

# A Method for Reasoning with Structured and Continuous Attributes in the INLEN-2 Multistrategy Knowledge Discovery System

Kenneth A. Kaufman and Ryszard S. Michalski\*

Machine Learning and Inference Laboratory,  
George Mason University,  
Fairfax, Virginia, 22030, USA  
{kaufman, michalsk}@aic.gmu.edu

\* Also GMU Departments of Computer Science and Systems Engineering  
and the Institute of Computer Science, Polish Academy of Sciences

## Abstract

Structured attributes have domains (value sets) that are partially ordered sets, typically hierarchies. Such attributes allow knowledge discovery programs to incorporate background knowledge about hierarchical relationships among attribute values. Inductive generalization rules for structured attributes have been developed that take into consideration the type of nodes in the domain hierarchy (anchor or non-anchor) and the type of decision rules to be generated (characteristic, discriminant or minimum complexity). These generalization rules enhance the ability of knowledge discovery system INLEN-2 to exploit the semantic content of the domain knowledge in the process of generating hypotheses. If the dependent attribute (e.g., a decision attribute) is structured, the system generates a system of hierarchically organized rules representing relationships between the values of this attribute and independent attributes. Such a situation often occurs in practice when the decision to be assigned to a situation can be at different levels of abstraction (e.g., this is a liver disease, or this is a liver cancer). Continuous attributes (e.g., physical measurements) are quantized into a hierarchy of values (ranges of values arranged into different levels). These methods are illustrated by an example concerning the discovery of patterns in world economics and demographics.

## Introduction

Most symbolic learning systems represent information about objects or events in the form of attribute-value vectors. Attributes used in these systems are typically numerical (i.e., have totally ordered value sets) or nominal (i.e., have unordered value sets). In some applications it is useful to use attributes with value sets that are partially ordered, for example, representing a generalization hierarchy of concepts. Theoretically, almost any attribute can be viewed as having a hierarchical domain. For example, the domain of a numeric variable “age” can be split into a hierarchically organized set of classes—specific numerical values at the lowest level; toddler, child, teen, young adult, middle-age, senior and very senior at the second level; and possibly young, adult and mature at the third level. To represent such concepts, a system needs background

knowledge that relates the numerical age with the higher level concepts. In general, the structure of the domain does not have to be fixed; it may be changing with the context of the problem at hand.

Structuring attributes can prove advantageous for a knowledge discovery system. It allows facts, trends and regularities to be revealed both at high and low levels of abstraction, and for background knowledge to be stored and generalizations to be made at the appropriate levels.

The idea of grouping values of attributes into a structure of classes in order to reflect semantic relationships characteristic to the given application domain has led to the introduction of *structured* attributes (Michalski 1980). The domain of a structured attribute is a partially ordered set, typically a hierarchy. Domains of structured variables can be generated by a domain expert (e.g., Karni & Loksh 1996), or by an automatic process using a numerical or conceptual clustering method (e.g., Sokal & Sneath 1973; Michalski & Stepp 1983; Fisher 1987). The structure of the domain can be modified to suit a specific class of problems (e.g., Fisher 1995). Structured attributes can be used as independent (input) variables (Michalski 1980), as well as dependent (output) variables (e.g., Reinke 1984). The roles of structured variables in these two cases differ.

This paper discusses methods for reasoning with structured attributes in the process of data analysis and knowledge discovery. Many of the presented ideas have been adapted for knowledge discovery from the Inferential Theory of Learning, which provides a unifying framework for characterizing learning and discovery processes (Michalski, 1994). Among the novel ideas are the use of “anchor nodes” to guide the inductive generalization process, the use of non-hierarchical attribute domains, and the introduction of different kinds of generalization rules for structured attributes. The use of continuous variables in reasoning is closely related to that of structured attributes in that the way their domains are structured is tailored to the particular discovery problem. The presented ideas and methods have been implemented in the INLEN-2 knowledge discovery system, and are illustrated by an application to a knowledge discovery problem in world demographics.

## The Extension-Against Operator and Three Types of Descriptions

An inductive generalization rule (or transmutation) takes input information and background knowledge, and hypothesizes more general knowledge (Michalski 1980; 1994). For example, dropping a condition from a decision rule is a generalization transmutation.

A powerful inductive generalization rule used in the AQ learning program is the *extension-against* operator. If rule **R1**:  $[x_i = a]$  & **CTX1** characterizes positive concept examples  $E^+$  and rule **R2**:  $[x_i = b]$  & **CTX2** characterizes negative examples  $E^-$  (where the CTXs stand for any additional conditions), then the extension of R1 against R2 along dimension  $x_i$

$$R1 \text{ ---} | R2 / x_i$$

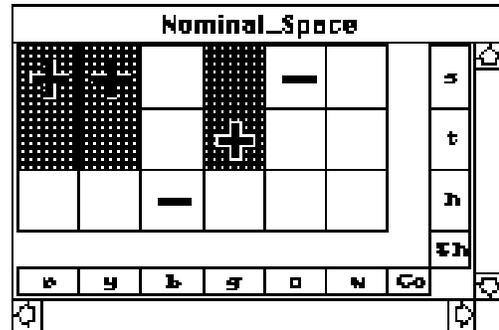
produces **R3**:  $[x_i \neq b]$ , which is the maximal consistent generalization of R1 involving  $x_i$  (Michalski & McCormick 1971; Michalski 1983).

By repeating the extension-against operator until the resulting rule no longer covers any negative examples, a consistent concept description (one that covers no negative examples) can be generated. Such a process can be applied to generate a description (cover) that is complete and consistent with regard to all the training examples (i.e., it covers all positive examples and no negative examples).

Another important concept that needs to be explained before introducing the central ideas of this paper is the type of description, as defined in the AQ rule learning methodology (Michalski et al. 1986). By applying the extension-against operator in different ways, one can generate a range of descriptions with different degrees of generality. Here we distinguish three types of descriptions: 1) *discriminant covers* (in which the extension-against is used to create maximal generalizations; such descriptions specify minimal conditions to discriminate between the concept and non-concept examples); 2) *characteristic covers* (in which the extension-against is specialized to create maximally specific generalizations; such generalizations specify the maximal number of conditions that characterize positive examples); and 3) *minimal complexity covers* (in which the extension-against is used to create the simplest possible generalizations).

To illustrate these concepts, consider the diagram shown in Figure 1, which represents a two-dimensional representation space spanned over three-valued attribute *Shape* (Sh) and six-valued attribute *Color* (Co). Positive and negative concept examples are marked by + and -, respectively. The characteristic cover (maximally specific) is represented by the shaded area. It states: *Color is red or yellow or green, and Shape is square or triangle*. A discriminant description (maximum generalization) of the same set of examples states: Color is red, yellow, green or white (or, equivalently, Color is not blue or orange). A minimum complexity description of these examples states: Color is red or yellow or green. The three descriptions are equally consistent with the five input examples.

These three types of descriptions are used in the INLEN-2 system for knowledge discovery in databases. In the following section, we present an extension of these ideas to descriptions with structured attributes (both as independent (input) and dependent (output) variables) and continuous independent attributes.



Color is red or yellow or green, and Shape is square or triangle

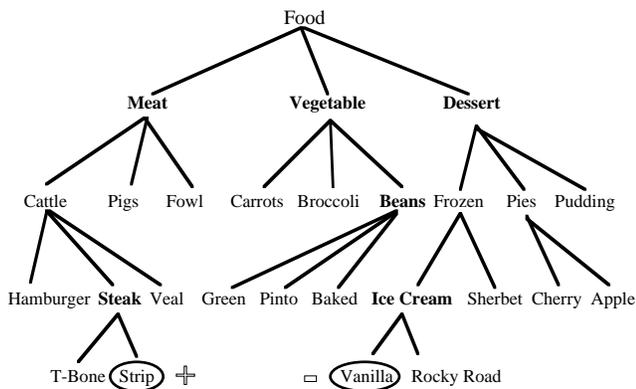
Figure 1. A characteristic cover of  $E^+$  vs.  $E^-$ .

## Generalization Rules for Structured Attributes

### Structured Input Variables

In order to apply the previously defined extension-against operator to structured attributes, new generalization rules need to be defined. Let us illustrate the problem by an example that uses a structured attribute “Food” shown in Figure 2. Each non-leaf node denotes a concept that is more general than its children nodes. These relationships need to be taken into consideration when developing a generalization of some facts. Suppose the concept to be learned is exemplified by statements: “John eats strip steak” and “John doesn’t eat vanilla ice cream.” Many consistent generalizations of these facts exist, for example, that John eats strip steak, steak, cattle, meat, meat or vegetables, or anything but vanilla ice cream. The first statement represents a maximally specific description, the last statement represents a maximally general description, and the remaining ones represent intermediate levels of generalization. A problem arises in determining the generalizations of most interest. We approach this problem by drawing insights from human reasoning.

We tend to assign different levels of significance to nodes in a generalization hierarchy. Some cognitive scientists explain this in part with the idea of *basic* level nodes, whose children share many sensorially recognizable commonalities (Roche et al. 1976). Other factors that are important for characterizing nodes are concept typicality (how common are a concept’s features among its sibling concepts), and the context in which the concept is being used (Klimesch 1988; Kubat, Bratko, & Michalski 1996).



Anchor nodes are shown in bold. Nodes marked by + and - are values occurring in positive and negative examples, respectively.

Figure 2. The domain of a structured attribute “Food.”

To capture such preferences simply, we introduce the idea of *anchor nodes* in a hierarchy. Such nodes are the ones that are desirable to use for the given task domain. To illustrate this idea, consider Figure 2 again.

In the presented hierarchy, vanilla and rocky road are kinds of ice cream; ice cream is a frozen dessert, which is a dessert, which is a type of food. In everyday usage, depending on the context, we will typically think of them as ice cream or dessert, but not so typically as frozen dessert or food.

In designing a knowledge discovery system, we should be able to encode the contextual significance of the nodes into the knowledge representation of the system, so that the created rules will represent desirable levels of abstraction. To this end, nodes in the hierarchy that are considered to be at preferable levels of abstraction are marked as *anchor nodes*.

Given the information which nodes are anchors and which are not, different types of descriptions can be created during the generalization phase of knowledge discovery. The meaning of discriminant and characteristic descriptions needs to be properly defined. In building a characteristic description, the following rule is assumed: Generalize positive examples to the next higher anchor node(s) if this maintains consistency conditions. For example, consider the nodes in bold in Figure 2 to be anchors. Then in characteristic mode, the extension of the positive attribute value “Strip” against the negative attribute “Vanilla” would generalize to “Steak.” In building a discriminant description from these examples, attribute values are generalized to the most general value that does not cover the nearest anchor node to the value of a negative example. In the example above, the positive value “Strip” would generalize (extend) against the value “Vanilla” to “not Ice Cream.” In building the equivalent of minimum complexity descriptions, any of the intermediate anchor nodes could be used if this would simplify the description. For instance, one generalization rule would be to generalize

positive examples to the highest anchor nodes possible, such that consistency is maintained. In this example, the description of “Strip” would generalize to “Meat.”

Another feature to increase the power of representing structured variables is that one does not need to limit the domain of structured attributes to strict hierarchies (one parent for every node). Instead, an arbitrary lattice may be used when desired. This feature is useful for representing overlapping or orthogonal classification hierarchies. The nature of the particular problem determines how the nodes are generalized. The system generalizes the nodes in the way that produces the most desirable (for example, the simplest) description.

Generalization in structured domains does not increase the complexity of the extension-against operator, save for the fact that the internal nodes in the hierarchy must now be among the attribute’s legal value set, while in an unstructured domain they may or may not be present. Any increase in discovery complexity will come during the postprocessing, when different possible generalizations of the discovered cover will be examined; in the worst case, this will be bounded in number of applications by the number of internal nodes in the tree times the tree’s maximum height.

## Structured Output Variables

Typically, dependent variables are numeric or nominal. In the latter case, values of an independent variable are independent concepts. Such a representation fails to take advantage of any generalization hierarchies that may exist among values of a dependent variable. For example, when selecting a personal computer to buy, the candidate models may be grouped into IBM-compatible and Macintosh-compatible. Rather than just choosing one system from among the entire set, it may be simpler to organize the knowledge to choose first the general type, and then the specific model of the assumed type. This leads us to the need for using structured attributes as dependent variables and defining an appropriate method for handling them in the process of rule search and generalization.

Given a structured dependent variable (a decision variable), decision rules or descriptions can relate to nodes of the dependent variable at different levels in the hierarchy. The proposed method focuses first on the top-level nodes, creating rules for them. Subsequently, rules are created for the descendant nodes in the context of their ancestors. For example, in the case of the computer selection problem, the generalization operator would first determine how to choose between IBM- and Macintosh-compatible machines, and then generate rules for distinguishing among the machines in each class, and that class alone. The rule for a specific IBM-compatible machine does not attempt to differentiate it from a particular Macintosh; it will be based on the assumption that an IBM-compatible computer is to be selected. Due to this algorithm, the rules at each level of abstraction will tend to be more concise and easier to interpret.

## Dynamic Quantization of Continuous Attributes

In some respects, numerical attributes are similar to structured attributes with multiple views. There are different ways to group the values into discrete ranges, and unless definitive background knowledge is available, the optimal organization for a particular discovery problem may not be apparent. Even when numeric values have been quantized into ranges by a domain expert, the expert-generated abstraction schema may not be optimal for the learning task (e.g., Karni & Loksh 1996). Furthermore, a particular set of ranges may be useful for one learning task, but irrelevant to another.

The implemented methodology performs automatic discretization of numeric data using the ChiMerge algorithm (Kerber 1992). With this method, neighboring distinct values of the attribute found in the data are merged into single ranges based on a  $\chi^2$  analysis of the classification of the values. When the classification patterns of adjacent ranges are statistically dependent on one another, the ranges are combined into one.

One important consequence of this algorithm is that the grouping of values into intervals will depend on the classification of the input data. Specifically, given a new learning problem from the same data set, the training data will likely be grouped into classes much differently, and thus the set of ranges of a given attribute most likely to generate concise and useful knowledge may be much different. For example, given an auto insurance customer database, there may be little correlation between the set of accident-prone clients and the set of customers who are likely to be interested in purchasing a service offered by the company. It is quite possible that the ranges into which the driver's ages are divided that are most useful for the first learning task are not appropriate for the second one. To combat this problem, ChiMerge recomputes the ranges for each numeric variable whenever new sets of data are added or a new discovery problem is specified.

A limitation of the ChiMerge methodology is that it is dependent on the classification of input training examples, and that it therefore can not be used to quantize an output variable (since that would depend on its already having been divided into classes). There are several ways that an output numeric variable can be discretized into a hierarchy of ranges. A domain expert can suggest ranges, ranges can be determined based on some other output variable, conceptual clustering can be used, etc. A discussion and comparison of these methods will be addressed in a forthcoming paper.

## Experiments

The above ideas and algorithms have been implemented in the INLEN-2 system for knowledge discovery in data. INLEN-2 belongs to the INLEN family of knowledge discovery programs that use an integrated, multi-operator approach to knowledge discovery (Kaufman, Michalski, & Kerschberg 1991; Michalski et al. 1992). This section

illustrates an application of the presented methods to a problem of knowledge discovery in world economics and demographics.

The predominant religion of different countries, as specified in the PEOPLE database of the World Factbook published by the Central Intelligence Agency, is used as an example of a structured attribute. The set of values found in the data contain some natural inclusion relationships. For example, there co-exist labels of "Christian", "Protestant" and "Lutheran" in the database. The organization of some of this attribute's values when structured is shown in Figure 3.

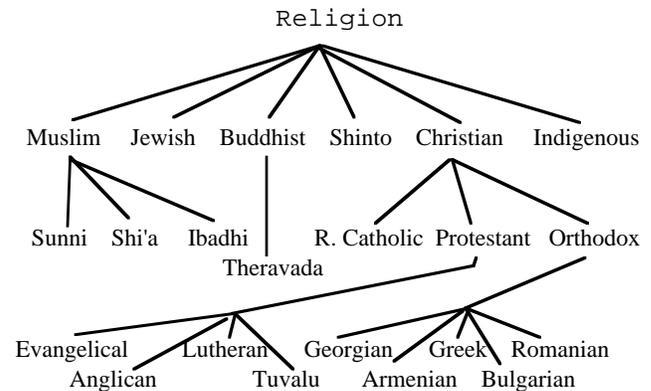


Figure 3. Part of the structure of the PEOPLE database's Religion attribute

If the Religion attribute were set up in an unstructured manner, the statement "Religion is Lutheran" would be regarded equally as antithetical to "Religion is Christian" as to the statement "Religion is Buddhist," leading to the possibility that some contradictions (such as "Religion is Lutheran, but not Christian") might be discovered.

Experiments using INLEN-2 have indicated very interesting findings regarding the usage of structured and non-structured attributes. Among the findings regarding their use as independent variables were that structuring attributes led to simpler rules than when the attributes were not structures, and that certain patterns were only found when the domains were structured. These findings are illustrated by the results below:

When INLEN-2 learned rules to distinguish the 55 countries with low (less than 1 per 1000 people) population growth rate (PGR) from other countries, in a version of the PEOPLE database in which the attribute "Religion" was not structured, one of the rules it found was as follows:

*PGR < 1 if: (20 examples)*  
*Literacy = 95% to 99%,*  
*Life Expectancy is 70 to 80 years,*  
*Religion is Roman Catholic or Orthodox or*  
*Romanian or Lutheran or*  
*Evangelical or Anglican or Shinto,*  
*Net Migration Rate  $\leq$  +20 per 1000 people.*

This rule was satisfied by 20 of the 55 countries with low growth rates.

When the same experiment was run with “Religion” structured, a similar rule was discovered:

*PGR < 1 if: (14 examples)*

*Literacy = 95% to 99%,  
Life Expectancy is 70 to 80 years,  
Religion is Roman Catholic or Orthodox or Shinto,  
Net Migration Rate  $\leq +10$  per 1000 people.*

This rule was satisfied by 14 of the 55 low-growth countries. The removal of some of the Protestant subclasses from the Religion condition and the tightening of the Net Migration Rate condition caused six countries which had been covered by the first rule, not to satisfy this rule. At a slight cost in consistency, simpler, more encompassing rules than either of the above were found:

*PGR < 1 if: (21 examples, 1 exception)*

*Literacy = 95% to 99%,  
Life Expectancy is 70 to 80 years,  
Religion is Christian or Shinto,  
Net Migration Rate  $\leq +10$  per 1000 people.*

Even when not relaxing the consistency constraint, a concise rule not generated in the unstructured dataset was discovered that described 20 countries including the six omitted in the structured rule:

*PGR < 1 if: (20 examples)*

*Birth Rate = 10 to 20 per 1000 people,  
Death Rate is 10 to 15 per 1000 people.*

Similar differences were obtained using structured output variables. By classifying events at different levels of generality, rules can classify both in general and in specific terms. This tends to reduce the complexity and increase the significance of the rules at all levels of generalization.

In the PEOPLE database it is difficult to learn a set of rules for predicting a country’s likely predominant religion given other demographic attributes without using the hierarchies. There are 30 religious categories listed for the 170 countries. The conditions making up the rules will likely have very low support levels (informational significance, defined as number of positive examples satisfying the condition divided by total number of examples satisfying the condition) because a single condition that exists in most of the countries with a certain predominant religion will typically be found, at least occasionally, elsewhere.

When learning rules for distinguishing between the religions in an unstructured format, the highest support level found in the entire rule base was 37% for the awkward condition “Literacy is 70-90% or 95-99%”, which described 26 of the 50 Roman Catholic countries, and 43 of the remainder of the world’s countries. In contrast, structuring the classification of religions led to several top-level conditions with higher support levels. One condition, with a 63% support level in its ability to distinguish the 88 Christian countries was “Population Growth Rate  $\leq 2$ ”.

More drastic effects were seen at the lower levels of the hierarchy. In the unstructured dataset, five rules, each with two to five conditions, were required to define the 11 Sunni

Muslim countries. The only one to describe more than two of the 11 countries was this set of fragmented conditions:

*Religion is Sunni\_Muslim if: (4 examples)*

*Literacy = 100% or less than 30%,  
Infant Mortality Rate is 25 to 40 or  
greater than 55 per 1000 people,  
Fertility Rate is 1 to 2 or 4 to 5 or 6 to 7,  
Population Growth Rate is 1 to 3 or greater than 4.*

The ranges in each of the four conditions are divided into multiple segments, suggesting that this is not at all a strong pattern. In contrast, the structured dataset produced two rules, each with one condition, to distinguish Sunni Muslim countries from other predominantly Islamic nations. The first, “Religion is Sunni\_Muslim if Infant Mortality Rate  $\geq 40$ ”, described all but one of the positive examples (Jordan was the exception), and only one non-Sunni Islamic nation. The second, “Religion is Sunni\_Muslim if Birth Rate is 30 to 40” alone discriminated Algeria, Egypt, Jordan and Tajikistan from the rest of the Muslim world.

Experiments also indicate the practical utility of problem-oriented quantization of continuous variables. Advantages include simpler rules that will often have higher accuracy than with a fixed discretization schema. When characterizing different regions of the world based on an economic database, INLEN-2 was able set thresholds that would generate knowledge with high support levels. For example, the allocation of a country’s GNP to agriculture of greater than 28% (with that number generated by ChiMerge) was a useful indicator for distinguishing Eastern African countries from other regions of the world.

## Conclusion

This paper describes some novel features implemented in the INLEN-2 system for knowledge discovery. Structured and numeric domains share the common trait that many organizational schemas are possible, and the selection of one can have an impact on the success of the discovery process. Structured variables provide a very useful method for providing learning programs with background information about a feature domain, when such is available. A schema for structuring an attribute may be provided either by a domain expert or by a learning system. The implementation of structured attributes can introduce the concepts of generalization, agglomeration, anchor nodes and multiple domain views to a discovery system.

In large databases the structuring of nominal or numeric attributes can assist in the discovery process. In most attribute domains in which there are more than just a few distinct values, structuring of the domain is usually both possible and recommended. There will generally be a way to organize the values according to some classification schema. This allows the import of background knowledge, even to those empirical discovery engines that traditionally rely on a minimum of background knowledge. This domain knowledge, in turn, can result in the discovery of

relationships more attuned to the user's existing understanding of the background domain through such techniques as the definition of anchor nodes. Attribute structuring can also be used for output variables. Doing so provides a means for separating the tasks of determining the general class of the decision and determining the specific decision within that class.

By allowing multiple representations of structured and numeric data, adaptive representation selection may be possible, potentially leading to discovering relationships that can alter a user's preconceived notions about the domain. The learning engine can select representations for the attribute domains after, rather than before the learning. These representations may include not only variations on one basis for classification, but also orthogonal classification hierarchies. Similarly, problem-oriented quantization of numeric data may enhance the likelihood of useful results in continuous domains.

Techniques such as anchor nodes and multiple domain views can help create an environment in which a representation space suitable to the problem is selected, while attention is focused on the levels of abstraction of greatest utility to the user. One area for future research is the development of a representation of a node's significance beyond a simple anchor/non-anchor value, and the exploration of thresholds for determining the proper level of generalization in such an environment.

### Acknowledgments

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 Defense Advanced Research Projects Agency under Grant No. N00014-91-J-1854 administered by the Office of Naval Research, in part by the Defense Advanced Research Projects Agency under Grants No. F49620-92-J-0549 and F49620-95-1-0462 administered by the Air Force Office of Scientific Research, in part by the Office of Naval Research under Grant No. N00014-91-J-1351, and in part by the National Science Foundation under Grants No. DMI-9496192 and IRI-9020266.

### References

Fisher, D. 1987. Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning*, 2:139-172.

Fisher, D. 1995. Optimization and Simplification of Hierarchical Clusterings. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, Montreal, PQ, 118-123.

Karni, R. and Loksh, S. 1996. Generalization Trees as Discovered Knowledge for Manufacturing Management. Forthcoming.

Kaufman, K., Michalski, R.S. and 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*, Menlo Park, CA: AAAI Press, 449-462.

Kerber, R. 1992. ChiMerge: Discretization of Numeric Attributes. *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Jose, CA, 123-127.

Klimesch, W. 1988. *Struktur und Aktivierung des Gedächtnisses. Das Vernetzungsmodell: Grundlagen und Elemente einer uebergreifenden Theorie*. Bern: Verlag Hans Huber.

Kubat, M., Bratko, I. and Michalski, R.S. 1996. A Review of Machine Learning Techniques. Chapter in *Methods and Applications of Machine Learning and Discovery* (forthcoming).

Michalski, R.S. and McCormick, B.H. 1971. Interval Generalization of Switching Theory. *Proceedings of the 3rd Annual Houston Conference on Computer and System Science*, Houston, TX.

Michalski, R.S. 1980. Inductive Learning as Rule-Guided Generalization and Conceptual Simplification of Symbolic Descriptions: Unifying Principles and a Methodology. *Workshop on Current Developments in Machine Learning*, Carnegie Mellon University, Pittsburgh, PA.

Michalski, R.S. 1983. A Theory and Methodology of Inductive Learning. Chapter in Michalski, R.S., Carbonell, J. and Mitchell, T. (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Palo Alto: Tioga Publishing, Co., 83-134.

Michalski, R.S. 1994. Inferential Theory of Learning: Developing Foundations for Multistrategy Learning. Chapter in *Machine Learning: A Multistrategy Approach*, Michalski, R.S. and Tecuci, G. (Eds.), San Francisco: Morgan Kaufmann, 3-61.

Michalski, R.S., Kerschberg, L., Kaufman, K. and Ribeiro, J. 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.

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

Michalski, R.S. and Stepp, R.E. 1983. Automated Construction of Classifications: Conceptual Clustering versus Numerical Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4): 396-410.

Reinke, R.E. 1984. Knowledge Acquisition and Refinement Tools for the Advise Meta-Expert System. Master's Thesis, University of Illinois at Urbana-Champaign.

Rosch, E., Mervis, C., Gray, W., Johnson, D. and Boyes-Braem, P. 1976. Basic Objects in Natural Categories, *Cognitive Psychology*, 8:382-439.

Sokal, R.R. and Sneath, P.H. 1973. *Principles of Numerical Taxonomy*. San Francisco: W.H. Freeman.