AN ARCHITECTURE FOR INTEGRATING MACHINE LEARNING AND DISCOVERY PROGRAMS INTO A DATA ANALYSIS SYSTEM

by

K. Kaufman
R. S. Michalski
L. Kerschberg

AAAI-91 Workshop on Knowledge Discovery in Databases, July 1991.

Presented at AAAI-91 Workshop on Knowledge Discovery in Databases, July 1991

AN ARCHITECTURE FOR INTEGRATING MACHINE LEARNING AND DISCOVERY PROGRAMS INTO A DATA ANALYSIS SYSTEM

Kenneth A. Kaufman, Ryszard S. Michalski and Larry Kerschberg

Artificial Intelligence Center
George Mason University
Fairfax VA 22030

AN ARCHITECTURE FOR INTEGRATING MACHINE LEARNING AND DISCOVERY PROGRAMS INTO A DATA ANALYSIS SYSTEM

Kenneth A. Kaufman, Ryszard S. Michalski and Larry Kerschberg

ABSTRACT

The architecture of a large-scale system, INLEN, for the discovery of knowledge from facts, is described and then illustrated by an exploratory application. INLEN combines database, knowledge base, and machine learning methods within a uniform user-oriented framework. Data and different forms of knowledge are managed in a uniform way by using *knowledge segments,* which are structures that link relational tables with rules, equations and/or hierarchies. A variety of machine learning programs are incorporated into the system to serve as high-level *knowledge generation operators* (KGOs). These operators are used for generating various kinds of knowledge about the properties and regularities existing in the data. For example, such operators can hypothesize general rules from facts, determine differences and similarities among groups of facts, propose new variables, create conceptual classifications, determine equations governing numeric variables and the conditions under which the equations apply, and many others. The system also includes capabilities for determining a variety of statistical characteristics. Operators may be combined into macros or programs for repeated applications or automatic branching. An application of INLEN for knowledge discovery in the domain of international trade is briefly described.

ACKNOWLEDGEMENTS

The authors thank Michael Hieb, Jim Ribeiro, Gheorghe Tecuci and Brad Utz for their comments and criticism of this paper and its earlier versions.

This research was done in the Artificial Intelligence Center of George Mason University. The activities of the Center are supported in part by the Defense Advanced Research Projects Agency under grant, administered by the Office of Naval Research No. N00014-87-K-0874, in part by the Office of Naval Research under grant No. N00014-88-K-0226, and in part by the Office of Naval Research under grant No. N00014-88-K-0397.

AN ARCHITECTURE FOR INTEGRATING MACHINE LEARNING AND DISCOVERY PROGRAMS INTO A DATA ANALYSIS SYSTEM

1. Introduction

In response to the rapid expansion and widespread use of database technology, there is a growing interest in developing new techniques for extracting knowledge from data. In particular, there is interest in developing methods that will go beyond the traditional statistical analyses, and produce symbolic rather than numerical data descriptions. Such methods should be able to discover "conceptual" patterns in the data, suggest explanations for them, and generate plausible predictions.

This paper presents the ongoing research on the development of a system, INLEN, that employs a variety of machine learning and discovery techniques to extract useful knowledge from databases. The paper discusses ideas on how to integrate these techniques into a large-scale data analysis system, and presents an updated architecture of the system. The machine learning and discovery techniques are implemented in the form of *knowledge generation operators* (KGOs). A knowledge generation operator, given a *knowledge segment* (a combination of data from a database and knowledge from a knowledge base) derives a new knowledge segment. Such a derivation is accomplished by various forms of inference, such as deduction, induction or analogy. In the INLEN design, the KGOs include operators for creating conceptual descriptions of sets of facts, identifying logical regularities and similarities among facts or groups of facts, inventing conceptual classifications of data, generating new attributes to better describe data, selecting relevant examples or attributes, formulating equations governing quantitative data as well as the conditions of their applicability, etc. These operators can be invoked by a user at any step of data analysis.

The development of such a system is a very complex task. It consists of building a software environment for the incorporation and support of the functions of individual modules, and the development of these modules. Each module by itself is a complex program, which may represent an advanced machine inference or learning capability or a group of related capabilities. Consequently, the implementation of INLEN has to proceed incrementally.

This paper presents recent results in the INLEN development. In particular, they include improvements and extensions of the original architecture [Kaufman, Michalski and Kerschberg,

1989], and the additions and specifications of new knowledge generation operators. To illustrate the practical aspects of this research, the paper also describes an experimental application of INLEN to the domain of international trade.

2. The Conceptual Structure of INLEN

The motivating idea behind the INLEN system is to provide a user with an integrated set of tools for manipulating both data and knowledge, and for extracting new or better knowledge from that data and knowledge. To this end, INLEN integrates database, knowledge base and machine learning technologies. By integrating such components, INLEN can perform a wide range of functions. It thus can be viewed as an "intelligent data analysis assistant," that performs sophisticated data analysis operations, either under the directed guidance of a user, or partially autonomously. Specifically, such an assistant can conduct "conceptual" data analysis, search for unknown regularities, and propose explanations for the patterns discovered. To reflect the needs of the user, the system is equipped with criteria that characterize the classes of patterns that are important to a user.

Given a set of data, INLEN searches for all kinds of qualitative and/or quantitative patterns, and notifies a data analyst about the patterns discovered that it views as possibly important. These patterns are determined by various methods of symbolic concept learning and reasoning. Experiments with some existing machine learning programs have shown that these programs can find unexpectedly simple patterns that are difficult to find by people [e.g., Michalski, 1983], or discover regularities that would be hard to formulate without the aid of a program [e.g., Falkenhainer and Michalski, 1990]. When a database is large, it is difficult for an analyst to find patterns just because of the sheer volume of data. An intelligent database assistant can also help to avoid possible human errors of overlooking something of note in the data. Because the system has the ability to process data faster than a human analyst, and with a low error rate, it can be especially useful in domains requiring an analysis of large databases.

The approach that we are employing is to build a synergistic system that allows a human expert and a computer tool to perform the tasks that each party is better suited for. Data and knowledge management functions, searches through large data sets, consistency checking and discovery of certain classes of patterns are relatively easy to perform by a learning and discovery system. On the other hand, defining criteria for judging what is important and what is not, making decisions

about what data to process, and evaluating findings from the viewpoint of human utility are easier for a human expert. Working together, such a human-computer data analysis system can exhibit a synergistic effect in extracting useful knowledge from data, and have an increased potential for making discoveries. A machine learning system may also be potentially useful in formulating explicit criteria that experts are using implicitly in evaluating the "interestingness" of a pattern. Systems able to assist in the extraction of useful knowledge from large databases can be useful in many fields, such as complex decision making, resource allocation and management, business transactions, medicine, chemistry, physics, economics, demographics, scheduling, planning, etc.

INLEN's basic assumption is that the data is stored in a relational database system, and that some locations in a table may be marked "?" to indicate unknown values, or marked "N/A" denoting the non-applicability of some attributes to some objects. It is also assumed that encoded in the system is rudimentary background knowledge about the variables that characterize the data, the values they can have, and the relationships among those values (linear, hierarchical, etc.), and possibly among the variables. Given such a table, the INLEN operators will try to discover useful facts about the objects represented in the table.

The underlying knowledge representation in INLEN is *knowledge segments*, which link these relational tables with rules, equations and/or hierarchies. Since different discovery algorithms will discover different types of knowledge associated with different domain objects, the knowledge segment has to be a flexible structure for storing both background and discovered knowledge. Its format is designed to provide an object-oriented structure that can facilitate the interaction between relevant new knowledge and both the data and the previously acquired knowledge, so as to allow for an active data/knowledge base. In addition, this format should facilitate the user's understanding of the concepts stored within the data and knowledge bases.

Not only must INLEN's knowledge structure be capable of incorporating different types of knowledge into the knowledge base, it must also be able to cope with the different relationships between the knowledge and the data. For instance, a discovered fact may correspond to an integrity constraint in the database, or it may point out a simple regularity for which exceptions found in incoming data should be pointed out to the domain expert or should trigger an incremental learning process. The expertise of the data and knowledge administrators can be critical in classifying discovered knowledge and properly integrating it into the data and knowledge schemas.

3. Knowledge Generation Operators

INLEN's KGOs can be conceptually divided into ten high-level groups. In the process of design and development, we have incorporated several novel knowledge generation operators and also the idea of macrooperators and data-analysis programs into the INLEN environment. The macrooperators differ from the other operators in that they consist of sequences of the other operators, along with instructions to control the flow of commands within the macro. For more complex operations, programs can be written to direct discovery in INLEN.

In general, a knowledge generation operator takes at least one table from the database and at least one knowledge segment from the knowledge base, and generates one or more new tables and/or knowledge segments. A description of the KGO classes and the individual operators follows:

GENRULE: Generate Rules

Operators in the GENRULE class take some form of data and/or knowledge as an input, and return a ruleset consisting of facts induced from the input examples. There are four different operators in the GENRULE class.

CHARSET (Characterize Set) determines a description characterizing a class of entities by discovering characteristic rules. DIFFSET (Differentiate Set) takes one set of objects (each object represented as a tuple in a relational table that may represent data or metaknowledge) as a primary input, and one or more sets of objects as a controlling input, and induces general rules that encapsulate the differences between the primary set and the other sets. Existing programs that perform such tasks include UNICLASS [Stepp, 1979], AQ11 [Michalski and Larson, 1983], and AQ15 [Hong, Mozetic and Michalski, 1986]. DIFFSET is demonstrated in further detail (under the name DIFF) by Kaufman, Michalski and Kerschberg [1989].

CHARSEQ (Characterize Sequence) determines descriptions characterizing a sequence of objects or events. This is a more complex operator than CHARSET, since the learner must now take into account the influences of ordering and positioning of examples in the sequence, and it may also have to consider negative examples -- objects that do not belong at a given point in the sequence.

DIFFSEQ (Differentiate Sequence) discovers differences between two or more sequences of objects or events. These operators represent an extension of the SPARC methodology for determining patterns in sequences, described by Michalski, Ko and Chen [1985, 1986] and Dieterrich and Michalski [1986].

GENTREE: Generate Decision Trees

The two GENTREE operators output knowledge in the form of decision trees. EVENTREE (Event to Tree) uses events in a relational table as input, and generates a tree for classifying the input examples, while RULETREE (Rule to Tree) organizes a set of decision rules into a tree. EVENTREE is based on ideas implemented in the C4.5 program for generating decision trees from examples [Quinlan, 1990] and in ASSISTANT [Cestnik, Kononenko and Bratko, 1987]. The RULETREE operator utilizes the OPTTREE program for creating decision trees from rules [Michalski, 1978; Layman, 1979].

GENEQ: Generate Equations

GENEQ is a single operator that discovers equations that describe numeric data in a set of examples, and formulates conditions for applying these equations. GENEQ is based on the ABACUS-2 system for integrated qualitative and quantitative discovery [Greene, 1988], an extension of ABACUS [Falkenhainer and Michalski, 1990]. These quantitative discovery programs are related to systems such as BACON [Langley, Bradshaw and Simon, 1983], FAHRENHEIT [Zytkow, 1987] and COPER [Kokar, 1986]. [Kaufman, Michalski and Kerschberg, 1989] gives a detailed example of an application of the GENEQ operator.

GENHIER: Generate Hierarchies

The GENHIER operators conceptually classify an input set of tuples, rules, equations, etc. The CLUSTER operator creates a logical division of the input objects into two or more groups (a hierarchy one level deep), while the TAXONOMY operator generates a full-fledged classification hierarchy, and can be viewed as a recursive invocation of CLUSTER. In addition to the generated hierarchies, both operators determine a set of rules characterizing the created groups. CLUSTER and TAXONOMY use the conceptual clustering algorithm from the CLUSTER program described in [Michalski, Stepp and Diday, 1981; Stepp, 1983, 1984]. A detailed example of the application CLUSTER operator in INLEN is given in [Kaufman, Michalski and Kerschberg, 1989].

TRANSKS: Transform Knowledge Segments

The TRANSKS operators perform basic inferential transformations on knowledge segments, hence both the primary inputs and outputs are knowledge segments of the same type, typically decision rules. There are two pairs of inverse operators: ABSTRACT and CONCRETIZE, and GENERALIZE and SPECIALIZE, in addition to IMPROVE, an operator that improves knowledge by giving it new examples to learn from. ABSTRACT modifies its input knowledge segment by removing details from its description, and CONCRETIZE specifies details of an abstract concept description, while GENERALIZE and SPECIALIZE affect the set size covered by the input knowledge segment. Examples of these four operators are shown by Kaufman, Michalski and Kerschberg [1990]. IMPROVE detects exceptions to existing knowledge in its input examples, and refines the knowledge segments accordingly.

The methods used by the ABSTRACT and CONCRETIZE operators have not been heavily studied in the past. [Michalski, 1990] discusses the place of abstraction and concretion within multistrategy learning environments. A topic of current research is the development of intelligent control strategies for the GENERALIZE and SPECIALIZE operators. The methodology for IMPROVE is based on the AQ15 program [Hong, Mozetic, Michalski, 1986].

GENATR: Generate Attributes

The GENATR operators map relational tables to relational tables whose rows are the same but whose columns have been changed, either by the addition of new attributes or by the removal of old ones. SELATR (Select Attribute) determines the attributes in a relational table that are most relevant for differentiating between various classes of objects, and produces a reduced table that retains only those variables chosen by the operator. It is based on the VARSEL algorithm [Baim, 1982]. CONATR (Construct Attribute) applies mathematical operators specified in its arguments in order to combine variables into useful composites, and is based on the CONVART program [Davis, 1979].

GENEVE: Generate Events

The GENEVE class covers a wide variety of operators that generate a new set of tuples, either from an existing relational table, or from the entire event space to which a table's tuples belong.

SELEVE (Select Event) determines the examples that are the most representative of the examples contained in input relational tables. The output from this operator is a subtable of the input table, consisting of the most promising examples from the input table. SELEVE uses the ESEL methodology, described by Michalski and Larson [1978] and Cramm [1983]. CONEVE (Construct Event) searches the example set or event space for elements satisfying some selection criteria. PREDVAL (Predict Value) speculates on likely values for unknown attributes of incomplete or hypothetical data elements. CONEVE and PREDVAL are the subject of research to build upon the foundations set by the performance elements of the SPARC and APPLAUSE [Dontas, 1988] programs. SIMILIZE (Find Similar) seeks out events or relationships that are similar to the input in some defined sense. SIMILIZE is being developed from applications of object characterization programs such as AQ.

ANALYZE: Analyze Data

The ANALYZE family of operators return knowledge in the form of numerical weights that describe the elements in the database. These numbers can represent logical or statistical relationships. RELATR (Relate Attributes) determines a relationship, such as equivalence, implication, correlation or monotonic dependency that may exist between two or more attributes in a relational table, and RELEVE (Relate Events) similarly determines relationships among elements in a relational table. RELKS (Relate Knowledge Segments) discovers relationships such as inclusion, disjointedness, correlation, generalization and abstraction within a set of knowledge segments. All three of these operators are subjects of current research. GENSTAT (Generate Statistics) uses existing programs to perform a statistical analysis of the data in order to determine its various statistical properties.

TEST: Test Knowledge

The TEST operator determines the performance of a ruleset on a set of examples by testing the input knowledge segments for consistency and completeness with regard to the input examples. Consistency implies that no event in the example set is covered by two different rules. It uses the ATEST methodology [Reinke, 1984].

VISUALIZE: Diagrammatic Visualization

VISUALIZE displays a set of data, as specified by rules or transformations of it, graphically on the screen. The output from this operator appears as a two-dimensional representation of the event space, with the input set highlighted. VISUALIZE uses the DIAV diagrammatic visualization methodology, currently being developed [Wnek et al, 1990; Wnek and Michalski, 1990].

4. Macrooperators and Data Analysis Programs

In order to facilitate the repeated execution of certain operator sequences by users, INLEN is designed to provide mechanisms for creating macrooperators and high-level data analysis programs. Macrooperators allow for repeatable, standard sequences of operations. They encompass a small number of INLEN operators, and can be added to a KGO menu and called upon as single operators. Among the research and design issues to be addressed during INLEN's prototype development and testing will be which sequences of operators are most useful for inclusion as standard macros in the INLEN package, and what is an effective scheme in which users can develop, write and store their own macrooperators.

It may also be the case that there is a repeatable application that must call upon a longer sequence of operators, possibly making simple control decisions based on the output of earlier operators in the sequence. INLEN allows the user to read a data-analysis program from a file in order to perform such tasks. Because such programs of operators can make their own control decisions, they allow for long, unsupervised sessions. The language for these programs includes the capacity for branching, looping, and local variables. For example, a program may be called to invoke a data management operator for adding new records to a database from a file, until all records in a waiting area were cleared out. It can then call TEST to see if the new records are consistent with the relevant knowledge stored in the knowledge base. In the case of inconsistency, it can then call the DIFFSET operator to modify the inconsistent knowledge. A more complex example is shown along with accompanying pseudocode in [Kaufman, Michalski and Kerschberg, 1990].

5. The Implementation of INLEN

INLEN is a very complex large-scale system composed of many intricate modules, some of which can serve as powerful stand-alone systems. The design and implementation of the system builds

upon the development of the QUIN system [Michalski, Baskin and Spackman, 1982; Michalski and Baskin, 1983; Spackman, 1983]. Many of the INLEN operators are based on the research results and programs developed over the last 15 years at this and other laboratories. Incorporating these programs into INLEN requires different amounts of effort. In a few cases, this includes primarily a change of the program interface. In other cases, the programs have to undergo major modifications or be redeveloped from scratch. Finally, some other operators are still at the stage of research and initial implementation. Therefore, INLEN implementation is regarded as a multi-stage task.

The first stage of development (version INLEN-0) includes a knowledge base of simple decision rules, a relational database, and an extensive user-oriented menu-based graphical interface. The knowledge generation operators (KGOs) include preliminary versions of such operators as: CHARSET, DIFFSET, IMPROVE, TEST and PREDVAL. The system has been implemented on an AT-compatible computer.

The second stage includes the development of a larger prototype on a Sun platform that integrates a full-fledged knowledge base with a commercial-grade database. The system includes most of the knowledge generation operators indicated in Section 3, and a new system interface.

The third stage will involve the development and implementation of the remaining operators, and modifications to the structure of the system's components based on the results generated during the previous stage.

6. Experimental Application: Discovering Patterns in the U.S. Foreign Trade Data

One experimental application domain which we are exploring involves a database of United States import and export trade amounts, maintained by the US government. The goal of the research is to detect "unusual" patterns and anomalies in the data.

Entries in the data table are sorted by destination, by product, and by month. The products are organized into a hierarchy of about 150 groups. The amounts for all levels in the product class hierarchy are included in the data. Also available are the rankings of the different product groups based on the different criteria. Given the large tables of values, stored in a relational database

structure, we would like to discover facts in the data that will interest a foreign affairs analyst. Early knowledge of such a fact may provide an advantage to the party discovering it.

For example, in response to a high United States tariff on imported trucks, there was *a sharp increase in imports from Japan in the "auto parts" category, and a decrease in the trucks category* in the early 1980s. Japan had begun avoiding the tariff by exporting chassis and truck beds separately, and having the trucks assembled in the US. When US analysts discovered this fact, conditions relating to the import of trucks and truck parts were included in trade talks with Japan.

An example of the trade data, showing United States exports to Japan in the latter half of 1988 is shown in Table 1. By inspecting such data, one can also see trends and aberrations, for example a noticeable across-the-board decrease in significant Japanese imports of non-miscellaneous foodstuffs between November and December, 1988. There are also short-term anomalies, such as during December, 1988, when Japan imported an abnormally large amount of refined petroleum. These are expressed as rules in the INLEN knowledge representation language:

Japan_Import_Nov_88 > Japan_Import_Dec_88 if
    [product is_a Foodstuffs] & [product is_not_a Other Foods] &
    [Japan_Import_Nov_88 > 1000]

Import_level is abnormally_high if
    [month is December] & [year is 88] & [country is Japan] & [product is Refined Petroleum]

Since the purpose of a discovery system in this domain is to detect trends and report "interesting" ones to a domain expert, the system must go beyond simple discovery, and be able to anticipate what will interest the expert. A discovery system can detect and report on "interesting" features. If an expert informs a machine learning system of the interestingness or lack thereof of these discoveries, the system can acquire knowledge of what is important and interesting by use of an example-based discovery system. With greater efficiency, the discovery module can key in upon interesting trends in the data, and pass them along to the expert. Such notifications could lead to a strategic advantage, or the neutralization of a disadvantage, as was indicated by the truck import example given above.

7. Conclusion

Most research on the discovery of knowledge in databases is concerned with some specific type of discovery or knowledge extraction. The main aim of this paper is to discuss design ideas and an architecture of a system that integrates many different machine learning and discovery programs. Specifically, the paper presents the recent progress on the development of INLEN, a large-scale system capable of performing a wide variety of complex inferential operations on data in order to discover interesting regularities in them. These regularities can be detected in qualitative data, quantitative data, and in the knowledge base itself. In addition, INLEN provides functions that facilitate the manipulation of both the data and the knowledge base.

Since the aim or the paper was to discuss the whole system, many details have been omitted. Details about the individual machine learning and discovery programs (corresponding to various operators) that are incorporated in the system are described in various papers referred to in the text.

Commodity	*Jul 88*	*Aug 88*	*Sept 88*	*Oct 88*	*Nov 88*	*Dec 88*	*Total 88*
Total Trade	2250908	2273982	2260283	2228584	2164687	2357652	26191388
FOODSTUFFS	684373	646947	520971	539072	552766	476000	6205520
Dairy	2758	2140	2331	1513	2983	2152	25841
Meat and Fish	380434	289821	200805	176525	156386	110899	2115725
Meat, live animals	77674	97870	103502	100739	127716	82038	1002670
Fish	302760	191951	97303	75796	38670	28861	1113055
Grains	186253	271374	217134	246901	255289	228533	2574579
Wheat	28330	32540	15423	24813	30783	26729	298177
Rice	10	6	18	23	43	81	277
Animal Feed	19068	24646	25011	20071	28948	16610	231241
Other cereal	7252	5542	30109	15504	21527	19751	175160
Fruit and Vegetables	49656	44948	39767	44926	45620	36842	587260
Fruit	41418	35956	28040	31450	30505	25096	447626
Vegetables	8238	8992	11727	13470	15115	11846	139634
Coffee, Cocoa	3431	3373	5540	5702	5201	5159	52203
Coffee	1049	814	530	793	794	1204	11758
Cocoa, Chocolate	2382	2569	5010	4909	4407	3955	40445
Other Food	55307	47690	46771	58065	83002	88163	782533
Misc. Food Preps.	8161	7256	8354	9340	9601	9802	97927
Beverages	12340	8796	7131	4247	4692	3703	108170
Tobacco	34806	31638	31286	45478	68709	74658	576436
FUELS	73159	68216	72587	73569	62695	102717	996972
Refined petroleum	26821	21583	22412	23279	23126	70995	461765
Gas, electricity	10166	14143	9731	8933	8639	8766	112140

Table 1. An example of US-Japan trade data

This research aims at developing a laboratory for studying the extraction of knowledge from large databases, and at developing a useful tool that can be applied to various practical discovery problems. INLEN is intended to demonstrate a system that is capable of examining large quantities of data, detecting trends, correlations and anomalies in the data, analyzing the importance of these discoveries, reporting significant patterns, and predicting missing or future data elements. The ability to process and analyze increasing volumes of data can become a tremendous asset to anyone faced with more information than they can absorb. Using AI and machine learning techniques, the search through the data can be made in far less time, and with a greater signal-to-noise ratio.

INLEN implements a number of novel ideas. It integrates a variety of knowledge generation operators that permit a user to search for various kinds of relationships and regularities in the data. This integration allows it to exploit the strengths of diverse learning and discovery programs, to reduce its limitation to specific tasks, and to attain the capability for multistrategy learning and discovery. Depending on the situation at hand, operators may be called upon to perform empirical induction, constructive induction or deduction, abductive hypothesizing, analogical reasoning, or deductive inference.

To achieve this integration, the concept of a *knowledge segment* has been introduced. The knowledge segment stands for a variety of knowledge representations such as rules, networks, equations, etc., each possibly associated with a relational table in the database (as in the case of a set of constraints), or for any combination of such basic knowledge segments. INLEN also utilizes macrooperators and data analysis programs to facilitate operation of the system and to allow more flow control to be handled by INLEN itself. Users can easily develop and invoke both of these tool sets.

The first stage of INLEN's implementation has already been completed, expanding upon the foundations of the QUIN, ADVISE [Baskin and Michalski, 1989] and AURORA [INIS, 1988] systems. In addition, many of the modules to be incorporated in INLEN have been implemented as stand-alone systems, or as parts of larger units. Other tools and the general integrated interface are under development. The domain of international trade is a promising application for the testing of some of these tools. Future work will involve bringing these systems together, applying them to various larger databases, and completing the control system to facilitate access to them in the form of simple, uniform commands.

References

Baskin, A.B. and Michalski, R.S., "An Integrated Approach to the Construction of Knowledge-Based Systems: Experiences with ADVISE and Related Programs," in *Topics in Expert System Design*, G. Guida and C. Tasso (Eds.), Elsevier Science Publishers B. V., Amsterdam, 1989, pp. 111-143.

Baim, P.W., "The PROMISE Method for Selecting Most Relevant Attributes for Inductive Learning Systems," Report No. UIUCDCS-F-82-898, Department of Computer Science, University of Illinois, Urbana IL, Sept. 1982.

Cestnik, B. Kokonenko, I. and Bratko, I., "ASSISTANT 86: A Knowledge Elicitation Tool for Sophisticated Users," *Proceedings of the Second European Working Session on Learning*, I. Bratko and N. Lavrac (Eds.), Bled, Yugoslavia, May 1987.

Collins, A. and Michalski, R.S., The Logic of Plausible Reasoning: A Core Theory," *Cognitive Science*, vol. 13, no. 1, pp. 1-49, 1989.

Cramm, S.A., ESEL/2: "A Program for Selecting the Most Representative Training Events for Inductive Learning," Report No. UIUCDCS-F-83-901, Department of Computer Science, University of Illinois, Urbana IL, Jan. 1983.

Davis, J. H., "CONVART: A Program for Constructive Induction on Time Dependent Data," Master's Thesis, Department of Computer Science, University of Illinois, Urbana IL, 1981.

Dietterich, T. and Michalski, R.S., "Learning to Predict Sequences," Chapter in *Machine Learning: An Artificial Intelligence Approach Vol. II*, R. S. Michalski, J. Carbonell and T. Mitchell (Eds.), Morgan Kaufmann Publishers, Los Altos, CA, pp. 63-106, 1986.

Dontas, K., "Applause: An Implementation of the Collins-Michalski Theory of Plausible Reasoning," Master's Thesis, University of Tennessee, Knoxville TN, August 1988.

Falkenhainer, B. and Michalski, R.S., "Integrating Quantitative and Qualitative Discovery in the ABACUS System," in *Machine Learning: An Artificial Intelligence Approach, Volume III*, Kodratoff and Michalski, Eds., Morgan Kaufmann Publishers, San Mateo CA, 1990.

Greene, G., "Quantitative Discovery: Using Dependencies to Discover Non-Linear Terms," Master's Thesis, Department of Computer Science, University of Illinois, Urbana IL, 1988.

Hong, J., Mozetic, I. and Michalski, R.S., "AQ15: Incremental Learning of Attribute-Based Descriptions from Examples, the Method and User's Guide," Report No. UIUCDCS-F-86-949, Department of Computer Science, University of Illinois, Urbana IL, May 1986.

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

Kaufman, K., Michalski, R.S. and Kerschberg, L., "Mining For Knowledge in Data: Goals and General Description of the INLEN System," IJCAI-89 Workshop on Knowledge Discovery in Databases, Detroit MI, August 1989.

Kaufman, K., Michalski, R.S. and Kerschberg, L., "An Architecture for Knowledge Discovery from Facts: Integrating Database, Knowledge Base and Machine Learning in INLEN," submitted to *Reports of Machine Learning and Inference Laboratory*, 1990.

Kerschberg, L. (ed.), Expert Database Systems: Proceedings from the First International Workshop, Benjamin/Cummings Publishing Company, Menlo Park, CA, 1986.

Kerschberg, L. (ed.), Expert Database Systems: Proceedings from the First International Conference, Benjamin/Cummings Publishing Company, Menlo Park, CA, 1987.

Kerschberg, L. (ed.), Expert Database Systems: Proceedings from the Second International Conference, George Mason University, Fairfax, VA, 1988. (to appear in book form, Benjamin/Cummings Publishing Company, Menlo Park, CA, 1988.

Kokar, M.M., "Coper: A Methodology for Learning Invariant Functional Descriptions," in *Machine Learning: A Guide to Current Research*, Michalski, Mitchell, Carbonell Eds., Kluwer Academic Publishers, 1986.

Langley, P., Bradshaw G.L. and Simon, H.A., "Rediscovering Chemistry with the BACON System," in *Machine Learning: An Artificial Intelligence Approach*, R.S. Michalski, J.G. Carbonell and T.M. Mitchell, (Eds.), Morgan Kaufmann, 1983.

Layman, T.C., "A PASCAL Program to Convert Extended Entry Decision Tables into Optimal Decision Trees," Department of Computer Science, Internal Report, University of Illinois, Urbana IL, 1979.

Michalski, R.S., "Designing Extended Entry Decision Tables and Optimal Decision Trees Using Decision Diagrams," Report No. UIUCDCS-R-78-898, Department of Computer Science, University of Illinois, Urbana IL, March 1978.

Michalski, R.S., "Theory and Methodology of Inductive Learning," in *Machine Learning: An Artificial Intelligence Approach*, R.S. Michalski, J.G. Carbonell and T.M. Mitchell, (Eds.), Morgan Kaufmann, 1983.

Michalski, R.S., "Toward a Unified Theory of Learning: Multistrategy Task-adaptive Learning," MLI Report No. 90-1, Artificial Intelligence Center, George Mason University, 1990.

Michalski R.S. and Baskin, A.B., "Integrating Multiple Knowledge Representations and Learning Capabilities in an Expert System: The ADVISE System," Proceedings of the 8th IJCAI, Karlsruhe, West Germany, August 8-12, 1983, pp. 256-258.

Michalski, R.S., Baskin, A.B. and Spackman, K.A., "A Logic-based Approach to Conceptual Database Analysis," Sixth Annual Symposium on Computer Applications in Medical Care (SCAMC-6), George Washington University Medical Center, Washington, DC, November 1-2, 1982, pp. 792-796.

Michalski, R.S., Baskin, A.B., Uhrik, C. and Channic, T., "The ADVISE.1 Meta-Expert System: The General Design and a Technical Description", Report No. UIUCDCS-F-87-962, Department of Computer Science, University of Illinois, Urbana IL, Jan. 1987.

Michalski, R.S., Iwanska, L., Chen, K., Ko, H. and Haddawy, P., "Machine Learning and Inference: An Overview of Programs and Examples of their Performance," Artificial Intelligence Laboratory, Department of Computer Science, University of Illinois, Urbana IL, Sept. 1986.

Michalski, R.S., Ko, H. and Chen, K., "SPARC/E(V.2), An Eleusis Rule Generator and Game Player," ISG 85-11, UIUCDCS-F-85-941, Department of Computer Science, University of Illinois, Urbana, IL, February 1985.

Michalski, R.S., Ko, H. and Chen, K., "Qualitative Process Prediction: A Method and Program SPARC/G," *Expert Systems*, C. Guetler, (Ed.), Academic Press Inc., London, 1986.

Michalski R.S., and Larson, J.B., "Selection of Most Representative Training Examples and Incremental Generation of VL_1 Hypotheses: the underlying methodology and the description of programs ESEL and AQ11," Report No. 867, Department of Computer Science, University of Illinois, Urbana, May 1978.

Michalski, R.S. and Larson, J.B., rev. by Chen, K., "Incremental Generation of VL1 Hypotheses: The Underlying Methodology and the Description of the Program AQ11," Report No. UIUCDCS-F-83-905, Department of Computer Science, University of Illinois, Urbana IL, Jan. 1983.

Michalski, R.S., Mozetic, I., Hong, J. and Lavrac, N., "The AQ15 Inductive Learning System: An Overview and Experiments," Report No. UIUCDCS-R-86-1260, Department of Computer Science, University of Illinois, Urbana IL, July 1986.

Michalski, R.S., Stepp, R.E. and Diday, E., "A Recent Advance in Data Analysis: Clustering Objects into Classes Characterized by Conjunctive Concepts," in *Progress in Pattern Recognition*, Vol. 1, L. N. Kanall and A. Rosenfeld (Eds.), New York: North-Holland, pp. 33-56, 1981.

Michalski R.S., and Stepp, R.E., "Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983.

Quinlan, J.R., "Discovering Rules by Induction from Large Collections of Examples," in *Expert Systems in the Micro Electronic Age*, D. Michie (Ed.), Edinburgh University Press, 1979.

Quinlan, J.R., "Learning Efficient Classification Procedures and Their Application to Chess Endgames," in *Machine Learning: An Artificial Intelligence Approach*, R.S. Michalski, J.G. Carbonell and T.M. Mitchell, (Eds.), Morgan Kaufmann, 1983.

Quinlan, J.R., "Probabilistic Decision Trees," in *Machine Learning: An Artificial Intelligence Approach, Volume III*, Y. Kodratoff and R.S. Michalski, (Eds.), Morgan Kaufmann, 1990.

Reinke, R.E., "Knowledge Acquisition and Refinement Tools for the ADVISE Meta-Expert System," Master's Thesis, Department of Computer Science, University of Illinois, Urbana IL, July 1984.

Spackman, K.A., "QUIN: Integration of Inferential Operators within a Relational Database," ISG 83-13, UIUCDCS-F-83-917, M. S. Thesis, Department of Computer Science, University of Illinois, Urbana, 1983.

Stepp, R.E., "Learning without Negative Examples via Variable-Valued Logic Characterizations: The Uniclass Inductive Program AQ7UN1," Report No. 982, Department of Computer Science, University of Illinois, Urbana IL, July 1979.

Stepp, R.E., "A Description and User's Guide for CLUSTER/2, a Program for Conceptual Clustering," Department of Computer Science, University of Illinois, Urbana IL, Nov. 1983.

Stepp, R.E., "Conjunctive Conceptual Clustering: A Methodology and Experimentation," PhD Thesis, Department of Computer Science, University of Illinois, Urbana IL, 1984.

Wiederhold, G., Walker, M.G., Blum, R.L. and Downs, S., "Acquisition of Knowledge from Data", International Symposium on Methodologies for Intelligent Systems, Knoxville TN, Oct. 1986.

Wnek, J. and Michalski, R.S., "A System for Visualization of Learning and Inference Processes," in preparation.

Wnek, J., Sarma, J., Wahab, A. and Michalski, R.S., "Comparing Learning Paradigms via Diagrammatic Visualization," *Proceedings of International Symposium on Methodologies of Intelligent Systems ISMIS-90*, Ras, Z., Zemankova, M., Emrich, M. (Eds.), Elsevier Press, 1990.

Zytkow, J.M., "Combining Many Searches in the FAHRENHEIT Discovery System," *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA, Morgan Kaufmann, pp. 281-287, 1987.