Proceedings of the Ninth International Symposium on Methodologies for Intelligent Systems (ISMIS-96), Zakopane, Poland, June 10-13, 1996

The AQ17-DCI System for Data-Driven Constructive Induction and Its Application to the Analysis of World Economics

Eric Bloedorn and Ryszard S. Michalski[1]

Machine Learning and Inference Laboratory, George Mason University, Fairfax VA

[1]Also with GMU Departments of Computer Science and Systems Engineering, and the Institute of Computer Science at the Polish Academy of Sciences, Warsaw, Poland

Abstract. Constructive induction divides the problem of learning an inductive hypothesis into two intertwined searches: one—for the "best" representation space, and two—for the "best" hypothesis in that space. In *data-driven* constructive induction (DCI), a learning system searches for a better representation space by analyzing the input examples (data). The presented data-driven constructive induction method combines an AQ-type learning algorithm with two classes of representation space improvement operators: *constructors*, and *destructors*. The implemented system, AQ17-DCI, has been experimentally applied to a GNP prediction problem using a World Bank database. The results show that decision rules learned by AQ17-DCI outperformed the rules learned in the original representation space both in predictive accuracy and rule simplicity.

1 Introduction

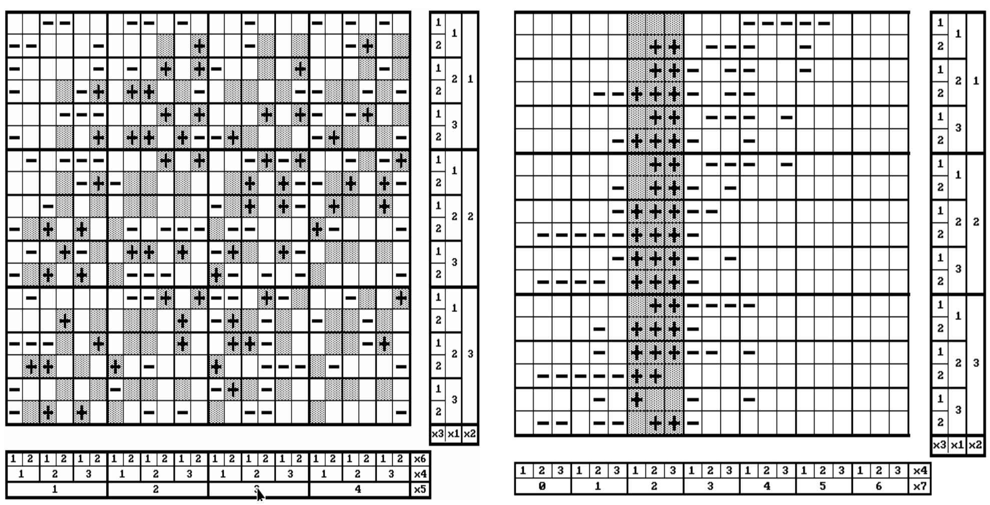
The basic premise of research on *constructive induction* (CI) is that results of a learning process directly depend on the quality of the representation space in which is occurs. If the representation space is well designed, then learning results will tend to be satisfactory with almost any method (assuming an adequate representation language); otherwise, they may be poor regardless of the method. Constructive induction is oriented toward learning problems in which the representation space, as defined by attributes in the training examples, is of low quality or there is a mismatch between the representation language used and the target concept. A low quality representation space means that the space is spanned over attributes that are weakly relevant or irrelevant for the given learning task. To cope with such problems, constructive induction splits a learning process to two intertwined searches — one for the "best" representation space, and the second for the "best" hypothesis in the found space. Using another terminology that is also used in machine learning, constructive induction includes the problem of automatically determining the best "representation space bias" as a part of the induction process.

The idea of constructive induction is not new [23]. Initial research on this topic concentrated solely on constructing new attributes beyond those provided in the input data [21] [36] [16] [32]. Michalski [23] presented a set of *constructive generalization rules* that describe various ways in which new attributes can be generated. More recent work has viewed constructive induction more generally, namely, as a double-search process, in which one search is for an improved representation space and the second for "best" hypothesis in this space [2] [3] [38]. The improvement of a representation space is done in several ways—by generating new attributes, by removing less relevant or irrelevant attributes, and/or by abstracting values of given attributes (grouping values to larger units).

The search for an improved representation space can be guided by information from three sources [38]: training data (as in data-driven constructive induction—DCI), initial hypotheses learned from the data (as in hypothesis-driven constructive induction—HCI), or expert knowledge provided by the user to the system (as in knowledge-driven constructive induction—KCI). These sources can also be combined into a multistrategy constructive induction method. This paper describes a data-driven method of constructive induction and its application to a problem of learning economic relationships.

2. An Illustration of the Importance of the Representation Space

The concept representation space is defined as a space in which inductive hypotheses are generated. In conventional machine learning methods this space is identical to the space in which training examples are represented. As mentioned earlier, the choice of the representation space has a profound effect on the quality of the generated hypotheses. This effect is well illustrated by the second Monk's problem [34]. The original representation space with the training examples denoted by + and - is shown in Figure 1(a) using DIAV [39]. The shaded area represents the target concept.



(a) (b)

Figure 1. Diagrammatic visualization of the Monk 2 representation spaces: (a) the initial space; and (b) improved space due to the data-driven constructive induction.

In this original representation space, the learning problem is difficult because the target concept is highly irregular. An improved representation space, as found by the AQ17-DCI system described in this paper, is shown in Fig. 1,b. In the improved representation space the target concept is highly regular and therefore easy to learn. In this case, the new representation was found by applying a space modification operator that generates the so-called *counting attribute*: #Attr(S, P), which counts the number of attributes in the set S with a property P. The system found that the counting attribute in which S contains all original attributes {x1, x2, x3, ..., x6}, and P is the first value of attribute in its domain is highly relevant for the task at hand. The learned concept was: : #Attr({x1,...x6}, Firstvalue) = 2, that is, *an example belongs to the concept, if exactly two of six attributes take their first value.* This rule exactly represents the intended target concept, and thus has a predictive accuracy of 100%.

3. Relation to Other Research

Research on constructive induction has produced a number of working programs. The first program that was explicitly dedicated to exhibit constructive induction capabilities was INDUCE [22]. INDUCE generates new attributes or new predicates by applying various constructive generalization rules. BACON.3 [14] and ABACUS [9] search for mathematical relationships or laws that summarize numerical (or numerical and symbolic) data. Lenat’s AM and Eurisko programs [15] can be viewed as performing a form of knowledge-based constructive induction, as they generate new concepts according to certain heuristics.

Schlimmer’s STAGGER [33] is a constructive induction program that uses three cooperating learning modules: weight adjustment, Boolean feature construction, and attribute value aggregation. Muggleton's Duce [25] is an oracle-based approach (knowledge-driven). Pagallo and Haussler's FRINGE, GREEDY3 and GROVE [26] base the construction of new attributes on patterns found in learned decision trees. Another decision tree-based method is CITRE [16], which constructs new terms by repeatedly applying Boolean operators to nodes on the positively labeled branches. A hypothesis-driven approach based on decision rules is AQ17-HCI [38]. In this system patterns prevalent in strong rules are used for constructing new attributes. An approach which uses disjunctive or arithmetic combinations of the original attributes to extend the initial attribute set was developed by Utgoff in STABB [35].

As mentioned earlier, constructive induction is a process in which the original representation space is improved during learning. This can be done by generating new attributes and/or removing less relevant or redundant ones. The latter process has been investigated in the rough set approach (e.g., [27], [41] [28], and [10]). The AQ17-DCI system presented in this paper executes a complementary process of selecting the most relevant attributes. This is done by applying some measure of attribute relevance, for example, PROMISE [1] or information gain ratio [30]. The rough set approach also applies an attribute-value abstraction operator, which removes values that are not needed for describing data. In contrast to this, AQ17-DCI combines less relevant values with adjacent values into larger units, that is, performs an abstraction of the attribute domain.

In summary, AQ17-DCI has the following characteristics which together distinguish it from all other constructive induction methods: 1) the search for a better representation space is based on patterns found in the training data, and is thus not tied to a specific knowledge representation language (as is the case of hypothesis-driven induction); 2) it applies three classes of operators: attribute construction, attribute reduction, and attribute value abstraction 3) it uses both domain-independent or domain-dependent constructive generalization rules; 4) it supports binary as well as multi-argument attribute construction operators.

4. The AQ17-DCI System for Data-Driven Constructive Induction

4.1 Overview

The AQ17-DCI system consists of two components. One component performs a data-driven search for an improved representation space (hence DCI—data-driven constructive induction). The second component employs the AQ15c program for searching for the "best" hypothesis within the current representation space (the name AQ17-DCI means that this is the 17th program in the family of AQ algorithm based induction programs). The AQ algorithm generates an optimal or near-optimal set of rules characterizing training examples, according to a given criterion of optimality (originally described in [18], and [19]). The criterion of optimality may take into consideration such factors as the number of rules, the number of conditions, the cost of attributes in the rules, and others. These factors can be combined into a multicriterion measure of optimality that best reflects the needs of the learning problem at hand.

4.2 Search for an Improved Representation Space

The search for an improved representation space employs three types of representation space modification operators: 1) space expansion through attribute construction (GENERATE), 2) space contraction through selecting only the most relevant attributes from the original space (SELECT), and 3) space contraction through attribute-value abstraction (QUANT). Experiments have shown that representation space expansion is very useful when the attribute construction operators are well-matched with the problem at hand. Representation space contraction, however, must be performed with great care, as it may lead to a removal of information that is crucial for learning a correct hypothesis [5]. For this reason, default thresholds on the space contraction operators are set conservatively. The following sections describe these operators in more detail.

4.2.1 Space Expansion: Attribute Construction By GENERATE

The GENERATE method for constructing new attributes employs both mathematical and logical operators to construct new attributes. In selecting attributes for applying mathematical operators, the system takes into consideration the attribute types. The operation to be performed on the selected attributes is done according to predefined rules (defined by the user). With the attributes and operation selected, the values for the new attribute are calculated. The usefulness of new attributes is evaluated using an attribute quality measure. If the quality measure is above a user-defined threshold, the attribute is added to the available attribute set. If it is below the threshold or the attribute is too complex, the new attribute is discarded. The algorithm applies a variety of relational or arithmetic operators to numeric attributes. A summary of representation space expansion operators is presented in Table 1.

Operator	Arguments	Notation	Interpretation
Equivalence	Attributes x,y	x = y	If x = y then 1, otherwise 0
Greater Than	Attributes x,y	x > y	If x = y then 1, otherwise 0
Greater Than or Equal	Attributes x,y	x>=y	If x $\geq$y then 1, otherwise 0
Addition	Attributes x,y	x + y	Sum of x and y
Subtraction	Attributes x,y	x - y	Difference between x and y
Difference	Attributes x,y	|x - y|	Absolute difference between x and y

Multiplication	Attributes x,y	x * y	Product of x and y
Division	Attributes x,y	x/y	Quotient of x divided by y
Maximum	Attribute set S	Max{S}	Maximum value in S
Minimum	Attribute set S	Min{S}	Minimum value in S
Average	Attribute set S	Ave{S}	Average of values in S
Counting	Attribute set S,P	#Attr(S,P)	No of attributes in S with property P

Table 1. A summary of representation space expansion operators in AQ17-DCI.

4.2.2 Representation Space Contraction

AQ17-DCI contracts the representation space by abstracting attribute-values using QUANT, or by selecting most relevant attributes using SELECT. Table 2 summarizes the representation space contraction operators used in AQ17-DCI. The next two subsections provide details.

Operator	**Arguments**	**Notation**	**Interpretation**
Quantization	An attribute x, and a method M	QUANT(x, M)	Quantization of x using method M Methods available: Chi-merge, Equal-interval and Equal frequency
Selection	Set of attributes S	SELECT(S)	Select subset S' of S by method M Methods: Promise, Information gain

Table 2. A summary of representation space contraction operators in AQ17-DCI.

4.2.2.1 Attribute-value Abstraction Using QUANT

Research on attribute-value abstraction is usually performed under the name attribute-value discretization [7], [8]. We view this process as a form of abstraction because the result of it is a decrease of information about an object [24]. By replacing original attribute values by more abstract ones the representation space is reduced, thus this process represents a representation space contraction transformation. QUANT abstracts attribute values using the ChiMerge method described by Kerber [13]. This abstraction, a.k.a. scaling, is performed for continuous attributes and for discrete attributes with large domains. Because it reduces the size of the representation space, abstraction can significantly speed up the search for hypothesis. It can also improve the quality of hypothesis due to a simplification of the generated descriptions.

The ChiMerge algorithm is a bottom-up process in which initially all values are stored in separate intervals which are then merged until a termination condition is met. The interval merging process consists of continuously repeating two steps: 1) compute χ^2 values (correlations between the value of the class attribute and the value of an attribute), and 2) merge the pair of adjacent intervals with the lowest χ^2 value. Intervals are merged until all pairs of intervals have χ^2 values exceeding the user defined chi-threshold. The chi-threshold is a function of the desired significance level and the number of degrees of freedom (1 fewer than the number of classes), and can be determined from a table. The χ^2 value measures the probability that the attribute interval and class value are dependent. If the interval has a χ^2 value greater than threshold then class and interval are correlated and are retained. High χ^2 threshold settings cause more intervals to be merged which results in fewer intervals, or abstracted attribute values.

4.2.2.2 Attribute Selection Using SELECT

This operation is conventionally described in the literature as *feature selection*. Here, we view this process more generally as a form of reduction of the representation space. SELECT uses a measure of attribute relevance, such as well-known information gain ratio [30], or PROMISE [1] for the purpose of determining which attributes should be used for defining the representation space in which search for the inductive hypothesis will occur. The attributes that score on the attribute relevance measure above a certain threshold are selected as dimensions of the transformed representation space.

5. An Experimental Application to World Economics

This section describes an application of the AQ17-DCI system to a problem of determining the economic and demographic patterns in the countries of the world. The data were obtained from a World Bank database [6]. This database contains economic and demographic records for the countries of the world from 1965 to 1990 [12]. The goal of the experiment presented here was to determine a set of rules characterizing the dependence of GNP (Gross National Product) in various countries during the period from 1986 to 1990 on the available economic and demographic characteristics.

In the experiment we considered 41 countries. Changes of GNP were quantified into four equal-intervals: low (0 to 0.5625), medium (0.5626 to 1.125), high (1.126 to 1.6875) and very high (over 1.6875). The countries were described by 11 attributes, each sampled over a period of 5 years. Thus, each country was described by 55 attributes. AQ17-DCI was applied to determine rules that characterize GNP changes in terms of the given attributes (or their relevant subset).The quality of the generated rules was evaluated by the 10-fold cross-validation method [37]. The learning process involved the following steps:

1. <u>Remove less relevant attributes</u>. AQ17-DCI removed 24 attributes using the SELECT method described in Section 4.2.2.2 with a information threshold set to 0.6.

2. <u>Abstract away unnecessary detail</u>. The domain size of the attributes were reduced from an average of 15.4 values per attribute to 4.3, using the QUANT described in section 4.2.2.1.

3. <u>Construct new problem-relevant attributes.</u> On the average, AQ17-DCI constructed ten new attributes in each run, using the DCI GENERATE method. Examples of these new attributes are shown in Table 3.

Name	Operator used	Description
ChgeEnergyCons86-88	Minus	Change in energy consumption of a country between 1986 and 1988
Birth89ByEnergyCons90	Division	Ratio of Crude Birth Rate in 1989 to Energy Consumption in 1990
AveEnergyCons86-90	Average	Average Energy Consumption of a country between 1986 and 1990

Table 3: Examples of new relevant attributes constructed by AQ17-DCI

The transformations generated by AQ17-DCI resulted in an approximately 80% increase in predictive accuracy. Rules learned in the original representation were only 41.7% accurate, while rules learned in the improved representation space were 76.3% accurate on the testing data. While this improvement is significant, it is not entirely satisfactory. The problem may have been caused by the presence of misclassification errors (incorrectly classified training examples or to the lack in the original data of important relevant attributes (e.g., the type of government, presence of natural disasters, or war). Further research is being done to try to understand how predictive accuracy can be improved.

Constructive induction helped not only to increase the predictive accuracy, but also generated a number of meaningful new attributes. A significant advantage of this method is that rules generated can be easily interpreted by a data analyst, as they are in the form that is directly interpretable in natural language. Here is an example of such a rule:

> **Countries with very high increase in GNP are characterized by:**
>
> **[DeathRate is low] &**
> **[*AVG(%PopulationAgeBracketB)* is very high] &**
> **[*AVG(PopulationGrowthRate)* is low]**
> *(Total: 13, Unique 9)*
> **OR**
> **[*AVG(UrbanPopulationGrowth)* is very low] &**
> **[*AVG(UrbanVsRuralGrowthDifference)* is very low]**
> *(Total: 7, Unique 3)*

where

--"very high increase" is defined as *GNP per capita (in US$) 1990 / GNP 1986 > 1.7,*

--low death rate (per thousand) is defined as 5% to 6%,

--*AVG(%PopulationAgeBracketB)* stands for the average % of population in the age bracket B (ages 15..64); very high means 57%-69%

--*AVG(PopulationGrowthRate)* stands for the average of population growth rate; low means less than 3%

--very low average urban population growth rate means lower than 1%

--very low average difference between the urban and rural growth means less than 2%

--*Total* denotes the total number of countries in the study that are covered by the rule

--*Unique* denotes the number of countries in the study that were covered by this rule and not by any other rule.

Comments:
All averages were computed for the period 1986 to 1990.
Attributes in italics were constructed by the program from the initial attributes.

Rule Interpretation
Countries with a very high increase in GNP are characterized by low death rate, the average percentage of the population age 15 to 64 year olds is very high, and the overall population growth rate is low OR

the average urban population growth is very low, and the average difference between urban and rural population growth rate is very low.

Another useful aspect of the AQ17-DCI system is that it can generate rules that optimize criteria set by the user to best reflect the needs of the task at hand. The user is able to select the type of operators to be used, as well as the information threshold above which new attributes must score in order to be retained. This gives a data analyst a way to generate data descriptions that are most suitable for a given task, and may also help to get insights into the problem that were not possible in the original representation space. The program has default values so that the user does not have to set all of these parameters before making use of these capabilities.

6. Summary

This paper described a method and for data-driven constructive induction, which combines AQ-type rule learning with operators for representation space improvement. These operators can expand the space through attribute construction, and/or contract the space through attribute removal and attribute-value abstraction. The system implementing the method, AQ17-DCI, was tested by applying it to a problem in the area of world economics. In the experiments, AQ17-DCI produced decision rules that had higher predictive accuracy than rules learned in the original representation space, and generated new attributes that provided additional insights into the data. In the GNP prediction problem the space obtained by applying all three types of operators (attribute generation, attribute removal and value abstraction) produced the rules with the highest predictive accuracy. Future research will focus on implementing other operators for attribute construction and the development of a control strategy for guiding the selection of individual operators.

Acknowledgments

The authors thank members of the INLEN group for discussions and criticism. 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. Baim, P.W., "The PROMISE Method for Selecting the Most Relevant Attributes for Inductive Learning Systems", Rep. No. UIUCDCS-F-82-898, Dept. of Computer Science, University of Illinois-Urbana Champaign, IL, 1982.

2. Bloedorn, E. and Michalski, R.S. "Data-Driven Constructive Induction in AQ17-PRE: A Method and Experiments", *Proceedings of the Third International Conference on Tools for AI*, November 1991a.

3. Bloedorn, E. and Michalski, R.S., "Constructive Induction from Data in AQ17-DCI: Further Experiments," *Reports of the Machine Learning and Inference Laboratory,* MLI 91-12, School

of Information Technology and Engineering, George Mason University, Fairfax, VA, December 1991b.

4. Bloedorn, E., Michalski, R., and Wnek, J., "Multistrategy Constructive Induction," *Proceedings of the Second International Workshop on Multistrategy Learning*," Harpers Ferry, WV, May 26-29, 1993.

5. Bloedorn, E., Michalski, R.S., and Wnek, J., "Matching Methods with Problems: A Comparative Analysis of Constructive Induction Approaches", *Reports of the Machine Learning and Inference Laboratory*, MLI 94-2, George Mason University, Fairfax, VA, 1994.

6. Bloedorn, E. and Kaufman, K, "Data-Driven Constructive Induction in INLEN", *Reports of the Machine Learning and Inference Laboratory*, George Mason University, Fairfax, VA, 1996 (to appear).

7. Dougherty, J., Kohavi, R., and Sahami, M., "Supervised and Unsupervised Discretization of Continuous Features", *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 194-201, June 1995.

8. Fulton, T., Kasif, S., and Salzberg S., "Efficient Algorithms for Finding Multi-way Splits for Decision Trees", *Proceedings of the Twelfth International Conference on Machine Learning,* pp. 244-251., June 1995.

9. Greene, G.H., "Quantitative Discovery: Using Dependencies to Discover Non-Linear Terms", *M.S. Thesis*, University of Illinois at Urbana-Champaign, 1988.

10. Gryzmala-Busse, J.W., "LERS - A System for Learning from Examples based on Rough Sets" in Slowinski, R., Ed. Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, pp. 3-18. 1992.

11. Jensen, G., "SYM-1: A Program that Detects Symmetry of Variable-Valued Logic Functions", *Report UIUCDCS-R-75-729*, Department of Computer Science, University of Illinois at Urbana-Champaign, 1975.

12. Kaufman, K., "Comparing International Development Patterns Using Multi-Operator Learning and Discovery Tools", *Proceedings of the AAAI-94 Workshop on Knowledge Discovery in Databases,* Seattle, WA, pp. 431-440. 1994.

13. Kerber, R., "ChiMerge: Discretization of Numeric Attributes", *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 123-128, San Jose, CA, 1992.

14. Langley, P., "Rediscovering Physics with Bacon 3," *Fifth International Joint Conference on Artificial Intelligence*, pp. 505-507, Cambridge, MA: , 1977.

15. Lenat, Douglas, "Learning from Observation and Discovery", in *Machine Learning: An Artificial Intelligence Approach, Vol. I,* R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), Palo Alto, CA: Morgan Kaufmann (reprint), 1983

16. Matheus, C. J. and Rendell, L.A., "Constructive Induction on Decision Trees", In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 645-650, 1989.

17. Michalski, R.S., "Recognition of Total or Partial Symmetry in a Completely or Incompletely Specified Switching Function," Proceedings of the IV Congress of the International Federation on Automatic Control (IFAC), Vol. 27 (Finite Automata and Switching Systems), pp. 109-129, Warsaw, June 16-21, 1969.

18. Michalski, R.S., "On the Quasi-Minimal Solution of the Covering Problem" *Proceedings of the V International Symposium on Information Processing* (FCIP 69), Vol. A3 (Switching Circuits), Bled, Yugoslavia, pp. 125-128, 1969.

19. Michalski, R.S. and McCormick, B.H., "Interval Generalization of Switching Theory." Report No. 442, Dept. of Computer Science, University of Illinois, Urbana. 1971.

20. Michalski, R.S., “Variable-Valued Logic: System VL_1, *Proceedings of the 1974 International Symposium on Multiple-Valued Logic*, pp. 323-346. West Virginia University, Morgantown, 1974.

21. 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*, Dept. of Computer Science, University of Illinois, Urbana, 1978.

22. Michalski, R.S., “Pattern Recognition as Rule-Guided Inductive Inference,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI)*, Vol. 2, No. 4, pp. 349-361, 1980.

23. Michalski, R.S., “A Theory and Methodology of Inductive Learning: Developing Foundations for Multistrategy Learning,” in *Machine Learning: An Artificial Intelligence Approach, Vol. I*, R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), Palo Alto, CA: Morgan Kaufmann (reprint), 1983.

24. Michalski, R.S., “Inferential Theory of Learning,” in *Machine Learning: An Multistrategy Approach,* Vol. IV, R.S. Michalski, and G. Tecuci (Eds.), Palo Alto, CA: Morgan Kaufmann, 1994.

25. Muggleton, S., "Duce, an Oracle-Based Approach to Constructive Induction", *Proceedings of IJCAI-87*, pp. 287-292, Morgan Kaufman, Milan, Italy, 1987.

26. Pagallo, G., and Haussler, D., "Boolean Feature Discovery in Empirical Learning", *Machine Learning*, vol. 5, pp. 71-99, 1990.

27. Pawlak, Z. “Rough Sets and their Applications”, Workshop on Mathematics and AI, Schloss Reisburg, W. Germany. Vol II. pp. 543-572. 1988.

28. Pawlak, Z. “Rough Sets: Theoretical Aspects of Reasoning about Data”, Kluwer Academic Publishers, AA Dordrecht, The Netherlands, 1991.

29. Quinlan, J. R., “Learning Efficient Classification Procedures,” *Machine Learning: An Artificial Intelligence Approach*, Michalski, R.S., Carbonell, J.G, and Mitchell, T.M. (Eds.), Morgan Kaufmann 1983, pp. 463-482.

30. Quinlan, J.R., “C4.5: Programs for Machine Learning”, Morgan Kaufmann, San Mateo, CA, 1993.

31. Reinke, R.E., “Knowledge Acquisition and Refinement Tools for the ADVISE Meta-expert System,” *Master’s Thesis*, University of Illinois, 1984.

32. Rendell, L., and Seshu, R., “Learning Hard Concepts Through Constructive Induction: Framework and Rationale,” *Computer Intelligence,* Vol. 6, pp. 247-270, 1990.

33. Schlimmer, J., "Concept Acquisition Through Representational Adjustment," *Machine Learning*, Vol. 1, pp. 81-106, 1986.

34. Thrun, S.B., Bala, J., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., De Jong, K., Dzerowski, S., Fahlman, S.E., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R.S., Mitchell, T., Pachowicz, P., Vafaie, H., Van de Velde, W., Wenzel, W., Wnek, J., and Zhang, J., “The MONK’S Problems: A Performance Comparison of Different Learning Algorithms,” *(revised version)*, Carnegie Mellon University, Pittsburgh, PA, CMU-CS-91-197, 1991.

35. Utgoff, P., "Shift of Bias for Inductive Learning,", in *Machine Learning: An Artificial Intelligence Approach*, Vol. II, R. Michalski, J. Carbonell, and T. Mitchell (eds.), Morgan Kaufman, Los Altos, CA, pp. 107-148, 1986.

36. Watanabe, L., and Elio, R., “Guiding Constructive Induction for Incremental Learning from Examples,” *Proceedings of IJCAI-87*, pp. 293-296, Milan, Italy: , 1987.

37. Weiss, S. M., and Kulikowski, C. A., *Computer Systems that Learn*, Morgan Kaufmann, San Mateo, CA. 1991.

38. Wnek, J. and Michalski, R., "Hypothesis-driven Constructive Induction in AQ17-HCI: A Method and Experiments," *Machine Learning*, Vol. 14, No. 2, pp. 139-168.1993.

39. Wnek, J. "DIAV 2.0 User Manual: Specification and Guide through the Diagrammatic Visualization System," *Reports of the Machine Learning and Inference Laboratory*, MLI95-5, George Mason University, Fairfax, VA 1995.

40. Wnek, J., Kaufman, K., Bloedorn, E., and Michalski, R.S., "Selective Induction Learning System AQ15c: The Method and User's Guide", *Reports of the Machine Learning and Inference Laboratory*, MLI 95-4.

41. Ziarko, W. "On Reduction of Knowledge Representation", *Proceedings of the 2nd International Symposium on Methodologies for Intelligent Systems*, Charlotte, NC. North Holland, pp. 99-113.