptual_cohesiveness takes into consideration the "fit" of cluster description to the data, the simplicity of the description, and some other elementarily criteria (Michalski, Stepp and Diday, 1981). Section 3 gives an illustration of conceptual clustering.

2.3. Other symbolic operators on data

Methods for learning rules from examples usually assume that the examples are expressed in terms of attributes that are given a priori. These attributes must be sufficiently relevant to the problem, otherwise, the resulting rules will be poor. One important advantage of symbolic methods is that they can relatively easily determine irrelevant attributes. In these methods, an attribute is irrelevant or weakly relevant, if there is a complete and consistent class description that does not use this attribute. Inductive learning programs such as rule-learning AQ or decision tree learning ID3 can relatively easily cope with large number of irrelevant attributes.

If very many irrelevant attributes are present in a data set, the speed of a rule learning program is affected. In such a situation, one can employ an operator that determines the most relevant attributes in the data for the given learning task (e.g., Quinlan, 1986). Only these attributes are used in the learning process.

There can also be many examples of the same class, more than necessary for successful learning. In a such situation, one may apply an operator that selects the most representative

examples of a given class. A method for determining such examples is described in (Michalski and Larson, 1978).

In many applications, it is not easy to determine a priori what attributes are most relevant to the problem at hand. The original attributes are usually dictated by the available measurements. In such a situation, one may apply an operator that searches for new attributes that represent certain functions or transformations of the original attributes (Bongard, 1970). Methods for designing such operators are considered in the area of *constructive induction* (Michalski, 1983). For example, constructive induction programs, AQ 17-HCI and AQl7-DCI, can generate new attributes by combining initially given attributes in may different ways (Michalski, Bleodorn and Wnek, 1991), or by detecting patterns in decision rules (Wnek and Michalski, 1993).

3. Illustrating Data Mining Operators Via Data Table

Many symbolic data mining operators can be illustrated by a data table (Figure 3). The columns in such a table correspond to the initial attributes selected to characterize given entities. Each attribute is assigned a *domain* and a *type*. The domain is the set of all possible values that an attribute can take on, which may include "?" ("unknown") and N/A ("not applicable"). The type defines the order of the values in the domain (the scale). The attributes can be of nominal type (no order), linear type (total order), and structured type (a hierarchical order).

Rows in the table correspond to individual entities characterized by attributes assigned to columns. An entry in the table can thus be a specific value of an attribute, a symbol ? (meaning that value is unknown), or a symbol N/A, if the given attribute does not apply to the given entity. For example, the "color" attribute applies to physical objects, but does not apply to abstract entities such as "freedom".

One problem of data analysis is to determine if a designated ("output") attribute in a table depends on other attributes. A more complex problem is to determine the form of this relationship. The latter problem becomes a concept learning problem, if the output attribute is nominal (its value set is unordered). In such a case, values of this attribute denote classes whose descriptions are to be learned.

In Figure 3, the first column corresponds to output attribute. In conceptual clustering, there are no a priori classes to which entities belong (therefore, it is a form of "unsupervised learning"). Figure 3 illustrates operators summarized below.

Concept learning from examples

Classes of examples are sets of rows in the table that have the same value of the output variable. The operator determines general descriptions (rulesets) characterizing the classes of examples.

Example selection

This operator selects rows in the table that correspond to the most representative examples of different classes. One of the methods for doing this is based on choosing *outstanding representatives.*

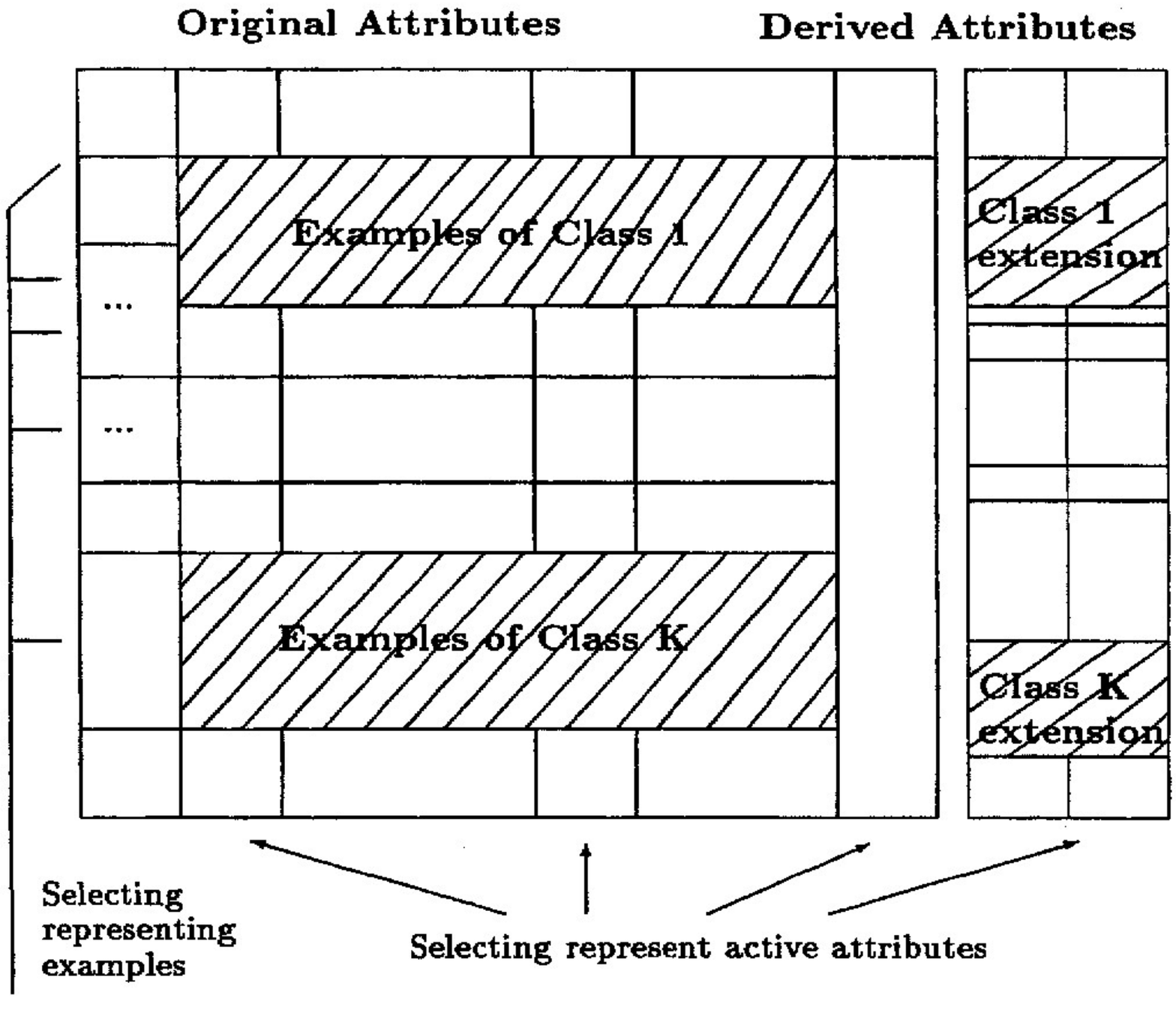

Figure 3. An illustration of the role of different knowledge generation operators

Attribute selection

This operator selects columns that correspond to the most relevant attributes for characterizing given classes or the differences among them. This operator can employ any of the many criteria for evaluating the usefulness of attributes for classification (e.g., Quinlan, 1986).

Generating new attributes

The problem is to generate additional columns that correspond to new attributes generated by a constructive induction method. These new attributes are created by using the problem background knowledge, and/or special heuristic procedures (Michalski, Bloedorn and Wnek, 1991; Wnek and Michalski, 1993; Bloedorn and Michalski, 1996).

Conceptual clustering

The problem is to split the rows of the table to groups of rows that correspond to "conceptual clusters", that is sets of entities with high conceptual cohesiveness. An additional column is added to the table that corresponds to a new "output attribute". The values of this attribute in the table denote the proposed class of each entity (Michalski, Stepp and Diday, 1991; Michalski and Stepp, 1993).

Learning from imperfect data

In some situations, the entries of the data table are missing, or are incorrect. The problem is to determine the best (e.g., the most plausible) hypothesis that accounts for all or the most of the data.

Machine learning research has developed a large number of methods that can be used as data mining operators. Many of these methods and their implementations have been described in (Michalski, Carbonell and Mitchell, 1983 and 1986; Forsyth and Rada, 1986; Kodratoff, 1988, and Kodratoff and Michalski, 1990, Michie, 1991; Shapiro, 1993). Below we illustrate some of the knowledge generation operators employed in the INLEN knowledge discovery system (see sec. 4). The operators are illustrated by means of a simple example involving microcomputers from a computer museum. Suppose we are given a data table with there microcomputers, as shown in Figure 4.

Microcomputer	Display	RAM	ROM	Processor	No_Keys
Apple II	Color_TV	48K	10K	6502	52
Atari 800	Color_TV	48K	10K	6502	57-63
Comm. VIC 20	Color_TV	32K	11-16K	6502A	64-73
Exidi Sorceror	B/W_TV	48K	4K	Z80	57-63
Zenith 118	Built_in	64K	1K	8080A	64-73
Zenith 1189	Built_in	64K	8K	Z80	64-73
HP 85	Built_in	32K	80K	HP	92
Horizon	Terminal	64K	8K	Z80	57-63
Challenger	B/W_TV	32K	10K	6502	53-56
O-S 11 Series	B/W_TV	48K	10K	6502C	53-56
TRS-80 I	B/W_TV	48K	12K	Z80	53-56
TRS-80 III	Built_in	48K	14K	Z80	64-73

Figure 4. A Data Table with Ancient Microcomputers

Suppose now that we would like to determine a conceptual classification of the microcomputers in the table. This is done by applying a conceptual clustering operator, CLUSTER, which takes data in the table in Figure 4 as one of its inputs (other inputs are value sets of the attributes, a criterion for measuring clustering quality, and a parameter suggesting the number of classes).

The results of applying this operator are shown in Figure 5, for the suggested number of classes 2 and 3. The results consist of two components: an extended data table, and a set of rules. The new data table has two additional columns: the first column indicates the "numerical name" of the class assigned to each tuple (entity) in the generated two-class clustering, and the second column indicates the numerical name of the class in three-class clustering. The second component are two sets of rules: the first rule set describes classes in the two-class clustering, and the second ruleset describes classes in the three-class clustering (Figure 6).

Rules characterizing 2-class clustering

[Class 1] $\Leftarrow$ [RAM =16K..48K]
[Class 1] $\Leftarrow$ [No Keys $\leq$ 63]
[Class 2] $\Leftarrow$ [RAM = 64K] & [No Keys $>$ 63]

INPUT						OUTPUT	
Microcomputer	Display	RAM	ROM	Processor	No_Keys	2-Group	3-Group
Apple II	Color TV	48K	lOK	6502	52	1	1
Atari 800	Color TV	48K	lOK	6502	57-63	1	1
COmIll. VIC 20	Color TV	32K	11-16K	6502A	64-73	1	2
Exidi Sorceror	B/W TV	48K	4K	Z8O	57-63	1	2
Zenith 118	Built_in	64K	1K	8080A	64-73	2	3
Zenith 1189	Built_in	64K	8K	Z80	64-73	2	3
HP 85	Built_in	32K	80K	HP	92	1	2
Horizon	Terminal	64K	8K	Z80	57-63	1	2
Challenger	B/W TV	32K	10K	6502	53-56	1	1
0-S 11 Series	B/W TV	48K	lOK	6502C	53-56	1	2
TRS-80 I	B/W TV	48K	12K	Z80	53-56	1	1
TRS-80 III	Built_in	48K	14K	Z80	64-73	1	1

Figure 5. An extended table generated by the CLUSTER operator

Rules characterizing 3-class clustering

[Class 1] $\Longleftarrow$ [Processor = 6502 v 8080A v Z80] & [ROM = l OK..14K]

[Class 2] $\Longleftarrow$ [Processor = 6502A v 6502C v HP]
[Class 2] $\Longleftarrow$ [ROM=1K..BK] & [Display $\neq$ Built in]

[Class 3] $\Longleftarrow$ [Processor = 6502 v 8080A v Z80] &
[ROM = l K.. 8K] & [Display = Built in]

Figure 6. Rules characterizing classes created by the CLUSTER operator

Suppose now that we use the extended data table in Figure 5 as an input to a program for learning concepts from examples. Suppose that the parameters of the operator, GENRULE, call for determining *discriminant* descriptions of the classes (descriptions that use the minimum conditions needed to discriminate between given classes). The results are shown in Figure 7.

Rules for a 2-class differentiation created by the GENRULE operator

[Class 1] $\Longleftarrow$ [Display $\neq$ Built in]
[Class 1] $\Longleftarrow$ [ROM $\geq$ 14K]
[Class 2] $\Longleftarrow$ [RAM = 64K] & [No_Keys = 64-73]

Rules for a 3-class differentiation created by the GENRULE operator

[Class 1] $\Longleftarrow$ [Processor = Z80 v 6502] & [ROM = l OK..14K]

[Class 2] $\Longleftarrow$ [Processor = 6502C v 6502A v HP]
[Class 2] $\Longleftarrow$ [ROM = 4K..BK] & [Display = B/W_TV v Term]

[Class 3] $\Longleftarrow$ [ROM =1K..BK] & [Display = Built_in

Figure 7. Discriminant rules for old microcomputers generated by the GENRULE operator

Comparing rules in Figure 6 with those on Figure 7 (the latter were generated without knowledge of the former), one can see that they are similar but not identical. Rules in

Figure 7 are simpler, and express only information needed for discriminating between the classes. The rules in Figure 6 are called *characteristic* descriptions; such rules may contain the maximal number of characteristics common for a given class (Michalski, 1983). Both sets of rules, in Figure 6 and 7, are *complete* and *consistent* with all the examples in the table in Figure 5, i.e., they cover all examples and do not cover any counter-examples of each class.

4. INLEN: A Multistrategy System for Knowledge Discovery

CLUSTER and RULEGEN are examples of operators that produce new knowledge from given data and background knowledge. These operators have been described in the Inferential Theory of Learning (Michalski, 1994) as fundamental *knowledge generation* operators. The theory, which views every form of learning and discovery as a search through a knowledge space, has identified several other fundamental operators, generally called *knowledge transmutations*, such as abstraction, explanation, similization, agglomeration, an others.

To make knowledge generation operators easily available to a data analyst, they have been integrated into one system, called INLEN (Kaufman, Michalski and Kerschberg, 1990; Michalski et al. 1992, Kaufman and Michalski, 1996). INLEN has integrated a wide range of knowledge generation operators, including both symbolic operators (developed in machine learning) and conventional statistical data analysis operators. To facilitate the application of these operators, INLEN combines a relational *data base* technology with a *knowledge* base technology. The database technology is used for storing and updating data tables, and the knowledge base technology is used for storing and updating rules.

A general diagram of INLEN is presented in Figure 8. (The name is an acronym from inference and learning.)

Figure 8. A general diagram of the INLEN system for conceptual data exploration

The system offers a data analyst three classes of data manipulation and knowledge generation operators:

- DMO: Database Maintenance Operators. These operators are conventional operators for creating, modifying and accessing data tables.

- KMO: Knowledge Maintenance Operators. These operators play a similar role as the DMO, but they apply to the rules and other structures in the knowledge base.
- KGO: Knowledge Generation Operators. These operators perform symbolic and numerical data mining operations on data and knowledge to produce new knowledge. They are based on machine learning and inference programs, as well as on conventional data analysis techniques.

INLEN also includes data and knowledge visualization operators for visualizing results of data mining. The diagrammatic visualization method, described briefly above, is used for displaying effects of symbolic operations on discrete data.

The KGOs operators are the heart of the INLEN system. To facilitate their use, the concept of a *knowledge segment* was introduced. A knowledge segment is a structure that links some table or tables from the database with some rules from the knowledge base. Such knowledge segments are both inputs and outputs of KGO operators. Thus, KGOs can be viewed as modules for performing inferences on knowledge segments in order to create new knowledge segments. An execution of a KGO usually requires some background knowledge (BK), and is guided by some parameters. BK specifies the facts about the application domain, provides information about legal value sets of attributes, about their types and the scale, constraints and relationships among attributes, etc. The parameters specify how to choose an output description from multiple possibilities. KGOs can usually work in either incremental or batch mode. In the incremental mode, they try to improve or refine the existing knowledge; while in the batch mode, they try to create entirely new knowledge from facts in the database, using knowledge from the knowledge base.

KGOs in INLEN can be classified into several groups, based on the type of the output they generate. Each group includes a number of specific operators.

- GENRULE operators generate various kinds of rules from given facts. They include operators that generate symbolic descriptions of data, e.g., generate rules characterizing a set of facts, discriminate between groups of facts, build decision trees, characterize a sequence of events, and determine differences between sequences. They also include operators generating equations characterizing qualitatively and quantitatively numerical data sets, and build conceptual hierarchies.

- TRANSRULE operators perform various transformations of the rules, e.g., generalize or specialize, abstract or concretize given rules.

- GENATR operators generate new attributes, or select the most representative attributes from a given set (using methods of constructive induction).

- GENEVE operators generate events, facts or examples that satisfy given rules, select the most representative events from a given set, determine an example that is similar to a given example, or predict a value of a given variable.

- ANAREL operators analyze mathematical, statistical and logical relationships existing in the data, e.g., they may determine the degree of similarity between two examples, check if there is an implicative relationship between two variables, determine statistical properties of the data.

- TEST operator tests the performance of given set of rules on a testing set of examples. The primary output from the operator is a confusion matrix, i.e., a table whose (i,j)th element shows how many examples from class i were classified by the rules for class j.

For more details about these operators the reader may consult (Michalski et al. 1992; Kaufman, Michalski and Kerschberg, 1990).

5. A Real-World Application: INLEN Searches for Demographic and Economic Patterns in a World Database

This section briefly describes application of INLEN to the problem of deriving economic and demographic patterns in different regions of the world in 1965 and 1990. The database consisted of characterizations of each country in terms of 95 attributes, such as

- population
- growth rate
- percentage of the labor force in industry
- percentage of land area devoted to agriculture
- per capita GNP
- individual life expectancy
- percentage of population over age 65 and others.

Using its constructive induction capabilities, the system is able to construct additional derived attributes that are especially relevant to a given class of tasks. For example, it may construct an attribute: "Change in the life expectancy between 1980-90."

In the experiment reported below, several operators have been applied, such as conceptual clustering, representation space optimization, empirical rule induction, rule optimization, rule testing and example matching.

Here are few examples of patterns found by INLEN:

- *SE Europe has rural, heavily agricultural societies*
- *There is low resource allocation to education in Mediterranean Europe*
- *Developed Far Eastern countries, such as Japan and Korea, have low death rate*

The system created classes of countries, "regional patterns", and defined typical characteristics for these regions. It also found exceptional countries that do not follow typical patterns for their region. Here are examples of such cases:

- *Canada resembles the Far East more than the US in some respects, such as population growth rate, allocation of GNP to medicine, agricultural labor force, infant mortality rate, death rate, percentage of the labor force in industry.*
- *Italy is influenced more by Western than Southern Europe*
- *China is very similar to (formerly) Communist countries of Southeastern Europe*
- *Island countries tend to deviate from the nearby mainland's patterns.*

Although many of the found characteristics and proposed classifications of the world regions are known, some of them are novel. The main achievement of this experiment is, however, not the knowledge created, but a demonstration of the capabilities of INLEN to discover plausible and understandable patterns in large volumes of data.

6. Summary

An enormous proliferation and growth of databases has created a demand for new type of data exploration systems that can conceptually characterize data and derive useful knowledge from them. In particular, such systems need to be able to determine logical relationships, qualitative evaluations and causal dependencies in the data, which are very important for human data interpretation and decision making. To derive such knowledge, these systems need to be able to represent and take advantage of prior knowledge about the data.

This paper described a methodology underlying the multistrategy knowledge discovery system, INLEN, designed with such objectives in mind. INLEN integrates data base, knowledge base, machine learning and discovery technologies. It incorporates a large family of

operators that perform symbolic data and knowledge manipulation, and synthesis of various kinds of knowledge. These operators include rule generation, selection of most relevant attributes, selection of the most representative data items, generation of new attributes, building conceptual hierarchies, generation of equations and others.

Many INLEN operators are based on modules that have been originally developed as stand-alone programs. More research is required to make some of them more integrated with each other, or more efficient in exploring large volumes of data. Future research includes integration in INLEN other learning and discovery operators useful for data mining and knowledge discovery.

Acknowledgments

The author thanks Ken Kaufman for helping to prepare examples illustrating the INLEN's GENRULE and CLUSTER operators. The illustration of decision rules using diagrammatic visualization was prepared by Janusz Wnek. INLEN is an extension of an earlier system, AURORA, designed by Michalski and implemented jointly by Bruce Katz and Michalski. The main contribution to the implementation of INLEN extension of AURORA was done by Ken Kaufman, who also performed experiments on the INLEN's application to determining economical and demographical patterns in the world database.

This research was conducted in the Machine Learning and Inference Laboratory at George Mason University. The Laboratory's research is supported in part by the National Science Foundation under grants IRI-9510644 and DMI-9496192, in part by the Office of Naval Research under grant N00014-91-J-1351, in part by the Advanced Research Projects Agency under grant No. N00014-91-J-1854 administered by the Office of Naval Research, in part by the Advanced Research Projects Agency under grants F49620-92-J-0549 and F49620-95-1-0462, administered by the Air Force Office of Scientific Research.

References

[1] Bergadano, F., Matwin, S., Michalski, R.S., and Zhang, J., "Learning Two-Tiered Descriptions of Flexible Concepts: The POSEIDON System," *Reports of Machine Learning and Inference Laboratory*, MLI 90-9, George Mason University, Fairfax, VA September 1990.

[2] Bloedorn, E. and Michalski, R. S., The AQ17-DCI System For Data-Driven Constructive Induction and Its Application to the Analysis of World Economics, *Proceedings of the 9th International Symposium on Methodologies for Intelligent Systems*, June 9-13, Zakopane,1996.

[3] Bongard, N., *Pattern Recognition*, Spartan Books, New York,1970 (a translation from Russian).

[4] Brachman, R., Selfridge, P., Terveen, L., Altman, B., Halper, F., Kirk, T., Lazar, A., McGuiness, D., Resnick, L., and Borgida, A., "Integrated Support for Data Archeology," *Proceedings of the AAAI-93 Workshop on Knowledge Discovery in Databases*, pp.197- 211, Washington D.C., 1993.

[5] Carbone, P., and Kerschberg, L., "Intelligent Mediation in Active Knowledge Mining: Goals and General Description," *Proceedings of the AAAI-93 Workshop on Knowledge Discovery in Databases*, pp. 241-253, Washington D.C.,1993.

[6] Chan, P., and Stolfo, S., "Toward Parallel and Distributed Learning by Meta-Learning," *Proceedings of the AAAI-93 Workshop on Knowledge Discovery in Databases*, pp. 227- 240, Washington D.C.,1993.

[7] Cheng, P.C.-H and Simon, H.A., "The Right Representation for Discovery: Finding the Conservation of Momentum," *Proceedings of the Ninth International Conference on Machine Learning*, Aberdeen Scotland, pp. 62-71,1992.
[8] Cooper, G., "A Bayesian Method for Learning Probabilistic Networks that Contain Hidden Variables," *Proceedings of the AAAI-93 Workshop on Knowledge Discovery in Databases*, pp.112-124, Washington D.C., 1993.
[9] Dzeroski, S., "Inductive Logic Programming for KDD: an Overview," *Proceedings of the AAAI-93 Workshop on Knowledge Discovery in Databases*, pp. 125-137, Washington D.C., 1993.
[10] Fayyad U. and Uthurusamy, R., The First International Conference on Knowledge Discovery and Data Mining, Montreal, Canada, August 20-21, 1995.
[11] Forsyth, R. and Rada, Roy, *Machine Learning: applications in expert systems and information retrieval*, Pitman, 1986.
[12] Imam, I.F., Michalski, R.S., and Kerschberg, L., "Discovering Attribute Dependence in Databases by Integrating Symbolic Learning and Statistical Techniques," *Proceedings of the AAAI-93 Workshop on Knowledge Discovery in Databases*, pp. 264-275, Washington D.C., 1993.
[13] Imam, I.F. and Michalski, R.S., Learning Decision Trees from Decision Rules: A Method and Initial Results from a Comparative Study, *Journal of Intelligent Information Systems*, Vol. 2, pp. 279-304, 1993.
[14] Kaufman, K.A., Michalski, R.S. and Schultz, A.C., EMERALD I: "An Integrated System of Machine Learning and Discovery Programs for Education and Research, User's Guide," *Reports of the Machine Learning and Inference Laboratory*, MLI 89-11, George Mason University, Fairfax, VA,1989.
[15] Kaufman, K.A., Michalski, R.S., and Kerschberg, L., AN ARCHITECTURE FOR Knowledge Discovery from Facts: Integrating Database, Knowledge Base and Machine Learning in INLEN, in *Machine Learning and Inference Reports*, 1991.
[16] Kaufman, K.A., Michalski, R.S., A Multistrategy Conceptual Analysis of Economic Data, *Proceedigns of the 4th International Workshop on Artificial Intelligence in Econoics and Management*, Tel-Aviv, Isreal, January 8-10, 1996.
[17] Kloesgen, W., "Some Implementation Aspects of a Discovery System," *Proceedings of the AAAI-93 Workshop on Knowledge Discovery in Databases*, pp. 212-226, Washington D.C., 1993.
[18] Kodratoff, Y., *Introduction to Machine Learning*, Pittman, 1988.
[19] Klopotek, M., Michalewicz M., Wierzchon, S., Ekstracja Wiedzy w Sieci Bayesowskiej, *Materialy Konferencji Naukowej Praktyczne Aspekty Sztucznej Inteligeacji*, Instytut Podstaw Informatyki, Polska Akademia Nauk (Proceedings of the Polish conference on the Practical Aspects of Artificial Intelligence), Zakopane 7-11, 12, 1992.
[20] Major, J., and Mangano, J., "Selecting Among Rules Induced from a Hurricane Database," *Proceedings of the AAAI-93 Workshop on Knowdedge Discovery in Databases*, pp. 28-44, Washington D.C., 1993.
[21] Michalski, R.S., "A Planar Geometrical Model for Representing Multi-Dimensional Discrete Spaces and Multiple-Valued Logic Functions," *Report No. 897*, Department of Computer Science, University of Illinois, Urbana, January 1978.
[22] Michalski, R. S., Stepp, R. and Diday, E.,"A Recent Advance in Data Analysis: Clustering Objects into Classes Characterized by Conjunctive Concepts," Invited chapter in the book *Progress in Pattern Recognition*, Vol.1, L. Kanal and A. Rosenfeld (Editors), pp. 33-55., 1981.
[23] Michalski, R.S., "A Theory and Methodology gf Inductive Learning," *Artificial Intelligence*, 1983,. pp.111-161, 1983.
[24] Michalski, R.S., "Learning Flexible Concepts: Fundamental Ideas and a Method Based on Two-tiered Representation," *Machine Learning: An Artificial Intelligence Approach, Vol. III*, Y. Kodratoff and R.S. Michalski (Eds.), Morgan Kaufmann Publishers, 1990.

[25] Michalski, "Inferential Theory of Learning: Developing Foundations for Multistrategy Learning, *Machine Learnin: A Multistrate Learning Approach Vol. 4*. Morgan Kaufmann Publishers 1994.

[26] Michalski, R.S., Kerschberg, L., Kaufman, K, and Ribeiro, J., "Mining for Knowledge in Databases: The INLEN Architecture, Initial Implementation and First Results," *Journal of Intelligent Information Systems*, pp. 85-113, Vol.1, No.1, August, 1992.

[27] Michalski, R.S. and Stepp, R., "Learning from Observation: Conceptual Clustering," *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufman, 1993.

[28] Michalski, R.S. and Kaufman, K., "Multistrategy Data Minining and Data Analysis," chapter in *Machine Learning and Data Mining*, R.S. Michalski, I. Bratko and M. Kubat (eds.), John Wiley & Sons, 1997 (in press).

[29] Michie, D., "Methodologies from Machine Learning in Data Analysis and Software," *The Computer Journal*, vol.34, Noi. 6, 1991.

[30] Pawlak, Z., *Rough Sets–Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, 1991.

[31] Piatetsky-Shapiro, G. (ed.), *Proceedings of the AAAI-91 Workshop on Knowledge Discovery in Databases*, 1991.

[32] Piatetsky-Shapiro, G. (ed.), *Proceedings of the AAAI-93 Workshop on Knowledge Discovery in Databases*, 1993.

[33] Quinlan, J.R., "Induction of Decision Trees," *Machine Learning*, Vol.1, No.1, pp. 81-106, 1986.

[34] Wnek, J., Sarma, J., Wahab, A., Michalski, R.S., "Comparing Learning Paradigms via Diagrammatic Visualization: A Case Study in Single Concept Learning using Symbolic, Neural Net and Genetic Algorithm Methods," *Reports of the Machine Learning and Inference Laboratory*, MLI 90-2, George Mason University, Fairfax, VA, January 1990.

[35] Wnek, J. and Michalski, R.S., "Hypothesis-driver Constructive Induction in AQ17-HCI: A Method and Experiments," *Machine Learning*, Special Issue on Evaluating and Changing Representations, 1993.

[36] Ziarko, W. P., Rough sets, fuzzy sets and knowledge discovery, *Proceedings of the International Workshop RSKD'94*, Banff, Alberta, Canada, October 12-15, 1993.

[37] Zembowicz, R. and Zytkow, J.M., "Discovery of Equations: Experimental Evaluation of Convergence," Proceedings of AAAI-92, San Jose, CA, pp. 70-73, 1992.

[38] Zytkow, J.M., Zhu, J. and Zembowicz, R., "The First Phase of Real-World Discove Determining Repeatability and Error of Experiments," *Proceedings of the Ninth International Conference on Machine Learning*, Aberdeen Scotland, pp. 480-485, 1992.